Package ‘GeoModels’

December 23, 2020

Version 1.0.3-4

Date 2018-01-15

Title A Package for Geostatistical Gaussian and non Gaussian Data Analysis

Author Moreno Bevilacqua [aut, cre],
Víctor Morales-Onate [aut]

Maintainer Moreno Bevilacqua <moreno.bevilacqua@uv.cl>

Description This package provides a set of procedures for a) simulation and estimation of some spatial and spatio-temporal random fields using standard likelihood and a likelihood approximation method called composite likelihood and b) prediction using best linear unbiased prediction. Spatio (temporal) bivariate data estimation involves estimation of both regression and covariance parameters.
Gaussian and some non Gaussian Random fields can be analyzed using the GeoModels package. Among them, Weibull, logGaussian, skewGaussian, T, binomial, negative binomial and circular random fields can be analyzed.

Imports methods, spam, scatterplot3d, dfoptim, dotCall64, optimParallel, parallel, pracma,mcGlobaloptim, pbivnorm, zipfR, sn, numDeriv, hypergeo, ucminf, LatticeKrig, RANN, VGAM

Suggests actuar, GoFKernel, sphereplot

Depends R (>= 2.12.0), Rfast, fields, mapproj, plot3D, shape

License GPL (>= 2)

URL https://vmoprojs.github.io/GeoModels-page/

Repository GitHub

Encoding UTF-8

BugReports https://github.com/vmoprojs/GeoModels/issues

R topics documented:

anomalies 2
austemp 3
CheckBiv 3
CheckDistance 4
CheckSph 5
CheckST 5
CkCorrModel 6

CkInput 7
CkLikelihood 8
CkModel 9
CkType 10
CkVarType 10
CompLik 11
CompLik2 13
CorrelationPar 15
CorrParam 16
DeviceInfo 17
GeoCovariogram 18
GeoCovDisplay 23
GeoCovmatrix 24
GeoCV 34
GeoFit 36
GeoKrig 50
GeoKrigloc 61
GeoNeighborhood 66
GeoQQ 69
GeoResiduals 70
GeoScatterplot 72
GeoSim 74
GeoTests 80
GeoVarestbootstrap 84
GeoVariogram 85
GeoWLS 90
Lik 93
MatDecomp 95
MatSqrt, MatInv, MatLogDet 96
NuisParam 97
Prscores 98
StartParam 99
winds 101
winds.coords 101
WlsStart 102

Index **104**

`anomalies` *Annual precipitation anomalies in U.S.*

Description

A ($7252x3$)-matrix containing lon/lat and yearly total precipitation anomalies registered at 7.352 location sites in USA. For more details see http://www.image.ucar.edu/Data/precip_tapering/.

Usage

```
data(anomalies)
```

Format

A numerical matrix of dimension $7252x3$.

Source

Kaufman, C.G., Schervish, M.J., Nychka, D.W. (2008) Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association, Theory & Methods*, **103**, 1545–1555.

`austemp` *Maximum australian temperature*

Description

A matrix containing maximum temperature in Australia in July 2011.

Usage

```
data(austemp)
```

Format

A (446×4)-matrix containing longitude, latitude, maximum temperature, and the 'so called' geometric temperature covariate.

Source

Bevilacqua M., Caamaño C., Morales-Oñate V., Arellano-Valle R. B. (2020) Non-Gaussian Geostatistical Modeling using (skew) t Processes, *Scandinavian Journal of Statistics*.

`CheckBiv` *Checking Bivariate covariance models*

Description

The procedure control if the correlation model is bivariate.

Usage

```
CheckBiv(numbermodel)
```

Arguments

`numbermodel`	numeric; the number associated to a given correlation model.

Details

The function check if the correlation model is bivariate.

Value

Returns TRUE or FALSE depending if the correlation model is bivariate or not.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

Examples

```
library(GeoModels)
CheckBiv(CkCorrModel("Bi_matern_sep"))
```

CheckDistance *Checking Distance*

Description

The procedure controls the type of distance.

Usage

```
CheckDistance(distance)
```

Arguments

distance	String; the type of distance, for the description see GeoCovmatrix. Default is Eucl. Other possible values are Geod and Chor that is euclidean, geodesic and chordal distance.

Details

The function check if the type of distance is valid.

Value

Returns 0,1,2 for euclidean,geodesic, chordal distances respectively. Otherwise returns NULL.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

`CheckSph` *Checking if a covariance is valid only on the sphere*

Description

Subroutine called by InitParam. The procedure controls if a covariance model is valid only on the sphere.

Usage

```
CheckSph(numbermodel)
```

Arguments

`numbermodel`	Numeric; the code number for the covariance model.

Details

The function checks if a covariance is valid only on the sphere

Value

Returns TRUE or FALSE

Author(s)

Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, `<victor.morales@uv.cl>`, https://sites.google.com/site/moralesonatevictor/

`CheckST` *Checking SpaceTime covariance models*

Description

The procedure control if the correlation model is spacetime.

Usage

```
CheckST(numbermodel)
```

Arguments

`numbermodel`	numeric; the number associated to a given correlation model.

Details

The function check if the correlation model is spacetime.

Value

Returns TRUE or FALSE depending if the correlation model is spacetime or not.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

Examples

```
library(GeoModels)
CheckST(CkCorrModel("gneiting"))
```

CkCorrModel *Checking Correlation Model*

Description

The procedure controls if the correlation model inserted is correct.

Usage

```
CkCorrModel(corrmodel)
```

Arguments

`corrmodel`	String; the name of a correlation model, for the description see `GeoCovmatrix`.

Details

The procedure controls if the correlation model is correct

Value

Return a number associated to a given correlation model if the model is considered in the package. Otherwise return NULL.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

CkInput *Checking Input*

Description

Subroutine called by the fitting procedures. The procedure controls the the validity of the input inserted by the users.

Usage

```
CkInput(coordx, coordy, coordt, coordx_dyn,corrmodel, data, distance,
           fcall, fixed, grid,likelihood, maxdist, maxtime,
            model, n,  optimizer, param, radius,
           start, taper, tapsep,  type, varest, vartype,
           weighted,X)
```

Arguments

`coordx`	A numeric ($d \times 2$)-matrix (where `d` is the number of points) assigning 2-dimensions of coordinates or a numeric vector assigning 1-dimension of coordinates.
`coordy`	A numeric vector assigning 1-dimension of coordinates; `coordy` is interpreted only if `coordx` is a numeric vector otherwise it will be ignored.
`coordt`	A numeric vector assigning 1-dimension of temporal coordinates.
`corrmodel`	String; the name of a correlation model, for the description see `GeoFit`.
`coordx_dyn`	A list of m numeric ($d_t \times 2$)-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is `NULL`
`data`	A numeric vector or a ($n \times d$)-matrix or ($d \times d \times n$)-matrix of observations.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details**.
`fcall`	String; `Fitting` to call the fitting procedure and `simulation` to call the simulation.
`fixed`	A named list giving the values of the parameters that will be considered as known values. The listed parameters for a given correlation function will be not estimated, i.e. if `list(nugget=0)` the nugget effect is ignored.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as a vector or a ($n \times d$)-matrix, instead if `TRUE` then ($d \times d \times n$)-matrix is considered.
`likelihood`	String; the configuration of the composite likelihood. `Marginal` is the default.
`maxdist`	Numeric; an optional positive value indicating the maximum spatial distance considered in the composite-likelihood computation.
`maxtime`	Numeric; an optional positive value indicating the maximum temporal lag separation in the composite-likelihood.
`radius`	Numeric; the radius of the sphere in the case of lon-lat coordinates. The default is 6371, the radius of the earth.
`model`	String; the density associated to the likelihood objects. `Gaussian` is the default.
`n`	Numeric; the number of trials in a binomial random fields. Default is 1.

optimizer	String; the optimization algorithm (see optim for details). 'Nelder-Mead' is the default.
param	A numeric vector of parameters, needed only in simulation. See GeoSim.
start	A named list with the initial values of the parameters that are used by the numerical routines in maximization procedure. NULL is the default.
taper	String; the name of the tapered correlation function.
tapsep	Numeric; an optional value indicating the separabe parameter in the space time quasi taper (see **Details**).
type	String; the type of the likelihood objects. If Pairwise (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods.
varest	Logical; if TRUE the estimate' variances and standard errors are returned. FALSE is the default.
vartype	String; the type of estimation method for computing the estimate variances, see GeoFit.
weighted	Logical; if TRUE the likelihood objects are weighted. If FALSE (the default) the composite likelihood is not weighted.
X	Numeric; Matrix of space-time covariates in the linear mean specification.

Details

Subroutine called by the fitting procedures. The procedure controls the the validity of the input inserted by the users.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit

CkLikelihood *Checking Composite-likelihood Type*

Description

Subroutine called by InitParam. The procedure controls the type of the composite-likelihood inserted by the users.

Usage

```
CkLikelihood(likelihood)
```

Arguments

likelihood	String; the configuration of the composite likelihood. Marginal is the default.

Details

The function controls the type of the composite-likelihood inserted by the users.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit

CkModel *Checking Random Field type*

Description

Subroutine called by InitParam. The procedure controls the type of random field inserted by the users.

Usage

```
CkModel(model)
```

Arguments

`model`	String; the density associated to the likelihood objects. `Gaussian` is the default.

Details

The function controls the type of random field inserted by the users.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit

CkType *Checking Likelihood Objects*

Description

Subroutine called by InitParam. The procedure controls the type of likelihood objects inserted by the users.

Usage

```
CkType(type)
```

Arguments

type	String; the type of the likelihood objects. If `Pairwise` (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods.

Details

The procedure checks the likelihood Object

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit

CkVarType *Checking Variance Estimates Type*

Description

Subroutine called by InitParam. The procedure controls the method used to compute the estimates' variances.

Usage

```
CkVarType(type)
```

Arguments

type	String; the method used to compute the estimates' variances. If `SubSamp` the estimates' variances are computed by the sub-sampling method, see GeoFit.

Details

The procedure controls the method used to compute the estimates' variances

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit

CompLik *Optimizes the Composite log-likelihood*

Description

Subroutine called by GeoFit. The procedure estimates the model parameters by maximisation of the composite log-likelihood.

Usage

```
CompLik(bivariate, coordx, coordy ,coordt, coordx_dyn, corrmodel, data, distance,
                flagcorr, flagnuis, fixed, GPU, grid,likelihood,local,lower,
                  model, n, namescorr, namesnuis, namesparam,
                numparam, numparamcorr, optimizer, onlyvar, parallel, param,
                spacetime, type,upper, varest, vartype,
              weigthed, winconst, winstp,winconst_t, winstp_t, ns, X,sensitivity)
```

Arguments

bivariate	Logical; if TRUE then the data come froma a bivariate random field. Otherwise from a univariate random field.
coordx	A numeric ($d{\times}2$)-matrix (where d is the number of points) assigning 2-dimensions of coordinates or a numeric vector assigning 1-dimension of coordinates.
coordy	A numeric vector assigning 1-dimension of coordinates; coordy is interpreted only if coordx is a numeric vector otherwise it will be ignored.
coordt	A numeric vector assigning 1-dimension of temporal coordinates. Optional argument, the default is NULL then a spatial random field is expected.
coordx_dyn	A list of m numeric ($d_t \times 2$)-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is NULL
corrmodel	Numeric; the id of the correlation model.
data	A numeric vector or a ($n \times d$)-matrix or ($d \times d \times n$)-matrix of observations.
distance	String; the name of the spatial distance. The default is Eucl, the euclidean distance. See the Section **Details**.
flagcorr	A numeric vector of binary values denoting which paramerters of the correlation function will be estimated.
flagnuis	A numeric vector of binary values denoting which nuisance paramerters will be estimated.
fixed	A numeric vector of parameters that will be considered as known values.
GPU	Numeric; if NULL (the default) no GPU computation is performed.

`grid`	Logical; if `FALSE` (the default) the data are interpreted as a vector or a $(n \times d)$-matrix, instead if `TRUE` then $(d \times d \times n)$-matrix is considered.
`likelihood`	String; the configuration of the compositelikelihood, see `GeoFit`.
`local`	Numeric; number of local work-items of the GPU
`lower`	An optional named list giving the values for the lower bound of the space parameter when the optimizer is `L-BFGS-B` or `nlminb` or `optimize`. The names of the list must be the same of the names in the `start` list.
`model`	Numeric; the id value of the density associated to the likelihood objects.
`n`	Numeric; number of trials in a binomial random fields.
`namescorr`	String; the names of the correlation parameters.
`namesnuis`	String; the names of the nuisance parameters.
`namesparam`	String; the names of the parameters to be maximised.
`numparam`	Numeric; the number of parameters to be maximised.
`numparamcorr`	Numeric; the number of correlation parameters.
`optimizer`	String; the optimization algorithm (see `optim` for details). `Nelder-Mead` is the default. Other possible choices are `ucminf`,`nlm`, `BFGS` `L-BFGS-B` and `nlminb`. In these last two cases upper and lower bounds can be passed by the user. In the case of one-dimensional optimization, the function `optimize` is used.
`onlyvar`	Logical; if `TRUE` (and varest is `TRUE`) only the variance covariance matrix is computed without optimizing. `FALSE` is the default.
`parallel`	Logical; if `TRUE` optmization is performed using optimParallel using the maximum number of cores, when optimizer is `L-BFGS-B`.`FALSE` is the default.
`param`	A numeric vector of parameters' values.
`spacetime`	Logical; if `TRUE` the random field is spatial-temporal otherwise is a spatial field.
`type`	String; the type of the likelihood objects. If `Pairwise` (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods.
`upper`	An optional named list giving the values for the upper bound of the space parameter when the optimizer is or `L-BFGS-B` or `nlminb` or `optimize`. The names of the list must be the same of the names in the `start` list.
`varest`	Logical; if `TRUE` the estimate' variances and standard errors are returned. `FALSE` is the default.
`vartype`	String; the type of estimation method for computing the estimate variances, see `GeoFit`.
`weigthed`	Logical; if `TRUE` then decreasing weigths coming from a compactly supported correlation function with compact support `maxdist` (`maxtime`)are used.
`winconst`	Numeric; a positive value for computing the spatial sub-window in the sub-sampling procedure.
`winstp`	Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the spatial sub-sampling procedure.
`winconst_t`	Numeric; a positive value for computing the temporal sub-window in the sub-sampling procedure.
`winstp_t`	Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the temporal sub-sampling procedure.
`ns`	Numeric; Number of (dynamical) temporal instants.
`X`	Numeric; Matrix of space-time covariates in the linear mean specification.
`sensitivity`	Logical; if `TRUE` then the sensitivy matrix is computed

Details

Subroutine called by GeoFit. The procedure estimates the model parameters by maximisation of the composite log-likelihood

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit

CompLik2 *Optimizes the Composite log-likelihood*

Description

Subroutine called by GeoFit. The procedure estimates the model parameters by maximisation of the composite log-likelihood.

Usage

```
CompLik2(bivariate, coordx, coordy ,coordt,coordx_dyn,corrmodel, data, distance, flagcorr, flagnui
      fixed, GPU,grid,likelihood, local,lower, model, n, namescorr, namesnuis, namesparam,
      numparam, numparamcorr, optimizer, onlyvar, parallel, param, spacetime, type,
      upper, varest, vartype, weigthed, winconst, winstp,winconst_t, winstp_t, ns, X,sensitivity,
         colidx,rowidx,neighb)
```

Arguments

`bivariate`	Logical; if `TRUE` then the data come froma a bivariate random field. Otherwise from a univariate random field.
`coordx`	A numeric (d×2)-matrix (where d is the number of points) assigning 2-dimensions of coordinates or a numeric vector assigning 1-dimension of coordinates.
`coordy`	A numeric vector assigning 1-dimension of coordinates; `coordy` is interpreted only if `coordx` is a numeric vector otherwise it will be ignored.
`coordt`	A numeric vector assigning 1-dimension of temporal coordinates. Optional argument, the default is `NULL` then a spatial random field is expected.
`coordx_dyn`	A list of m numeric $(d_t \times 2)$-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is `NULL`
`corrmodel`	Numeric; the id of the correlation model.
`data`	A numeric vector or a $(n \times d)$-matrix or $(d \times d \times n)$-matrix of observations.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details**.
`flagcorr`	A numeric vector of binary values denoting which paramerters of the correlation function will be estimated.

flagnuis	A numeric vector of binary values denoting which nuisance paramerters will be estimated.
fixed	A numeric vector of parameters that will be considered as known values.
GPU	Numeric; if NULL (the default) no GPU computation is performed.
grid	Logical; if FALSE (the default) the data are interpreted as a vector or a $(n \times d)$-matrix, instead if TRUE then $(d \times d \times n)$-matrix is considered.
likelihood	String; the configuration of the compositelikelihood, see GeoFit.
local	Numeric; number of local work-items of the GPU
lower	An optional named list giving the values for the lower bound of the space parameter when the optimizer is L-BFGS-B or nlminb or optimize. The names of the list must be the same of the names in the start list.
model	Numeric; the id value of the density associated to the likelihood objects.
n	Numeric; number of trials in a binomial random fields.
namescorr	String; the names of the correlation parameters.
namesnuis	String; the names of the nuisance parameters.
namesparam	String; the names of the parameters to be maximised.
numparam	Numeric; the number of parameters to be maximised.
numparamcorr	Numeric; the number of correlation parameters.
optimizer	String; the optimization algorithm (see optim for details). Nelder-Mead is the default. Other possible choices are ucminf,nlm, BFGS L-BFGS-B and nlminb. In these last two cases upper and lower bounds can be passed by the user. In the case of one-dimensional optimization, the function optimize is used.
onlyvar	Logical; if TRUE (and varest is TRUE) only the variance covariance matrix is computed without optimizing. FALSE is the default.
parallel	Logical; if TRUE optmization is performed using optimParallel using the maximum number of cores, when optimizer is L-BFGS-B.FALSE is the default.
param	A numeric vector of parameters' values.
spacetime	Logical; if TRUE the random field is spatial-temporal otherwise is a spatial field.
type	String; the type of the likelihood objects. If Pairwise (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods.
upper	An optional named list giving the values for the upper bound of the space parameter when the optimizer is or L-BFGS-B or nlminb or optimize. The names of the list must be the same of the names in the start list.
varest	Logical; if TRUE the estimate' variances and standard errors are returned. FALSE is the default.
vartype	String; the type of estimation method for computing the estimate variances, see GeoFit.
weigthed	Logical; if TRUE then decreasing weigths coming from a compactly supported correlation function with compact support maxdist (maxtime)are used.
winconst	Numeric; a positive value for computing the spatial sub-window in the subsampling procedure.
winstp	Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the spatial sub-sampling procedure.
winconst_t	Numeric; a positive value for computing the temporal sub-window in the subsampling procedure.

winstp_t	Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the temporal sub-sampling procedure.
ns	Numeric; Number of (dynamical) temporal instants.
X	Numeric; Matrix of space-time covariates in the linear mean specification.
sensitivity	Logical; if TRUE then the sensitivy matrix is computed
colidx	Numeric; Vector of indexes for spatial distances.
rowidx	Numeric; Vector of indexes for spatial distances.
neighb	Numeric; an optional positive integer indicating the order of neighborhood location.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit

CorrelationPar *Lists the Parameters of a Correlation Model*

Description

Subroutine called by InitParam and other procedures. The procedure returns a list with the parameters of a given correlation model.

Usage

```
CorrelationPar(corrmodel)
```

Arguments

corrmodel	Integer; an integer associated to a given correlation model.

Details

The function return a list with the Parameters of a Correlation Model

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit

CorrParam *Lists the Parameters of a Correlation Model*

Description

The procedure returns a list with the parameters of a given correlation model.

Usage

```
CorrParam(corrmodel)
```

Arguments

corrmodel	String; the name of a correlation model.

Details

The function return a list with the Parameters of a Correlation Model

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoCovmatrix

Examples

```
require(GeoModels)
################################################################
###
### Example 1. Parameters of the  Matern  model
###
###############################################################

CorrParam("Matern")


################################################################
###
### Example 2. Parameters of the  Generalized Wendland  model
###
###############################################################

CorrParam("GenWend")


################################################################
###
### Example 3. Parameters of the  Generalized Wendland  model
###
###############################################################
```

```
CorrParam("GenCauchy")


################################################################
###
### Example 4. Parameters of the  space time Gneiting  model
###
###############################################################

CorrParam("Gneiting")


################################################################
###
### Example 5. Parameters of the bi-Matern separable  model
###
###############################################################

CorrParam("Bi_Matern_sep")
```

DeviceInfo *Prints Device Information*

Description

Prints the device details available in your computer. Device name, Max compute units, whether it supports double precision, among others.

Usage

```
DeviceInfo()
```

Details

The user can take this information into account so that the `local` parameter is set up in `GeoFit` when GPU computation is chosen.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

Examples

```
library(GeoModels)
DeviceInfo()
```

GeoCovariogram *Computes the fitted variogram model.*

Description

The procedure computes and plots covariance or variogram estimated fitting a Gaussian, and non Gaussian spatio (temporal) bivariate random fields. Allows to add the empirical estimates in order to compare them with the fitted model.

Usage

```
GeoCovariogram(fitted, distance="Eucl",answer.cov=FALSE,
            answer.vario=FALSE, answer.range=FALSE, fix.lags=NULL,
             fix.lagt=NULL, show.cov=FALSE, show.vario=FALSE,
            show.range=FALSE, add.cov=FALSE, add.vario=FALSE,
            pract.range=95, vario, ...)
```

Arguments

fitted	A fitted object obtained from the GeoFit or GeoWLS procedures.
distance	String; the name of the spatial distance. The default is Eucl, the euclidean distance. See GeoFit.
answer.cov	Logical; if TRUE a vector with the estimated covariance function is returned; if FALSE (the default) the covariance is not returned.
answer.vario	Logical; if TRUE a vector with the estimated variogram is returned; if FALSE (the default) the variogram is not returned.
answer.range	Logical; if TRUE the estimated pratical range is returned; if FALSE (the default) the pratical range is not returned.
fix.lags	Integer; a positive value denoting the spatial lag to consider for the plot of the temporal profile.
fix.lagt	Integer; a positive value denoting the temporal lag to consider for the plot of the spatial profile.
show.cov	Logical; if TRUE the estimated covariance function is plotted; if FALSE (the default) the covariance function is not plotted.
show.vario	Logical; if TRUE the estimated variogram is plotted; if FALSE (the default) the variogram is not plotted.
show.range	Logical; if TRUE the estimated pratical range is added on the plot; if FALSE (the default) the pratical range is not added.
add.cov	Logical; if TRUE the vector of the estimated covariance function is added on the current plot; if FALSE (the default) the covariance is not added.
add.vario	Logical; if TRUE the vector with the estimated variogram is added on the current plot; if FALSE (the default) the correlation is not added.
pract.range	Numeric; the percent of the sill to be reached.
vario	A Variogram object obtained from the GeoVariogram procedure.
...	other optional parameters which are passed to plot functions.

Details

The function computes the fitted variogram model

Value

The returned object is eventually a list with:

- `covariance` The vector of the estimated covariance function;
- `variogram` The vector of the estimated variogram function;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

References

Cressie, N. A. C. (1993) *Statistics for Spatial Data*. New York: Wiley.

Gaetan, C. and Guyon, X. (2010) Spatial Statistics and Modelling. *Spring Verlang, New York*.

See Also

GeoFit.

Examples

```
library(GeoModels)
library(scatterplot3d)

################################################################
###
### Example 1. Plot of fitted covariance and fitted
### and empirical variograms from a  Gaussian RF
### with Matern correlation.
###
###############################################################
set.seed(21)
# Set the coordinates of the points:
x = runif(300, 0, 1)
y = runif(300, 0, 1)
coords=cbind(x,y)

# Set the model's parameters:
corrmodel = "Matern"
model = "Gaussian"
mean = 0
sill = 1
nugget = 0
scale = 0.2/3
smooth=0.5

param=list(mean=mean,sill=sill, nugget=nugget, scale=scale, smooth=smooth)
# Simulation of the Gaussian random field:
data = GeoSim(coordx=coords, corrmodel=corrmodel, model=model,param=param)$data
```

```
start=list(mean=0,scale=scale,sill=sill)
fixed=list(nugget=nugget,smooth=smooth)
# Maximum composite-likelihood fitting of the Gaussian random field:
fit = GeoFit(data=data,coordx=coords, corrmodel=corrmodel,model=model,
            likelihood="Marginal",type='Pairwise',start=start,
            optimizer="BFGS", fixed=fixed,maxdist=0.05)

# Empirical estimation of the variogram:
vario = GeoVariogram(data=data,coordx=coords,maxdist=0.5)

# Plot of covariance and variogram functions:
GeoCovariogram(fit, show.cov=TRUE,show.vario=TRUE, vario=vario,pch=20)

################################################################
###
### Example 2. Plot of fitted covariance and fitted
### and empirical variograms from a  Binomial
###  RF with exponential correlation.
###
###############################################################
set.seed(2111)

model="Binomial";n=20
# Set the coordinates of the points:
x = runif(500, 0, 1)
y = runif(500, 0, 1)
coords=cbind(x,y)

# Set the model's parameters:
corrmodel = "exponential"
mean = 0
sill = 1
nugget = 0
scale = 0.2/3

param=list(mean=mean,sill=sill, nugget=nugget, scale=scale)
# Simulation of the Gaussian RF:
data = GeoSim(coordx=coords, corrmodel=corrmodel, model=model,param=param,n=n)$data

start=list(mean=0,scale=scale,sill=sill)
fixed=list(nugget=nugget)
# Maximum composite-likelihood fitting of the BinomGaussian random field:
fit = GeoFit(data,coordx=coords, corrmodel=corrmodel,model=model,
             likelihood="Marginal",type='Pairwise',start=start,n=n,
             optimizer="BFGS", fixed=fixed,maxdist=0.03)

# Empirical estimation of the variogram:
vario = GeoVariogram(data,coordx=coords,maxdist=0.5)

# Plot of covariance and variogram functions:
GeoCovariogram(fit, show.cov=TRUE,show.vario=TRUE, vario=vario,pch=20)


################################################################
###
### Example 3. Plot of fitted covariance and fitted
### and empirical variograms from a  RF
```

```
###  RF with Wend0 correlation.
###
###############################################################
set.seed(211)

model="Gamma";shape=4
# Set the coordinates of the points:
x = runif(700, 0, 1)
y = runif(700, 0, 1)
coords=cbind(x,y)

# Set the model's parameters:
corrmodel = "Wend0"
mean = 0
sill = 1
nugget = 0
scale = 0.3
power2=4


param=list(mean=mean,sill=sill, nugget=nugget, scale=scale,shape=shape,power2=power2)
# Simulation of the Gaussian RF:
data = GeoSim(coordx=coords, corrmodel=corrmodel, model=model,param=param)$data

start=list(mean=0,scale=scale,shape=shape)
fixed=list(nugget=nugget,sill=sill,power2=power2)
# Maximum composite-likelihood fitting of the BinomGaussian random field:
fit = GeoFit(data,coordx=coords, corrmodel=corrmodel,model=model,
             likelihood="Marginal",type='Pairwise',start=start,
             optimizer="BFGS", fixed=fixed,maxdist=0.03)

# Empirical estimation of the variogram:
vario = GeoVariogram(data,coordx=coords,maxdist=0.5)

# Plot of covariance and variogram functions:
GeoCovariogram(fit, show.cov=TRUE,show.vario=TRUE, vario=vario,pch=20)

################################################################
###
### Example 4. Plot of  fitted  and empirical variograms
### from a space time Gaussian random fields
### with double exponential correlation.
###
###############################################################
set.seed(92)
# Define the spatial-coordinates of the points:
x = runif(50, 0, 1)
y = runif(50, 0, 1)
coords=cbind(x,y)
# Define the temporal sequence:
time = seq(0, 15, 1)


# Simulation of the spatio-temporal Gaussian random field:
data = GeoSim(coordx=coords, coordt=time, corrmodel="Exp_Exp",param=list(mean=mean,
              nugget=nugget,scale_s=0.5/3,scale_t=2/2,sill=sill))$data
```

```
fixed=list(nugget=0, mean=0)
start=list(scale_s=0.2, scale_t=0.5, sill=1)
# Maximum composite-likelihood fitting of the space-time Gaussian random field:
fit = GeoFit(data, coordx=coords, coordt=time, corrmodel="Exp_Exp", maxtime=2,
             maxdist=0.1, likelihood="Marginal", type="Pairwise",
             optimizer="BFGS",fixed=fixed, start=start)

# Empirical estimation of spatio-temporal covariance:
vario = GeoVariogram(data,coordx=coords, coordt=time, maxtime=5,maxdist=0.5)

# Plot of the fitted space-time variogram
GeoCovariogram(fit,vario=vario,show.vario=TRUE)

# Plot of covariance, variogram and spatio and temporal profiles:
GeoCovariogram(fit,vario=vario,fix.lagt=1,fix.lags=1,show.vario=TRUE,pch=20)



################################################################
###
### Example 5. Plot of parametric and empirical variograms
### estimated from a Bivariate Gaussian random fields with
### Matern correlation.
###
###############################################################

# Simulation of a bivariate spatial Gaussian random field:
set.seed(892)
# Define the spatial-coordinates of the points:
x = runif(200, -1, 1)
y = runif(200, -1, 1)
coords=cbind(x,y)

# Simulation of a bivariate Gaussian Random field
# with matern (cross)  covariance function
scale_1 = 0.25/3
scale_2 = 0.2/3
scale_12 = 0.15/3
sill_1=1
sill_2=1
smooth=0.5
pcol=0.3
param=list(mean_1=0,mean_2=0,scale_1=scale_1,scale_2=scale_2,scale_12=scale_12,
           sill_1=sill_1,sill_2=sill_2,nugget_1=0,nugget_2=0,
           smooth_1=smooth,smooth_12=smooth,smooth_2=smooth,pcol=pcol)
data = GeoSim(coordx=coords, corrmodel="Bi_Matern", param=param)$data

# Empirical bivariate variogram estimation:
biv_vario=GeoVariogram(data,coordx=coords, bivariate=TRUE,maxdist=c(1,1,1))

# selecting fixed and estimating parameters
fixed=list(mean_1=0,mean_2=0,nugget_1=0,nugget_2=0,
          smooth_1=smooth,smooth_12=smooth,smooth_2=smooth)
start=list(sill_1=var(data[1,]),sill_2=var(data[2,]),
          scale_1=scale_1,scale_2=scale_2,scale_12=scale_12,
          pcol=cor(data[1,],data[2,]))
```

```
# Maximum likelihood fitting of the bivariate random field:
fit= GeoFit(data, coordx=coords, corrmodel="Bi_Matern",likelihood="Marginal",
            optimizer="BFGS", type="Pairwise",
            start=start,fixed=fixed,maxdist=c(0.1,0.1,0.1))

GeoCovariogram(fit, vario=biv_vario,show.vario=TRUE,pch=20)
```

GeoCovDisplay — *Image plot displaying the pattern of the sparsness of a covariance matrix.*

Description

Image plot displaying the pattern of the sparsness of a covariance matrix.

Usage

```
GeoCovDisplay(covmatrix,limits=FALSE,pch=2)
```

Arguments

covmatrix	An object of class matrix. See the Section **Details**.
limits	Logical; If TRUE and the covariance matrix is spatiotemporal or spatial bivariate then vertical and horizontal lines are added to the image plot.
pch	Type of symbols to use in the image plot.

Details

For a given covariance matrix object (GeoCovmatrix) the function diplays the pattern of the sparsness of a covariance matrix where the white color represents 0 entries and black color represents non zero entries

Value

Returns an image plot.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoCovmatrix

Examples

```
library(GeoModels)


  # Define the spatial-coordinates of the points:
x <- runif(100, 0, 2)
y <- runif(100, 0, 2)
coords=cbind(x,y)
matrix1 <- GeoCovmatrix(coordx=coords, corrmodel="GenWend", param=list(smooth=0,
                      power2=4,sill=1,scale=0.2,nugget=0))

GeoCovDisplay(matrix1)
```

`GeoCovmatrix` *Spatial and Spatio-temporal Covariance Matrix of (non) Gaussian random fields*

Description

The function computes the covariance matrix associated to a spatial or spatio-temporal or a bivariate spatial Gaussian or non Gaussian randomm field with given covariance model and a set of spatial location sites and temporal instants.

Usage

```
GeoCovmatrix(coordx, coordy=NULL, coordt=NULL, coordx_dyn=NULL, corrmodel,
          distance="Eucl", grid=FALSE, maxdist=NULL, maxtime=NULL,
          model="Gaussian", n=1, param, radius=6371, sparse=FALSE,
          taper=NULL, tapsep=NULL, type="Standard",X=NULL)
```

Arguments

`coordx` A numeric $(d \times 2)$-matrix (where `d` is the number of spatial sites) giving 2-dimensions of spatial coordinates or a numeric d-dimensional vector giving 1-dimension of spatial coordinates. Coordinates on a sphere for a fixed radius `radius` are passed in lon/lat format expressed in decimal degrees.

`coordy` A numeric vector giving 1-dimension of spatial coordinates; `coordy` is interpreted only if `coordx` is a numeric vector or `grid=TRUE` otherwise it will be ignored. Optional argument, the default is `NULL` then `coordx` is expected to be numeric a $(d \times 2)$-matrix.

`coordt` A numeric vector giving 1-dimension of temporal coordinates. At the moment implemented only for the Gaussian case. Optional argument, the default is `NULL` then a spatial random field is expected.

`coordx_dyn` A list of T numeric $(d_t \times 2)$-matrices containing dynamical (in time) coordinates. Optional argument, the default is `NULL`

`corrmodel` String; the name of a correlation model, for the description see the Section **Details**.

`distance` String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See `GeoFit`.

`grid`	Logical; if `FALSE` (the default) the data are interpreted as spatial or spatial-temporal realisations on a set of non-equispaced spatial sites (irregular grid). See `GeoFit`.
`maxdist`	Numeric; an optional positive value indicating the marginal spatial compact support in the case of tapered covariance matrix. See `GeoFit`.
`maxtime`	Numeric; an optional positive value indicating the marginal temporal compact support in the case of spacetime tapered covariance matrix. See `GeoFit`.
`n`	Numeric; the number of trials in a binomial random fields. Default is 1.
`model`	String; the type of RF. See `GeoFit`.
`param`	A list of parameter values required for the covariance model.
`radius`	Numeric; a value indicating the radius of the sphere when using covariance models valid using the great circle distance. Default value is the radius of the earth in Km (i.e. 6371)
`sparse`	Logical; if `TRUE` the function return an object of class spam. This option should be used when a parametric compactly supporte covariance is used. Default is FALSE.
`taper`	String; the name of the taper correlation function if type is `Tapering`, see the Section **Details**.
`tapsep`	Numeric; an optional value indicating the separabe parameter in the space-time non separable taper or the colocated correlation parameter in a bivariate spatial taper (see **Details**).
`type`	String; the type of covariance matrix `Standard` (the default) or `Tapering` for tapered covariance matrix.
`X`	Numeric; Matrix of space-time covariates.

Details

In the spatial case, the covariance matrix of the random vector

$$[Z(s_1), \dots, Z(s_n,)]^T$$

with a specific spatial covariance model is computed. Here n is the number of the spatial location sites.

In the space-time case, the covariance matrix of the random vector

$$[Z(s_1,t_1), Z(s_2,t_1), \dots, Z(s_n,t_1), \dots, Z(s_n,t_m)]^T$$

with a specific space time covariance model is computed. Here m is the number of temporal instants.

In the bivariate case, the covariance matrix of the random vector

$$[Z_1(s_1), Z_2(s_1), \dots, Z_1(s_n), Z_2(s_n)]^T$$

with a specific spatial bivariate covariance model is computed.

The location site s_i can be a point in the d-dimensional euclidean space with $d = 2$ or a point (given in lon/lat degree format) on a sphere of arbitrary radius.

Here there is the list of all the implemented space and space-time and bivariate correlation models. The argument `param` is a list including all the parameters of a given correlation model specified by the argument `corrmodel`. For each correlation model one can check the associated correlation parameters using `CorrParam`. In what follows $\kappa > 0$, $\beta > 0$, $\alpha, \alpha_s, \alpha_t \in (0, 2]$, and $\gamma \in [0, 1]$. The associated parameters in the argument `param` are `smooth`, `power2`, `power`, `power_s`, `power_t` and `sep` respectively. Moreover let $1(A) = 1$ when A is true and 0 otherwise.

- Spatial correlation models:
  1. $Cauchy$ defined as:
  $$R(h) = \left(1 + h^2\right)^{-\beta/2}$$
  It is a special case of the `Gencauchy` model.
  2. Exp defined as:
  $$R(h) = e^{-h}$$
  This model is a special case of the `Matern` and the `Stable` model.
  3. $GenCauchy$ (generalised $Cauchy$) defined as:
  $$R(h) = (1 + h^{\alpha})^{-\beta/\alpha}$$
  If h is the geodesic distance then $\alpha \in (0, 1]$.
  4. $Matern$ defined as:
  $$R(h) = 2^{1-\kappa}\Gamma(\kappa)^{-1}h^{\kappa}K_{\kappa}(h)$$
  If h is the geodesic distance then $\kappa \in (0, 0.5]$
  5. $Stable$ defined as:
  $$R(h) = e^{-h^{\alpha}}$$
  If h is the geodesic distance then $\alpha \in (0, 1]$.
  6. $Wave$ defined as:
  $$R(h) = sin(h)/h$$
  This model is valid only for dimensions less than or equal to 3.
  7. $Wend0$ defined as:
  $$R(h) = (1 - h)^{\mu}1(h \in [0, 1])$$
  where $\mu \geq 0.5(d + 1)$. If h is the geodesic distance then $\mu \geq 2$.
  8. $Wend1$ defined as:
  $$R(h) = (1 - h)^{\mu+1}(1 + (\mu + 1)h)1(h \in [0, 1])$$
  where $\mu \geq 0.5(d + 1) + 1$. If h is the geodesic distance then $\mu \geq 4$.
  9. $Wend2$ defined as:
  $$R(h) = (1 - h)^{\mu+2}(1 + (\mu + 2)h + (1/3)((\mu + 1)^2 - 1)h^2)1(h \in [0, 1])$$
  where $\mu \geq 0.5(d + 1) + 2$. If h is the geodesic distance then $\mu \geq 6$.
  10. $GenWend$ (Generalized Wendland) defined as:
  $$R(h) = \int_h^1 [(1 - x)^{\mu-1}(x^2 - h^2)^{\kappa-1}1(h \in [0, 1])]dx/B(2\kappa + 1, \mu)$$
  where $\mu \geq 0.5(d + 1) + \kappa$. The cases $\kappa = 0, 1, 2$ correspond to the $Wend0$, $Wend1$ and $Wend2$ respectively.
  11. $GenWendMatern$ (Generalized Wendland Matern) defined as:
  $$R(h) = \int_h^1 [(1 - x/a)^{\mu-1}((x/a)^2 - h^2)^{\kappa-1}1(h \in [0, a])]dx/B(2\kappa + 1, \mu)$$
  where $\mu \geq 0.5(d + 1) + \kappa$ and $a = (\Gamma(\mu + 1 + 2\kappa)/\Gamma(\mu))^(1/(1 + 2\kappa))$ The inverse parametrization is used for the μ parameter.

12. $Multiquadric$ defined as:

$$R(h) = (1 - \alpha 0.5)^{2\beta}/(1 + (\alpha 0.5)^2 - \alpha cos(h))^{\beta}, \quad h \in [0, \pi]$$

This model is valid on the unit sphere and h is the geodesic distance.

13. $Sinpower$ defined as:

$$R(h) = 1 - (sin(h/2))^{\alpha}, \quad h \in [0, \pi]$$

This model is valid on the unit sphere and h is the geodesic distance.

14. $Smoke$ defined as:

$$R(h) = K * 1F2(1/\alpha, 1/\alpha + 0.5, 2/\alpha + 0.5 + \kappa), \quad h \in [0, \pi]$$

where $K = (\Gamma(a)\Gamma(i))/\Gamma(i)\Gamma(o))$. This model is valid on the unit sphere and h is the geodesic distance.

- Spatio-temporal correlation models.
  - Non-separable models:
    1. $Gneiting$ defined as:

    $$R(h, u) = e^{-h^{\alpha_s}/((1+u^{\alpha_t})^{0.5\gamma\alpha_s})}/(1 + u^{\alpha_t})$$

    2. $Gneiting\_GC$

    $$R(h, u) = e^{-u^{\alpha_t}/((1+h^{\alpha_s})^{0.5\gamma\alpha_t})}/(1 + h^{\alpha_s})$$

    where h can be both the euclidean and the geodesic distance

    3. $Iacocesare$

    $$R(h, u) = (1 + h^{\alpha_s} + u_t^{\alpha})^{-\beta}$$

    4. $Porcu$

    $$R(h, u) = (0.5(1 + h^{\alpha_s})^{\gamma} + 0.5(1 + u^{\alpha_t})^{\gamma})^{-\gamma^{-1}}$$

    5. $Porcu1$

    $$R(h, u) = (e^{-h^{\alpha_s}(1+u^{\alpha_t})^{0.5\gamma\alpha_s}})/((1 + u^{\alpha_t})^{1.5})$$

    6. $Stein$

    $$R(h, u) = (h^{\psi(u)} K_{\psi(u)}(h))/(2^{\psi(u)}\Gamma(\psi(u) + 1))$$

    where $\psi(u) = \nu + u^{0.5\alpha_t}$

    7. $Wenx\_space$, $x = 0, 1, 2$ defined as:

    $$R(h, u) = \phi(u)^{3.5+2x} Wenx(h/\phi(u), \mu_s), \quad x = 0, 1, 2$$

    where $\phi(u) = (1 + u^{0.5\alpha_t})^{-\gamma}$, $0 < \gamma \leq \alpha_t/2$, $\mu_s \geq 0.5(d + 5) + x$.

    8. $Wenx\_time$, $x = 0, 1, 2$ defined as:

    $$R(h, u) = \phi(h)^{3.5+2x} Wenx(u/\phi(h); \mu_t), \quad x = 0, 1, 2$$

    where $\phi(h) = (1 + h^{0.5\alpha_s})^{-\gamma}$, $0 < \gamma \leq \alpha_s/2$, $\mu_t \geq 0.5(d + 5) + x$.

    9. $Multiquadric\_st$ defined as:

    $$R(h, u) = ((1 - 0.5\alpha_s)^2/(1 + (0.5\alpha_s)^2 - \alpha_s\psi(u)cos(h)))^{a_s}, \quad h \in [0, \pi]$$

    where $\psi(u) = (1 + (u/a_t)^{\alpha_t})^{-1}$. This model is valid on the unit sphere and h is the geodesic distance.

10. $Sinpower\_st$ defined as:

$$R(h,u) = (e^{\alpha_s cos(h)\psi(u)/a_s}(1+\alpha_s cos(h)\psi(u)/a_s))/k$$

where $\psi(u) = (1+(u/a_t)^{\alpha_t})^{-1}$ and $k = (1+\alpha_s/a_s)exp(\alpha_s/a_s), \quad h \in [0,\pi]$ This model is valid on the unit sphere and h is the geodesic distance.

– Separable models.

Space-time separable correlation models are easly obtained as the product of a spatial and a temporal correlation model, that is

$$R(h,u) = R(h)R(u)$$

Several combinations are possible:

1. Exp_Exp defined as:
$$R(h,u) = Exp(h)Exp(u)$$
2. $Matern\_Matern$ defined as:
$$R(h,u) = Matern(h;\kappa_s)Matern(u;\kappa_t)$$
3. $Stable\_Stable$ defined as:
$$R(h,u) = Stable(h;\alpha_s)Stable(u;\alpha_t)$$
4. $Wendx\_Wendy$ defined as
$$R(h,u) = Wendx(h;\mu_s)Wendy(u;\mu_t), x,y = 0,1,2$$
.

Note that some models are nested. (The Exp_Exp with $Matern\_Matern$ for instance.)

- Spatial bivariate correlation models (see below):
  1. Bi_Matern (Bivariate full Matern model)
  2. Bi_Matern_contr (Bivariate Matern model with contrainsts)
  3. Bi_Matern_sep (Bivariate separable Matern model)
  4. Bi_LMC (Bivariate linear model of coregionalization)
  5. Bi_LMC_contr (Bivariate linear model of coregionalization with constraints)
  6. Bi_Wendx (Bivariate full Wendland model)
  7. Bi_Wendx_contr (Bivariate Wendland model with contrainsts)
  8. Bi_Wendx_sep (Bivariate separable Wendland model)
  9. Bi_Smoke (Bivariate full Smoke model on the unit sphere)
- Spatial taper.
  For spatial covariance tapering the taper functions are:
  1. $Bohman$ defined as:
  $$T(h) = (1-h)(sin(2\pi h)/(2\pi h)) + (1-cos(2\pi h))/(2\pi^2 h)1_{[0,1]}(h)$$
  2. $Wendlandx, \quad x = 0,1,2$ defined as:
  $$T(h) = Wendx(h;x+2), x = 0,1,2$$
  .

- Spatio-temporal tapers.
  For spacetime covariance tapering the taper functions are:
  1. $Wendlandx\_Wendlandy$ (Separable tapers) $x, y = 0, 1, 2$ defined as:
  $$T(h, u) = Wendx(h; x + 2)Wendy(h; y + 2), x, y = 0, 1, 2.$$
  2. $Wendlandx\_time$ (Non separable temporal taper) $x = 0, 1, 2$ defined as: $Wenx\_time$, $x = 0, 1, 2$ assuming $\alpha_t = 2$, $\mu_s = 3.5 + x$ and $\gamma \in [0, 1]$ to be fixed using `tapsep`.
  3. $Wendlandx\_space$ (Non separable spatial taper) $x = 0, 1, 2$ defined as: $Wenx\_space$, $x = 0, 1, 2$ assuming $\alpha_s = 2$, $\mu_t = 3.5 + x$ and $\gamma \in [0, 1]$ to be fixed using `tapsep`.
- Spatial bivariate taper (see below).
  1. $Bi\_Wendlandx, \quad x = 0, 1, 2$

Remarks:
In what follows we assume $\sigma^2, \sigma_1^2, \sigma_2^2, \tau^2, \tau_1^2, \tau_2^2, a, a_s, a_t, a_{11}, a_{22}, a_{12}, \kappa_{11}, \kappa_{22}, \kappa_{12}, f_{11}, f_{12}, f_{21}, f_{22}$ positive.

The associated parameters in `param` are `sill`, `sill_1`,`sill_2`, `nugget`, `nugget_1`,`nugget_2`, `scale`,`scale_s`,`scale_t`, `scale_1`,`scale_2`,`scale_12`, `smooth_1`,`smooth_2`,`smooth_12`, `a_1`,`a_12`,`a_21`,`a_2` respectively.

Let $R(h)$ be a spatial correlation model given in standard notation. Then the covariance model applied with arbitrary variance, nugget and scale equals to:

$$C(h) = (\sigma^2 + \tau^2 1(h = 0))R(h/a, , ...), \quad h \geq 0$$

Similarly if $R(h, u)$ is a spatio-temporal correlation model given in standard notation, then the covariance model is:

$$C(h, u) = (\sigma^2 + \tau^2 1(h = 0, u = 0))R(h/a_s, u/a_t, ...) \quad h \geq 0, u \geq 0$$

Here '...' stands for additional parameters.

Let $R(h)$ be a spatial taper given in standard notation. Then the taper function applied with an arbitrary compact support (d_s) equals to:

$$T(h) = R(h/d_s)$$

Then the tapered covariance function is given by:

$$C^{tap}(h) = T(h)C(h)$$

Similarly if $R(h, u)$ is a spatio-temporal taper given in standard notation, then the taper function applied with arbitrary compact supports $(d_s, d_t)^T$ equals to:

$$T(h, u) = R(h/d_s, u/d_t)$$

Then the tapered covariance function is given by:

$$C^{tap}(h, u) = T(h, u)C(h, u)$$

Compact supports d_s and d_t can be set by the user with `maxdist` and `maxtime`.

The bivariate models implemented are the following :

1. Bi_Matern defined as:

$$C_{ij}(h) = \rho_{ij}(\sigma_i\sigma_j + \tau_i^2 1(i=j, h=0))Matern(h/a_{ij}, \kappa_{ij}) \quad i,j = 1,2. \quad h \geq 0$$

   where $\rho = \rho_{12} = \rho_{21}$ is the correlation colocated parameter and $\rho_{ii} = 1$. The model Bi_Matern_sep (separable matern) is a special case when $a = a_{11} = a_{12} = a_{22}$ and $\kappa = \kappa_{11} = \kappa_{12} = \kappa_{22}$. The model Bi_Matern_contr (constrained matern) is a special case when $a_{12} = 0.5(a_{11} + a_{22})$ and $\kappa_{12} = 0.5(\kappa_{11} + \kappa_{22})$

2. Bi_Wendx ($x = 0, 1, 2$) defined as:

$$C_{ij}(h) = \rho_{ij}(\sigma_i\sigma_j + \tau_i^2 1(i=j, h=0))Wendx(h/a_{ij}, \nu_{ij} + 1) \quad i,j = 1,2. \quad h \geq 0$$

   where $\rho = \rho_{12} = \rho_{21}$ is the correlation colocated parameter and $\rho_{ii} = 1$. The model Bi_Wendx_sep (separable wendland) is a special case when $a = a_{11} = a_{12} = a_{22}$ and $\mu = \mu_{11} = \mu_{12} = \mu_{22}$. The model Bi_Wendx_contr (constrained matern) is a special case when $a_{12} = 0.5(a_{11} + a_{22})$ and $\mu_{12} = 0.5(\mu_{11} + \mu_{22})$

3. Bi_LMC defined as:

$$C_{ij}(h) = \sum_{k=1}^{2}(f_{ik}f_{jk} + \tau_i^2 1(i=j, h=0))R(h/a_k)$$

   where $R(h)$ is a correlation model. The model Bi_LMC_contr is a special case when $f = f_{12} = f_{21}$. Bivariate LMC models, in the current version of the package, is obtained with $R(h)$ equal to the exponential correlation model.

The bivariate spatial tapers implemented are the following :

1. $Bi\_Wendlandx$, $x = 0, 1, 2$ defined as:

$$T_{ij}(h) = r_{ij}Wendx(h/d_{ij}, x), \quad i,j = 1,2 \quad x = 0,1,2 \quad h \geq 0$$

   with $r_{ii} = 1$ and $r_{12} = r_{21}$ to be fixed using `tapsep`.

If $T_{ij}(h)$ is a bivariate taper, Then the tapered bivariate covariance function is given by:

$$C_{ij}^{tap}(h) = T_{ij}(h)C_{ij}(h)$$

Compact supports d_{11}, d_{12}, d_{22} can be set by the user with `maxdist`.

Value

Returns an object of class `CovMat`. An object of class `CovMat` is a list containing at most the following components:

`bivariate`	Logical:`TRUE` if the Gaussian random field is bivariaete otherwise `FALSE`;
`coordx`	A d-dimensional vector of spatial coordinates;
`coordy`	A d-dimensional vector of spatial coordinates;
`coordt`	A t-dimensional vector of temporal coordinates;
`coordx_dyn`	A list of t matrices of spatial coordinates;
`covmatrix`	The covariance matrix if `type` is`Standard`. An object of class spam if `type` is `Tapering` or `Standard` and `sparse` is TRUE.
`corrmodel`	String: the correlation model;
`distance`	String: the type of spatial distance;

grid	Logical:TRUE if the spatial data are in a regular grid, otherwise FALSE;
nozero	In the case of tapered matrix the percentage of non zero values in the covariance matrix. Otherwise is NULL.
maxdist	Numeric: the marginal spatial compact support if type is Tapering;
maxtime	Numeric: the marginal temporal compact support if type is Tapering;
n	The number of trial for Binomial RFs
namescorr	String: The names of the correlation parameters;
numcoord	Numeric: the number of spatial coordinates;
numtime	Numeric: the number the temporal coordinates;
model	The type of RF, see GeoFit.
param	Numeric: The covariance parameters;
tapmod	String: the taper model if type is Tapering. Otherwise is NULL.
spacetime	TRUE if spatio-temporal and FALSE if spatial covariance model;
sparse	Logical: is the returned object of class spam? ;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

References

Daley J. D., Porcu E., Bevilacqua M. (2015) Classes of compactly supported covariance functions for multivariate random fields. *Stochastic Environmental Research and Risk Assessment*. 29 (4), 1249–1263.

Gaetan, C. and Guyon, X. (2010) *Spatial Statistics and Modelling*. Spring Verlang, New York.

Gneiting, T. (2013), Strictly and Non-Strictly Positive Definite Functions on Spheres *Bernoulli*, 19, 1327-1349.

Gneiting, T. (2002). Nonseparable, stationary covariance functions for space-time data. *Journal of the American Statistical Association*, 97, 590–600.

Gneiting T, Kleiber W., Schlather M. 2010. Matern cross-covariance functions for multivariate random fields. *Journal of the American Statistical Association*, 105, 1167–1177.

Porcu, E.,Bevilacqua, M. and Genton M. (2015) Spatio-Temporal Covariance and Cross-Covariance Functions of the Great Circle Distance on a Sphere. *Journal of the American Statistical Association*. DOI: 10.1080/01621459.2015.1072541

See Also

GeoKrig, GeoSim, GeoFit

Examples

```
library(GeoModels)
library(spam)

################################################################
###
### Example 1. Spatial covariance matrix associated to
### a Matern correlation model
```

```
###
###############################################################

# Define the spatial-coordinates of the points:
x = runif(500, 0, 1)
y = runif(500, 0, 1)
coords = cbind(x,y)


# Correlation Parameters for Matern model
CorrParam("Matern")

# Matern Parameters
param=list(smooth=0.5,sill=1,scale=0.2,nugget=0)

matrix1 = GeoCovmatrix(coordx=coords, corrmodel="Matern", param=param)
dim(matrix1$covmatrix)



################################################################
###
### Example 2. Spatial covariance matrix associated to
### a Generalized Wendland correlation model
###
###############################################################

# Gen Wendland Parameters
param=list(sill=1,scale=0.2,nugget=0,smooth=0,power2=4)


matrix3 = GeoCovmatrix(coordx=coords, corrmodel="GenWend", param=param,sparse=TRUE)

# Percentage of no zero values in the tapered matrix
matrix3$nozero


################################################################
###
### Example 3. Spatial covariance matrix associated to
### a Generalized Cauchy correlation model
###
###############################################################

# Gen Cauchy Parameters
param=list(sill=1,scale=0.2,nugget=0,power1=1,power2=1)

# Correlation Parameters for Gen Cauchy model
CorrParam("GenCauchy")

matrix4 = GeoCovmatrix(coordx=coords, corrmodel="GenCauchy", param=param)

matrix4$covmatrix[1:4,1:4]


################################################################
###
```

```
### Example 4. Covariance matrix associated to
### a space-time double exponential correlation model
###
###############################################################

# Define the temporal-coordinates:
times = seq(1, 4, 1)

# Define covariance parameters
param=list(scale_s=0.3,scale_t=0.5,sill=1)

# Correlation Parameters for double exp model
CorrParam("Exp_Exp")

# Simulation of a spatial Gaussian random field:
matrix5 = GeoCovmatrix(coordx=coords, coordt=times, corrmodel="Exp_Exp",
                     param=param)

dim(matrix5$covmatrix)

################################################################
###
### Example 5. Covariance matrix associated to
### a  skew gaussian RF with Exp correlation model
###
###############################################################

param=list(sill=1,scale=0.3/3,nugget=0,skew=4)
# Simulation of a spatial Gaussian random field:
matrix6 = GeoCovmatrix(coordx=coords,  corrmodel="Exp", param=param,
                     model="SkewGaussian")

# covariance matrix
matrix6$covmatrix[1:10,1:10]

################################################################
###
### Example 6. Covariance matrix associated to
### a  Weibull RF with Genwend correlation model
###
###############################################################

param=list(sill=1,scale=0.3,nugget=0,shape=4,mean=0,smooth=1,power2=5)
# Simulation of a spatial Gaussian random field:
matrix7 = GeoCovmatrix(coordx=coords,  corrmodel="GenWend", param=param,
                     sparse=TRUE,model="Weibull")

# covariance matrix
matrix7$nozero


################################################################
###
### Example 7. Covariance matrix associated to
### a  binomial gaussian RF with Wendland correlation model
###
###############################################################
```

```
param=list(sill=1,scale=0.2,nugget=0,power2=4)
# Simulation of a spatial Gaussian random field:
matrix8 = GeoCovmatrix(coordx=coords,  corrmodel="Wend0", param=param,n=5,
                     model="Binomial")

# covariance matrix
matrix8$covmatrix[1:10,1:10]



################################################################
###
### Example 8.  Covariance matrix associated to
### a bivariate Matern exponential correlation model
###
###############################################################

set.seed(8)
# Define the spatial-coordinates of the points:
x = runif(10, -1, 1)
y = runif(10, -1, 1)
coords = cbind(x,y)

# Parameters
param=list(scale=0.3,mean_1=0,mean_2=0,sill_1=1,sill_2=1,
             nugget_1=0,nugget_2=0,smooth=0.5,pcol=-0.25)

# Covariance matrix
matrix9 = GeoCovmatrix(coordx=coords, corrmodel="Bi_matern_sep", param=param)$covmatrix

head(matrix9)
```

GeoCV *n-fold kriging Cross-validation*

Description

The procedure use the GeoKrig function to compute n-fold kriging cross-validation using informations from a GeoFit object. The function returns some prediction scores.

Usage

```
GeoCV(fit, K=100, n.fold=0.05,local=FALSE,neighb=NULL,
                      maxdist=NULL,maxtime=NULL,sparse=FALSE, which=1,seed=1)
```

Arguments

fit	An object of class GeoFit.

K	The number of iterations in cross-validation.
n.fold	Numeric; the percentage of data to be deleted (and predicted) in the cross-validation procedure.
local	Logical; If local is TRUE, then local kriging is performed. The default is FALSE.
neighb	Numeric; an optional positive integer indicating the order of neighborhood.
maxdist	Numeric; an optional positive value indicating the distance in the spatial neighborhood if local kriging is performed.
maxtime	Numeric; an optional positive value indicating the distance in the temporal neighborhood if local kriging is performed.
sparse	Logical; if TRUE kriging is computed with sparse matrices algorithms using spam package. Default is FALSE. It should be used with compactly supported covariances.
which	Numeric; In the case of bivariate (tapered) cokriging it indicates which variable to predict. It can be 1 or 2
seed	Numeric; The seed used in the n-fold kriging cross-validation. Default is 1. Comparison between different models in terms of n-fold kriging cross-validation must be performed using the same seed

Value

Returns an object containing the following informations:

predicted	A list of the predicted values in the CV procedure;
data_to_pred	A list of the data to predict in the CV procedure;
mae	The vector of mean absolute error in the CV procedure;
rmse	The vector of root mean squared error in the CV procedure;
lscore	The vector of log-score in the CV procedure;
crps	The vector of continuous ranked probability score in the CV procedure;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoKrig.

GeoFit *Max-Likelihood-Based Fitting of Gaussian and non Gaussian RFs.*

Description

Maximum weighted composite-likelihood fitting for Gaussian and some Non-Gaussian univariate spatial, spatio-temporal and bivariate spatial RFs The function returns the model parameters' estimates and the estimates' variances. Moreover the function allows to fix any of the parameters and setting upper/lower bound in the optimization.

Usage

```
GeoFit(data, coordx, coordy=NULL, coordt=NULL, coordx_dyn=NULL,corrmodel, distance="Eucl",
      fixed=NULL,GPU=NULL, grid=FALSE, likelihood='Marginal', local=c(1,1),
      lower=NULL,maxdist=Inf,neighb=NULL,
    maxtime=Inf, memdist=TRUE,method="cholesky", model='Gaussian',n=1, onlyvar=FALSE ,
      optimizer='Nelder-Mead', parallel=FALSE,
    radius=6371,  sensitivity=FALSE,sparse=FALSE, start=NULL, taper=NULL, tapsep=NULL,
    type='Pairwise', upper=NULL, varest=FALSE, vartype='SubSamp', weighted=FALSE, winconst=NULL, w
      winconst_t=NULL, winstp_t=NULL,X=NULL)
```

Arguments

`data` A d-dimensional vector (a single spatial realisation) or a $(d \times d)$-matrix (a single spatial realisation on regular grid) or a $(t \times d)$-matrix (a single spatial-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatial-temporal realisation on regular grid). For the description see the Section **Details**.

`coordx` A numeric $(d \times 2)$-matrix (where `d` is the number of spatial sites) assigning 2-dimensions of spatial coordinates or a numeric d-dimensional vector assigning 1-dimension of spatial coordinates. Coordinates on a sphere for a fixed radius `radius` are passed in lon/lat format expressed in decimal degrees.

`coordy` A numeric vector assigning 1-dimension of spatial coordinates; `coordy` is interpreted only if `coordx` is a numeric vector or `grid=TRUE` otherwise it will be ignored. Optional argument, the default is `NULL` then `coordx` is expected to be numeric a $(d \times 2)$-matrix.

`coordt` A numeric vector assigning 1-dimension of temporal coordinates. Optional argument, the default is `NULL` then a spatial RF is expected.

`coordx_dyn` A list of m numeric $(d_t \times 2)$-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is `NULL`

`corrmodel` String; the name of a correlation model, for the description see the Section **Details**.

`distance` String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details**.

`fixed` An optional named list giving the values of the parameters that will be considered as known values. The listed parameters for a given correlation function will be not estimated.

`GPU` Numeric; if `NULL` (the default) no OpenCL computation is performed. The user can choose the device to be used. Use `DeviceInfo()` function to see available devices, only double precision devices are allowed

- `grid` — Logical; if `FALSE` (the default) the data are interpreted as spatial or spatial-temporal realisations on a set of non-equispaced spatial sites (irregular grid).
- `likelihood` — String; the configuration of the composite likelihood. `Marginal` is the default, see the Section **Details**.
- `local` — Numeric; number of local work-items of the OpenCL setup
- `lower` — An optional named list giving the values for the lower bound of the space parameter when the optimizer is `L-BFGS-B` or `nlminb` or `optimize`. The names of the list must be the same of the names in the `start` list.
- `maxdist` — Numeric; an optional positive value indicating the maximum spatial distance considered in the composite or tapered likelihood computation. See the Section **Details** for more information.
- `neighb` — Numeric; an optional positive integer indicating the order of neighborhood in the composite likelihood computation. See the Section **Details** for more information.
- `maxtime` — Numeric; an optional positive value indicating the maximum temporal separation considered in the composite or tapered likelihood computation (see **Details**).
- `memdist` — Logical; if `TRUE` then all the distances useful in the composite likelihood estimation are computed before the optimization. `FALSE` is deprecated.
- `method` — String; the type of matrix decomposition used in the simulation. Default is cholesky. The other possible choices is `svd`.
- `model` — String; the type of RF and therefore the densities associated to the likelihood objects. `Gaussian` is the default, see the Section **Details**.
- `n` — Numeric; number of trials in a binomial RF; number of successes in a negative binomial RF
- `onlyvar` — Logical; if `TRUE` (and varest is `TRUE`) only the variance covariance matrix is computed without optimizing. `FALSE` is the default.
- `optimizer` — String; the optimization algorithm (see `optim` for details). `Nelder-Mead` is the default. Other possible choices are `ucminf`,`nlm`, `BFGS` `L-BFGS-B` and `nlminb`. In these last two cases upper and lower bounds can be passed by the user. In the case of one-dimensional optimization, the function `optimize` is used. Two option for global searching optimization using mcGlobaloptim package are possible with `multinlminb` and `multiNelder-Mead` options.
- `parallel` — Logical; if `TRUE` optmization is performed using optimParallel using the maximum number of cores, when optimizer is `L-BFGS-B`.`FALSE` is the default.
- `radius` — Numeric; the radius of the sphere in the case of lon-lat coordinates. The default is 6371, the radius of the earth.
- `sensitivity` — Logical; if `TRUE` then the sensitivy matrix is computed
- `sparse` — Logical; if `TRUE` then maximum likelihood is computed using sparse matrices algorithms (spam packake).It should be used with compactly supported covariance models.`FALSE` is the default.
- `start` — An optional named list with the initial values of the parameters that are used by the numerical routines in maximization procedure. `NULL` is the default (see **Details**).
- `taper` — String; the name of the type of covariance matrix. It can be `Standard` (the default value) or `Tapering` for taperd covariance matrix.

`tapsep` Numeric; an optional value indicating the separabe parameter in the space time adaptive taper (see **Details**).

`type` String; the type of the likelihood objects. If `Pairwise` (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods (see **Details**).

`upper` An optional named list giving the values for the upper bound of the space parameter when the optimizer is or `L-BFGS-B` or `nlminb` or `optimize`. The names of the list must be the same of the names in the `start` list.

`varest` Logical; if `TRUE` the estimates' variances and standard errors are returned. `FALSE` is the default.

`vartype` String; (`SubSamp` the default) the type of method used for computing the estimates' variances, see the Section **Details**.

`weighted` Logical; if `TRUE` the likelihood objects are weighted, see the Section **Details**. If `FALSE` (the default) the composite likelihood is not weighted.

`winconst` Numeric; a bivariate positive vector for computing the spatial sub-window in the sub-sampling procedure. See **Details** for more information.

`winstp` Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the spatial sub-sampling procedure. The case 1 correspond to no overlapping. See **Details** for more information.

`winconst_t` Numeric; a positive value for computing the temporal sub-window in the sub-sampling procedure. See **Details** for more information.

`winstp_t` Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the temporal sub-sampling procedure. The case 1 correspond to no overlapping. See **Details** for more information.

`X` Numeric; Matrix of spatio(temporal)covariates in the linear mean specification.

Details

Note, that the standard likelihood may be seen as particular case of the composite likelihood. In this respect `GeoFit` provides standard maximum likelihood fitting for Gaussian models. For Gaussian and non Gaussian models, only composite likelihood estimation based on pairs are considered. Specifically marginal and conditional pairwise likelihood is considered for each type of random field (Gaussian and not Gaussian). The optimization method is specified using `optimizer`. The default method is `Nelder-mead` and other available methods are `ucminf`, `nlm`, `BFGS`. `L-BFGS-B` and `nlminb`. In the last two cases upper and lower bounds constraints in the optimization can be specified using `lower` and `upper` parameters.

Depending on the dimension of `data` and on the name of the correlation model, the observations are assumed as a realization of a spatial, spatio-temporal or bivariate RF. Specifically, with `data`, `coordx`, `coordy`, `coordt` parameters:

- If `data` is a numeric d-dimensional vector, `coordx` and `coordy` are two numeric d-dimensional vectors (or `coordx` is $(d \times 2)$-matrix and `coordy=NULL`), then the data are interpreted as a single spatial realisation observed on d spatial sites;
- If `data` is a numeric $(t \times d)$-matrix, `coordx` and `coordy` are two numeric d-dimensional vectors (or `coordx` is $(d \times 2)$-matrix and `coordy=NULL`), `coordt` is a numeric t-dimensional vector, then the data are interpreted as a single spatial-temporal realisation of a RF observed on `d` spatial sites and for `t` times.
- If `data` is a numeric $(2 \times d)$-matrix, `coordx` and `coordy` are two numeric d-dimensional vectors (or `coordx` is $(d \times 2)$-matrix and `coordy=NULL`), then the data are interpreted as a single spatial realisation of a bivariate RF observed on `d` spatial sites.

- If `data` is a list, `coordxdyn` is a list and `coordt` is a numeric t-dimensional vector, then the data are interpreted as a single spatial-temporal realisation of a RF observed on dynamical spatial sites (different locations sites for each temporal instants) and for `t` times.

Is is also possible to specify a matrix of covariates using `X`. Specifically:

- In the spatial case `X` must be a $(d \times k)$ covariates matrix associated to `data` a numeric d-dimensional vector;
- In the spatiotemporal case `X` must be a $(N \times k)$ covariates matrix associated to `data` a numeric $(t \times d)$-matrix, where $N = t \times d$;
- In the spatiotemporal case `X` must be a $(N \times k)$ covariates matrix associated to `data` a numeric $(t \times d)$-matrix, where $N = 2 \times d$;

The `corrmodel` parameter allows to select a specific correlation function for the RF. (See `GeoCovmatrix`).

The `distance` parameter allows to consider differents kinds of spatial distances. The settings alternatives are:

1. `Eucl`, the euclidean distance (default value);
2. `Chor`, the chordal distance;
3. `Geod`, the geodesic distance;

The `likelihood` parameter represents the composite-likelihood configurations. The settings alternatives are:

1. `Conditional`, the composite-likelihood is formed by conditionals likelihoods;
2. `Marginal`, the composite-likelihood is formed by marginals likelihoods (default value);
3. `Full`, the composite-likelihood turns out to be the standard likelihood;

The `model` parameter indicates the type of RF considered. The available options are:

RF with marginal symmetric distribution:

- `Gaussian`, for a Gaussian RF.
- `StudentT`, for a StudentT RF (see Bevilacqua M., Caamaño C., Arellano Valle R.B., Morales-Oñate V., 2020).
- `Tukeyh`, for a Tukeyh RF.
- `Logistic`, for a Logistic RF.

RF with positive values and right skewed marginal distribution:

- `Gamma` for a Gamma RF (see Bevilacqua M., Caamano C., Gaetan, 2020)
- `Weibull` for a Weibull RF (see Bevilacqua M., Caamano C., Gaetan, 2020)
- `LogGaussian` for a LogGaussian RF (see Bevilacqua M., Caamano C., Gaetan, 2020)
- `LogLogistics` for a LogLogistic RF.

RF with with possibly asymmetric marginal distribution:

- `SkewGaussian` for a skew Gaussian RF (see Alegrıa et al. (2017))
- `SinhAsinh` for a Sinh-arcsinh RF.

RF with bounded supported data

- `Beta` for a Beta RF.

- `Kumaraswamy` for a Kumaraswamy RF.

RF with for directional data

- `Wrapped` for a wrapped Gaussian RF (see Alegria A., Bevilacqua, M., Porcu, E. (2016))

Rf with marginal counts data

- `Poisson` for a Poisson RF.
- `PoissonZIP` for a zero inflated Poisson RF.
- `Binomial` for a Binomial RF.
- `BinomialNeg` for a negative Binomial RF.
- `BinomialNegZINB` for a zero inflated negative Binomial RF.

For a given `model` the associated parameters are given by nuisance and covariance parameters. In order to obtain the nuisance parameter associated to a specific model use `NuisParam`. In order to obtain the covariance parameter associated to a given covariance model use `CorrParam`.

All the nuisance and covariance parameters must be specified by the user using the `start` and the `fixed` parameter. Specifically:

The `start` parameter allows to specify (as starting values for the optimization) the parameters to be estimated. The `fixed` parameter allows to fix some of the parameters.

Regression parameters in the linear specfication must be specified as `mean,mean1,..meank` (see `NuisParam`). In this case a matrix of covariates with suitable dimension can be specified using the parameter `X`. In the case of a single mean then `X` should not be specified and it is interpreted as a vector of ones.

The `taper` parameter, optional in case that `type=Tapering`, indicates the type of taper correlation model. (See `GeoCovmatrix`)

If a Gaussian or (any) non Gaussian RF is considered then the possible combination is marginal pairwise likelihoods (`likelihood=Marginal`) and `type="Pairwise"`) or (`likelihood=Conditional`) and `type="Pairwise"`)

If a Gaussian RF is considered (`model=Gaussian`) then:

- If the composite is formed by marginal likelihoods (`likelihood=Marginal`):
  - `Pairwise`, the composite-likelihood is defined by the pairwise likelihoods;
  - `Difference`, the composite-likelihood is defined by likelihoods which are obtained as difference of the pairwise likelihoods.
- If the composite is formed by conditional likelihoods (`likelihood=Conditional`)
  - `Pairwise`, the composite-likelihood is defined by the pairwise conditional likelihoods.
- If the composite is formed by a full likelihood (`likelihood=Full`):
  - `Standard`, the objective function is the classical multivariate likelihood;
  - `Restricted`, the objective function is the restricted version of the full likelihood (e.g. Harville 1977, see **References**);
  - `Tapering`, the objective function is the tapered 2 (unbiased estimating equation) version of the full likelihood (e.g. Kaufman et al. 2008, see **References**);
  - `Tapering1`, the objective function is the tapered 1 (biased estimating equation) version of the full likelihood (e.g. Kaufman et al. 2008, see **References**);
  - `CV`, the objective function is the cross validation estimation method ;

Two type of binary weights can be used in the composite likelihood estimation, one based on neighboords and one based on distances.

In the first case binary weights are set to 1 and 0 otherwise depending if the pairs are neighboords of a certain order (2, 3, ..) specified by the parameter (`neighb`). This weighting scheme is effecient for large-data sets since the computation of the 'useful' distance in based on the package RANN that provides fast nearest neighbour search.

In the second case, the `maxdist` parameter set the maximum spatial distance below which pairs of sites with inferior distances Pairs with distance less than `maxdist` have weight 1 and are included in the likelihood computation, instead those with greater distance have weight 0 and then excluded. This weighting scheme is ineffecient for large-data since to find the 'useful' distance all possible distances must be computed. The same arguments of `maxdist` are valid for `maxtime` but here the weigthed composite-likelihood regards the case of spatial-temporal field.

The `varest=TRUE` parameter specifies if the standard error estimation of the estimated parameters must be computed. For Gaussian RFs and standard likelihood estimation, standard errors are computed as square root of the diagonal elements of the Fisher Information matrix (asymptotic covariance matrix of the estimates under increasing domain). For Gaussian and non Gaussian RFs and composite likelihood estimation, standard errors estimate are computed as square root of the diagonal elements of the Godambe Information matrix. (asymptotic covariance matrix of the estimates under increasing domain (see Bevilacqua et. al. (2012) , Bevilacqua and Gaetan (2013)).

In the composite likelihood case the option `varest=TRUE` allows to compute std errors using subsampling tecnique. This type of approximation works well for large datasets. A more robust method to compute std error estimation is trough parametric bootstrap using the function `GeoVarestbootstrap`. In the the sub-sampling procedure,`winconst` and `winstp` parameters represent respectively a positive constant used to determine the sub-window size and the the step with which the sub-window moves. In the spatial case (subset of R^2), the domain is seen as a rectangle $B \times H$, therefore the size of the sub-window side b is given by $b = winconst \times \sqrt(B)$ (similar is of H). For a complete description see Lee and Lahiri (2002). By default `winconst` is set $B/(4 \times \sqrt(B))$. The `winstp` parameter is used to determine the sub-window step. The latter is given by the proportion of the sub-window size, so that when `winstp=1` there is not overlapping between contiguous sub-windows. In the spatial case by default `winstp=0.5`. The sub-window is moved by successive steps in order to cover the entire spatial domain. Observations, that fall in disjoint or overlapping windows are considered indipendent samples.

In the spatio-temporal case `winconst_t` represents the lenght of the temporal sub-window. By default the size of the sub-window is computed following the rule established in Li et al. (2007). By default `winstp` is the time step.

Value

Returns an object of class `GeoFit`. An object of class `GeoFit` is a list containing at most the following components:

`bivariate`	Logical:`TRUE` if the Gaussian RF is bivariate, otherwise `FALSE`;
`clic`	The composite information criterion, if the full likelihood is considered then it coincides with the Akaike information criterion;
`coordx`	A d-dimensional vector of spatial coordinates;
`coordy`	A d-dimensional vector of spatial coordinates;
`coordt`	A t-dimensional vector of temporal coordinates;
`coordx_dyn`	A list of dynamical (in time) spatial coordinates;
`convergence`	A string that denotes if convergence is reached;

corrmodel	The correlation model;
data	The vector or matrix or array of data;
distance	The type of spatial distance;
fixed	The vector of fixed parameters;
iterations	The number of iteration used by the numerical routine;
likelihood	The configuration of the composite likelihood;
logCompLik	The value of the log composite-likelihood at the maximum;
maxdist	The maximum spatial distance used in the weigthed composite likelihood. If no spatial distance is specified then it is NULL;
maxtime	The maximum temporal distance used in the weigthed composite likelihood. If no spatial distance is specified then it is NULL;
message	Extra message passed from the numerical routines;
model	The density associated to the likelihood objects;
missp	True if a misspecified Gaussian model is ued in the composite likelihhod;
n	The number of trials in a binominal RF;the number of successes in a negative Binomial RFs;
ns	The number of (different) location sites in the bivariate case;
nozero	In the case of tapered likelihood the percentage of non zero values in the covariance matrix. Otherwise is NULL.
numcoord	The number of spatial coordinates;
numtime	The number of the temporal realisations of the RF;
param	The vector of parameters' estimates;
param	The radius of the sphere in the case of great circle distance;
stderr	The vector of standard errors;
sensmat	The sensitivity matrix;
varcov	The matrix of the variance-covariance of the estimates;
varimat	The variability matrix;
vartype	The method used to compute the variance of the estimates;
type	The type of the likelihood objects.
winconst	The constant used to compute the window size in the spatial sub-sampling;
winstp	The step used for moving the window in the spatial sub-sampling;
winconst_t	The constant used to compute the window size in the spatio-temporal sub-sampling;
winstp_	The step used for moving the window in the spatio-temporal sub-sampling;
X	The matrix of covariates;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

References

Maximum Restricted Likelihood Estimator:

Harville, D. A. (1977) Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems. *Journal of the American Statistical Association*, **72**, 320–338.

Tapered likelihood:

Kaufman, C. G., Schervish, M. J. and Nychka, D. W. (2008) Covariance Tapering for Likelihood-Based Estimation in Large Spatial Dataset. *Journal of the American Statistical Association*, **103**, 1545–1555.

Composite-likelihood:

Varin, C., Reid, N. and Firth, D. (2011). An Overview of Composite Likelihood Methods. *Statistica Sinica*, **21**, 5–42.

Varin, C. and Vidoni, P. (2005) A Note on Composite Likelihood Inference and Model Selection. *Biometrika*, **92**, 519–528.

Weighted Composite-likelihood for non Gaussian RF:

Alegrıa A., Caro S., Bevilacqua M., Porcu E., Clarke J. (2017) *Estimating covariance functions of multivariate skew-Gaussian random fields on the sphere*. Spatial Statistics **22** 388-402

Alegria A., Bevilacqua, M., Porcu, E. (2016) Likelihood-based inference for multivariate space-time wrapped-Gaussian fields. *Journal of Statistical Computation and Simulation*. **86(13)**, 2583–2597.

Bevilacqua M., Caamano C., Gaetan C. (2020) On modeling positive continuous data with spatio-temporal dependence. *Environmetrics* **31(7)**

Bevilacqua M., Caamaño C., Arellano Valle R.B., Morales-Oñate V. (2020) Non-Gaussian Geostatistical Modeling using (skew) t Processes. *Scandinavian Journal of Statistics*.

Weighted Composite-likelihood for Gaussian RFs:

Bevilacqua, M. Gaetan, C., Mateu, J. and Porcu, E. (2012) Estimating space and space-time covariance functions for large data sets: a weighted composite likelihood approach. *Journal of the American Statistical Association, Theory & Methods*, **107**, 268–280.

Bevilacqua, M., Gaetan, C. (2015) Comparing composite likelihood methods based on pairs for spatial Gaussian random fields. *Statistics and Computing*, **25(5)**, 877-892.

Sub-sampling estimation:

Carlstein, E. (1986) The Use of Subseries Values for Estimating the Variance. *The Annals of Statistics*, **14**, 1171–1179.

Heagerty, P. J. and Lumley T. (2000) Window Subsampling of Estimating Functions with Application to Regression Models. *Journal of the American Statistical Association, Theory & Methods*, **95**, 197–211.

Lee, Y. D. and Lahiri S. N. (2002) Variogram Fitting by Spatial Subsampling. *Journal of the Royal Statistical Society. Series B*, **64**, 837–854.

Examples

```
library(GeoModels)
library(fields)

################################################################
############ Examples of spatial Gaussian RFs ################
################################################################
```

```
################################################################
###
### Example 1, 2 :  Estimation of a spatial Gaussian RF with
### Matern correlation using  pairwise likelihood and
### maximum likelihood with  BGGS and nlminb optimization
###
###############################################################

# Define the spatial-coordinates of the points:
set.seed(3)
N=400  # number of location sites
x <- runif(N, 0, 1)
set.seed(6)
y <- runif(N, 0, 1)
coords <- cbind(x,y)

# Define spatial matrix covariates
X=cbind(rep(1,N),runif(N))

# Set the covariance model's parameters:
corrmodel <- "Matern"
mean <- 0.2
mean1 <- -0.5
sill <- 1
nugget <- 0
scale <- 0.2/3
smooth=0.5
param<-list(mean=mean,mean1=mean1,sill=sill,nugget=nugget,scale=scale,smooth=smooth)

# Simulation of the spatial Gaussian RF:
data <- GeoSim(coordx=coords,corrmodel=corrmodel, param=param,X=X)$data

fixed<-list(nugget=nugget,smooth=smooth)
start<-list(mean=mean,mean1=mean1,scale=scale,sill=sill)



################################################################
###
### Example 1. Maximum pairwise likelihood fitting of
### Gaussian RFs with exponential correlation.
###
###############################################################
fit1 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
                    optimizer="BFGS",neighb=5,likelihood="Marginal",
                    type="Pairwise", start=start,fixed=fixed,X=X)
print(fit1)

################################################################
###
### Example 2. Standard Maximum likelihood fitting of
### Gaussian RFs with exponential correlation.
###
###############################################################
I=Inf
lower<-list(mean=-I,mean1=-I,scale=0,sill=0)
upper<-list(mean=I,mean1=I,scale=I,sill=I)
```

```
fit2 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
                    optimizer="nlminb",upper=upper,lower=lower,
                    likelihood="Full",type="Standard",
                    start=start,fixed=fixed,X=X)
print(fit2)


###############################################################
############ Examples of spatial non-Gaussian RFs #############
###############################################################


################################################################
###
### Example 3. Maximum pairwise likelihood fitting of spatial
### Gamma and Weibull  RFs with Generalized Wendland correlation
### using Nelder-Mead and BFGS
###############################################################
set.seed(524)
# Define the spatial-coordinates of the points:
N=500
x <- runif(N, 0, 1)
y <- runif(N, 0, 1)
coords <- cbind(x,y)
X=cbind(rep(1,N),runif(N))
mean=1; mean1=2 # regression parameters
nugget=0
shape=2
scale=0.2
smooth=0

model="Weibull"
corrmodel="GenWend"
param=list(mean=mean,mean1=mean1,sill=1-nugget,scale=scale,
                     shape=shape,nugget=nugget,power2=4,smooth=smooth)
# Simulation of a  non stationary weibull RF:
data <- GeoSim(coordx=coords, corrmodel=corrmodel,model=model,X=X,
           param=param)$data

fixed<-list(nugget=nugget,power2=4,sill=1-nugget,smooth=smooth)
start<-list(mean=mean,mean1=mean1,scale=scale,shape=shape)

# Maximum pairwise composite-likelihood fitting of the RF:
fit <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                    neighb=5,likelihood="Marginal",type="Pairwise",X=X,
                    optimizer="Nelder-Mead",
                    start=start,fixed=fixed)
print(fit$param)

model="Gamma"
start<-list(mean=mean,mean1=mean1,scale=scale)
fixed<-list(nugget=nugget,power2=4,sill=1-nugget,shape=6)
# Maximum pairwise composite-likelihood fitting of the RF:
fit <- GeoFit(data=data,coordx=coords,corrmodel="Wend0", model=model,
                    neighb=5,likelihood="Marginal",type="Pairwise",X=X,
                      optimizer="BFGS",
                    start=start,fixed=fixed)
```

```
print(fit$param)


################################################################
###
### Example 4. Maximum pairwise likelihood fitting of
### StudendT spatial  RFs with Wendland correlation
###
###############################################################

set.seed(15274)
# Define the spatial-coordinates of the points:
N=300
x <- runif(N, 0, 1)
y <- runif(N, 0, 1)
coords <- cbind(x,y)
X=cbind(rep(1,N),runif(N))
mean=1; mean1=2 # regression parameters
nugget=0
sill=0.5
scale=0.2
df=4   ## degrees of freedom
model="StudentT"
corrmodel="Wend0"

# Simulation of a studentT RF:
param=list(mean=mean,mean1=mean1,sill=sill,scale=scale,df=1/df,nugget=nugget,power2=4)
data <- GeoSim(coordx=coords, corrmodel=corrmodel,model=model,X=X,
           param=param)$data
##  estimation assuming df unknown
fixed<-list(nugget=nugget,power2=4)
start<-list(mean=mean,mean1=mean1,scale=scale,sill=sill,df=1/df)

I=Inf
lower<-list(mean=-I,mean1=-I,scale=0,sill=0,df=0)
upper<-list(mean=I,mean1=I,scale=I,sill=I,df=0.5)


# Maximum pairwise composite-likelihood fitting of the RF:

fit1 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                 neighb=5,likelihood="Marginal",type="Pairwise",X=X,
               lower=lower,upper=upper,optimizer="nlminb",start=start,fixed=fixed)
print(fit1$param)





##  df must be rounded and fixed
df=round(1/(as.numeric(fit1$param['df'])))
 fixed<-list(nugget=nugget,power2=4,df=1/df)
 start<-list(mean=mean,mean1=mean1,scale=scale,sill=sill)

lower<-list(mean=-I,mean1=-I,scale=0,sill=0)
upper<-list(mean=I,mean1=I,scale=I,sill=I)
```

```
# Maximum pairwise composite-likelihood fitting of the RF:
fit <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                    neighb=5,likelihood="Marginal",type="Pairwise",X=X,
                  lower=lower,upper=upper, optimizer="nlminb" , start=start,fixed=fixed)
print(fit$param)


################################################################
###
### Example 5. Maximum pairwise likelihood fitting of
### SinhAsinh-Gaussian spatial  RFs with Wendland correlation
###
###############################################################
set.seed(261)
model="SinhAsinh"
# Define the spatial-coordinates of the points:
x <- runif(500, 0, 1)
y <- runif(500, 0, 1)
coords <- cbind(x,y)

corrmodel="Wend0"
mean=0;nugget=0
sill=1
skew=-0.5
tail=1.5
power2=4
c_supp=0.2

# model parameters
param=list(power2=power2,skew=skew,tail=tail,
             mean=mean,sill=sill,scale=c_supp,nugget=nugget)
data <- GeoSim(coordx=coords, corrmodel=corrmodel,model=model, param=param)$data

plot(density(data))
fixed=list(power2=power2,nugget=nugget)
start=list(scale=c_supp,skew=skew,tail=tail,mean=mean,sill=sill)
# Maximum pairwise likelihood:
fit1 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                    maxdist=0.05,likelihood="Marginal",type="Pairwise",
                   optimizer="BFGS" , start=start,fixed=fixed)
print(fit1$param)
# Maximum likelihood:
fit2 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                    likelihood="Full",type="Standard",
                   optimizer="BFGS" , start=start,fixed=fixed)
print(fit2$param)


################################################################
###
### Example 6. Maximum pairwise likelihood fitting of
### Binomial and negative Binomial   RFs
### with exponential correlation
###
###############################################################
```

```
set.seed(422)
N=350
x <- runif(N, 0, 1)
y <- runif(N, 0, 1)
coords <- cbind(x,y)
mean=0.1; mean1=0.8; mean2=-0.5 # regression parameters
X=cbind(rep(1,N),runif(N),runif(N)) # marix covariates
corrmodel <- "Wend0"
param=list(mean=mean,mean1=mean1,mean2=mean2,sill=1,nugget=0,scale=0.2,power2=4)
# Simulation of the spatial Binomial-Gaussian RF:
data <- GeoSim(coordx=coords, corrmodel=corrmodel, model="Binomial", n=10,X=X,
             param=param)$data

fixed <- list(nugget=nugget,power2=4,sill=1)
start <- list(scale=0.2,mean=mean,mean1=mean1,mean2=mean2)
# Maximum pairwise likelihood:
fit1 <- GeoFit(data=data, coordx=coords, corrmodel=corrmodel,n=10, X=X, start=start,
                    maxdist=0.05,model="Binomial",  fixed=fixed, optimizer="BFGS")

print(fit1)

set.seed(220)
# Simulation of the spatial Negative Binomial-Gaussian RF:
data <- GeoSim(coordx=coords, corrmodel=corrmodel, model="BinomialNeg", n=5,X=X,
              param=param)$data
# Maximum pairwise likelihood:
fit2 <- GeoFit(data=data, coordx=coords, corrmodel=corrmodel, n=5,X=X, start=start,
              maxdist=0.05, model="BinomialNeg", fixed=fixed, optimizer="BFGS")


print(fit2)


###############################################################
######### Examples of spatio-temporal RFs ###########
###############################################################
set.seed(52)
# Define the temporal sequence:
time <- seq(1, 10, 1)

# Define the spatial-coordinates of the points:
x <- runif(20, 0, 1)
set.seed(42)
y <- runif(20, 0, 1)
coords=cbind(x,y)

# Set the covariance model's parameters:
corrmodel="Exp_Exp"
scale_s=0.2/3
scale_t=1
sill=1
nugget=0
mean=0

param<-list(mean=0,scale_s=scale_s,scale_t=scale_t,
            sill=sill,nugget=nugget)
```

```
# Simulation of the spatial-temporal Gaussian RF:
data <- GeoSim(coordx=coords,coordt=time,corrmodel=corrmodel,
              param=param)$data

################################################################
###
### Example 7. Maximum pairwise likelihood fitting of a
### space time Gaussian RF with double-exponential correlation
### using  BFGS
###
###############################################################
# Fixed parameters
fixed<-list(nugget=nugget)
# Starting value for the estimated parameters
start<-list(mean=mean,scale_s=scale_s,scale_t=scale_t,sill=sill)

# Maximum composite-likelihood fitting of the RF:
fit <- GeoFit(data=data,coordx=coords,coordt=time,
                    corrmodel="Exp_Exp",maxtime=1,maxdist=0.3,
                    likelihood="Marginal",type="Pairwise",
                    optimizer="BFGS" , start=start,fixed=fixed)
print(fit)


################################################################
###
### Example 8. Maximum standard likelihood fitting of a
### space time Gaussian RF observed on  dynamical spatial coordinates
### with double-exponential correlation using  BFGS
###
###############################################################

maxN=50
coordx_dyn=list()
set.seed(31)
for(k in 1:length(time))
{
NN=sample(1:maxN,size=1)
x <- runif(NN, 0, 1)
y <- runif(NN, 0, 1)
coordx_dyn[[k]]=cbind(x,y)
}

data <- GeoSim(coordx_dyn=coordx_dyn, coordt=time, corrmodel="Exp_Exp",
                param=param)$data
I=Inf
lower<-list(mean=-I,scale_s=0,scale_t=0,sill=0)
upper<-list(mean=-I,scale_s=I,scale_t=I,sill=I)
fit <- GeoFit(data=data,coordx_dyn=coordx_dyn,coordt=time,
              corrmodel="Exp_Exp",likelihood="Full",type="Standard",
              optimizer="BFGS", lower=lower,upper=upper,
              start=start,fixed=fixed)
print(fit)
```

```
################################################################
######### Examples of  spatial bivariate RFs ###########
################################################################


#################################################################
###
### Example 9. Maximum, and pairwise  likelihood fitting of a
### bivariate Gaussian RF with separable Bivariate  matern
### (cross) correlation model using BFGS and a parallel
### implementation of L-BFGS-B
################################################################

# Define the spatial-coordinates of the points:
set.seed(5)
x <- runif(250, 0, 1)
y <- runif(250, 0, 1)
coords=cbind(x,y)
# parameters
param=list(mean_1=0,mean_2=0,scale=0.1,smooth=0.5,sill_1=1,sill_2=1,
           nugget_1=0,nugget_2=0,pcol=0.2)
# Simulation of a spatial Gaussian RF:
data <- GeoSim(coordx=coords, corrmodel="Bi_Matern_sep",
              param=param)$data

# selecting fixed and estimated parameters
fixed=list(nugget_1=0,nugget_2=0,smooth=0.5)
start=list(mean_1=0,mean_2=0,sill_1=var(data[1,]),sill_2=var(data[2,]),
           scale=0.1,pcol=cor(data[1,],data[2,]))


# Maximum pairwise likelihood

fitcl<- GeoFit(data=data, coordx=coords, corrmodel="Bi_Matern_sep",
                     likelihood="Marginal",type="Pairwise",
                     optimizer="BFGS" , start=start,fixed=fixed,
                     maxdist=c(0.05,0.05,0.05))
print(fitcl)

I=Inf
lower=list(mean_1=-I,mean_2=-I,sill_1=0,sill_2=0,
           scale=0,pcol=-1)
upper=list(mean_1=I,mean_2=I,sill_1=I,sill_2=I,
           scale=I,pcol=1)
# Maximum likelihood :
require(optimParallel)
fitml<- GeoFit(data=data, coordx=coords, likelihood="Full",
               corrmodel="Bi_Matern_sep", type="Standard",
               upper=upper,lower=lower,optimizer="L-BFGS-B",
               parallel=TRUE,
               start=start,fixed=fixed)
print(fitml)
```

GeoKrig *Spatial (bivariate) and spatio temporal optimal linear prediction for Gaussian and non Gaussian RFs.*

Description

For a given set of spatial location sites and temporal instants, the function computes optimal linear prediction and associated mean square error for the Gaussian and non Gaussian case.

Usage

```
GeoKrig(data, coordx, coordy=NULL, coordt=NULL, coordx_dyn=NULL, corrmodel,distance="Eucl",
    grid=FALSE, loc, maxdist=NULL, maxtime=NULL, method="cholesky",
    model="Gaussian", n=1,nloc=NULL,mse=FALSE, lin_opt=TRUE,
    param, radius=6371, sparse=FALSE,taper=NULL,tapsep=NULL,
     time=NULL, type="Standard",type_mse=NULL,
     type_krig="Simple",weigthed=TRUE,which=1, X=NULL,Xloc=NULL)
```

Arguments

`data` A d-dimensional vector (a single spatial realisation) or a $(d \times d)$-matrix (a single spatial realisation on regular grid) or a $(t \times d)$-matrix (a single spatial-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatial-temporal realisation on regular grid) giving the data used for prediction.

`coordx` A numeric $(d \times 2)$-matrix (where `d` is the number of spatial sites) giving 2-dimensions of spatial coordinates or a numeric d-dimensional vector giving 1-dimension of spatial coordinates used for prediction. qndd-dimensional vector giving 1-dimension of spatial coordinates. Coordinates on a sphere for a fixed radius `radius` are passed in lon/lat format expressed in decimal degrees.

`coordy` A numeric vector giving 1-dimension of spatial coordinates used for prediction; `coordy` is interpreted only if `coordx` is a numeric vector or `grid=TRUE` otherwise it will be ignored. Optional argument, the default is `NULL` then `coordx` is expected to be numeric a $(d \times 2)$-matrix.

`coordt` A numeric vector giving 1-dimension of temporal coordinates used for prediction; the default is `NULL` then a spatial random field is expected.

`coordx_dyn` A list of m numeric $(d_t \times 2)$-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is `NULL`

`corrmodel` String; the name of a correlation model, for the description see the Section **Details**.

`distance` String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details** of `GeoFit`.

`grid` Logical; if `FALSE` (the default) the data used for prediction are interpreted as spatial or spatial-temporal realisations on a set of non-equispaced spatial sites (irregular grid).

`lin_opt` Logical;If TRUE (default) then optimal (pairwise) linear kriging is computed. Otherwise optimal (pairwise) kriging is computed in the mean square sense.

`loc` A numeric $(n \times 2)$-matrix (where `n` is the number of spatial sites) giving 2-dimensions of spatial coordinates to be predicted.

`maxdist` Numeric; an optional positive value indicating the maximum spatial compact support in the case of covariance tapering kriging.

maxtime	Numeric; an optional positive value indicating the maximum temporal compact support in the case of covasriance tapering kriging.
method	String; the type of matrix decomposition used in the simulation. Default is `cholesky`. The other possible choices is `svd`.
n	Numeric; the number of trials in a binomial random fields. Default is 1.
nloc	Numeric; the number of trials of the locations sites to be predicted in a binomial random fields type II. Default is 1.
mse	Logical; if `TRUE` (the default) MSE of the kriging predictor is computed
model	String; the type of RF and therefore the densities associated to the likelihood objects. `Gaussian` is the default, see the Section **Details**.
param	A list of parameter values required for the correlation model.See the Section **Details**.
radius	Numeric: the radius of the sphere if coordinates are passed in lon/lat format;
sparse	Logical; if `TRUE` kriging is computed with sparse matrices algorithms using spam package. Default is FALSE. It should be used with compactly supported covariances.
taper	String; the name of the taper correlation function, see the Section **Details**.
tapsep	Numeric; an optional value indicating the separabe parameter in the space time quasi taper (see **Details**).
time	A numeric $(m \times 1)$ vector (where `m` is the number of temporal instants) giving the temporal instants to be predicted; the default is `NULL` then only spatial prediction is performed.
type	String; if `Standard` then standard kriging is performed;if `Tapering` then kriging with covariance tapering is performed;if `Pairwise` then pairwise kriging is performed
type_mse	String; if `Theoretical` then theoretical MSE pairwise kriging is computed. If `SubSamp` then an estimation based on subsampling is computed.
type_krig	String; the type of kriging. If `Simple` (the default) then simple kriging is performed. (See the Section **Details**).
weigthed	Logical; if `TRUE` then decreasing weigths coming from a compactly supported correlation function with compact support `maxdist` (`maxtime`)are used in the pairwise kriging.
which	Numeric; In the case of bivariate (tapered) cokriging it indicates which variable to predict. It can be 1 or 2
X	Numeric; Matrix of spatio(temporal)covariates in the linear mean specification.
Xloc	Numeric; Matrix of spatio(temporal)covariates in the linear mean specification associated to predicted locations.

Details

Best linear unbiased predictor and associated mean square error is computed for Gaussian and some non Gaussian cases. Specifically, for a spatial or spatio-temporal or spatial bivariate dataset, given a set of spatial locations and temporal istants and a correlation model `corrmodel` with some fixed parameters and given the type of RF (`model`) the function computes simple kriging, for the specified spatial locations `loc` and temporal instants `time`, providing also the respective mean square error. For the choice of the spatial or spatio temporal correlation model see details in `GeoCovmatrix` function. The list `param` specifies mean and covariance parameters, see `CorrParam` and `GeoCovmatrix` for details. The type_krig parameter indicates the type of kriging. In the case of simple kriging, the known mean can be specified by the parameter mean in the list param (See examples). In the Gaussian case, it is possible to perform kriging based on covariance tapering for simple kriging (Furrer et. al, 2008). In this case, space or space-time tapered function and spatial or spatio- temporal compact support must be specified. For the choice of a space or space-time tapered function see GeoCovmatrix. When performing kriging with covariance tapering, sparse matrix algorithms are exploited using the package spam.

Value

Returns an object of class Kg. An object of class Kg is a list containing at most the following components:

bivariate	TRUE if spatial bivariate cokriging is performed, otherwise FALSE;
coordx	A d-dimensional vector of spatial coordinates used for prediction;
coordy	A d-dimensional vector of spatial coordinates used for prediction;
coordt	A t-dimensional vector of temporal coordinates used for prediction;
corrmodel	String: the correlation model;
covmatrix	The covariance matrix if type is Standard. An object of class spam if type is Tapering
data	The vector or matrix or array of data used for prediction
distance	String: the type of spatial distance;
grid	TRUE if the spatial data used for prediction are observed in a regular grid, otherwise FALSE;
loc	A $(n \times 2)$-matrix of spatial locations to be predicted.
n	The number of trial for Binomial RFs
nozero	In the case of tapered simple kriging the percentage of non zero values in the covariance matrix. Otherwise is NULL.
numcoord	Numeric:he number d of spatial coordinates used for prediction;
numloc	Numeric: the number n of spatial coordinates to be predicted;
numtime	Numeric: the number d of the temporal instants used for prediction;
numt	Numeric: the number m of the temporal instants to be predicted;
model	The type of RF, see GeoFit.
param	Numeric: The covariance parameters;
pred	A $(m \times n)$-matrix of spatio or spatio temporal kriging prediction;
radius	Numeric: the radius of the sphere if coordinates are pssed in lon/lat format;
spacetime	TRUE if spatio-temporal kriging and FALSE if spatial kriging;
tapmod	String: the taper model if type is Tapering. Otherwise is NULL.
time	A m-dimensional vector of temporal coordinates to be predicted;
type	String: the type of kriging (Standard or Tapering).
type_krig	String: the type of kriging.
mse	A $(m \times n)$-matrix of spatio or spatio temporal mean square error kriging prediction;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

References

Gaetan, C. and Guyon, X. (2010) *Spatial Statistics and Modelling*. Spring Verlang, New York. Furrer R., Genton, M.G. and Nychka D. (2006). *Covariance Tapering for Interpolation of Large Spatial Datasets*. Journal of Computational and Graphical Statistics, **15-3**, 502–523.

See Also

GeoCovmatrix

Examples

```
library(GeoModels)
library(fields)
library(hypergeo)

################################################################
############### Examples of Spatial kriging  ###################
################################################################

# Define the spatial-coordinates of the points:
set.seed(79)
x = runif(200, 0, 1)
y = runif(200, 0, 1)
coords=cbind(x,y)
# Set the exponential cov parameters:
corrmodel_1 = "exponential"
mean=0
sill=1
nugget=0
scale=0.3/3
param=list(mean=mean,sill=sill,nugget=nugget,scale=scale)

# Set the wendland parameters (two compatible correlations):
corrmodel_2 = "Wend0"
mean=0
sill=1
nugget=0
power2=3
c_supp=0.3
param_wen=list(mean=mean,sill=sill,nugget=nugget,scale=c_supp,power2=power2)

# Simulation of the spatial Gaussian random field:
data = GeoSim(coordx=coords, corrmodel=corrmodel_1,
              param=param)$data

# locations to predict
xx=seq(0,1,0.025)
loc_to_pred=as.matrix(expand.grid(xx,xx))

################################################################
###
```

```
### Example 1. Spatial simple kriging of n sites of a
### Gaussian random fields with exponential correlation.
###
###############################################################
pr=GeoKrig(loc=loc_to_pred,coordx=coords,corrmodel=corrmodel_1,
       param= param, data=data,mse=TRUE)


################################################################
###
### Example 3. Spatial  simple kriging of n sites of a
### Gaussian random fields using a compatible Wendland model
###
###############################################################
param=list(mean=mean,sill=sill,nugget=nugget,scale=scale,power2=power2)
pr_wen=GeoKrig(loc=loc_to_pred,coordx=coords,corrmodel=corrmodel_2,data=data,
       param=param_wen,sparse=TRUE,mse=TRUE)

colour = rainbow(100)
par(mfrow=c(2,2))
zlim=c(-2.6,2.6)
# simple kriging map prediction
image.plot(xx, xx, matrix(pr$pred,ncol=length(xx)),col=colour,
           zlim=zlim,xlab="",ylab="",
           main="Simple Kriging with  exponential model ")

# simple kriging MSE map prediction variance
image.plot(xx, xx, matrix(pr$mse,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Std error")

# simple  kriging map prediction
image.plot(xx, xx, matrix(pr_wen$pred,ncol=length(xx)),col=colour,
           zlim=zlim,xlab="",ylab="",main="Simple Kriging with Wendland model")

# simple kriging MSE map prediction variance
image.plot(xx, xx, matrix(pr_wen$mse,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Std error")




################################################################
###
### Example 4. Spatial  simple kriging of a binomial
###            random field
###
###############################################################
set.seed(312)
model="Binomial";n=6

# Define the spatial-coordinates of the points:
x = runif(800)
y = runif(800)
coords=cbind(x,y)
#### mean and covariance parameters ###
mean=0
sill=1
```

```
nugget=0
scale=0.2
#################
param=list(mean=mean,sill=sill,nugget=nugget,scale=scale,power2=4)
# Simulation of the Binomial Gaussian random field:
data = GeoSim(coordx=coords, corrmodel="Wend0",model=model,n=n,
              sparse=TRUE,param=param)$data


par(mfrow=c(1,2))

#### map of simulated data
quilt.plot(x, y, data,nlevel=n+1,col=rainbow(n+1),zlim=c(0,n), main="Data")

## estimation with pairwise likelihood
fixed=list(nugget=nugget,power2=4,sill=1)
start=list(mean=0,scale=scale)
# Maximum pairwise likelihood fitting :
fit = GeoFit(data, coordx=coords, corrmodel="Wend0",model=model,n=n,memdist=FALSE,
                    likelihood='Marginal', type='Pairwise',maxdist=0.03,
                    optimizer="BFGS", start=start,fixed=fixed)

# locations to predict
xx=seq(0,1,0.03)
loc_to_pred=as.matrix(expand.grid(xx,xx))

## simple kriging
pr=GeoKrig(data=data, coordx=coords,loc=loc_to_pred,corrmodel="Wend0",model=model,n=n,
       sparse=TRUE,param= as.list(c(fit$param,fixed)))

#standard   binomial kriging
map_binom=matrix(pr$pred,ncol=length(xx))
image.plot(xx, xx, map_binom,col=rainbow(n+1),zlim=c(0,n),
           xlab="",ylab="",main="Simple Kriging ")


################################################################
###
### Example 5. Spatial  simple kriging of the residuals of a
###            Weibull random field
###
###############################################################
set.seed(312)

model="Weibull"

corrmodel = "GenWend"
# Define the spatial-coordinates of the points:
NN=400
coords=cbind(runif(NN),runif(NN))
## matrix covariates
a0=rep(1,NN)
a1=runif(NN,-1,1)
X=cbind(a0,a1)
# Set model parameters
shape=2
## regression parameters
```

```
mean = 1
mean1= -0.2
# correlation parameters
sill = 1
nugget = 0
power2=4
scale = 0.3
smooth=0

## simulation
param=list(shape=shape,nugget=nugget,mean=mean,mean1=mean1,
  scale=scale,sill=sill,power2=power2,smooth=smooth)

data = GeoSim(coordx=coords,corrmodel=corrmodel, param=param,
                model=model,X=X)$data

#####starting and fixed parameters
fixed=list(nugget=nugget,power2=power2,smooth=smooth,sill=sill)
start=list(mean=mean,mean1=mean1, scale=scale,shape=shape)

## estimation with pairwise likelihood
maxdist=0.05
fit2 = GeoFit(data=data,coordx=coords,corrmodel=corrmodel,X=X,
              maxdist=maxdist,likelihood="Marginal",type="Pairwise",
              start=start,fixed=fixed, model = model)
## computing residuals
res=GeoResiduals(fit2)

# locations to predict
xx=seq(0,1,0.04)
loc_to_pred=as.matrix(expand.grid(xx,xx))

#optimal linear  kriging for residuals
pr=GeoKrig(data=res$data, coordx=coords,loc=loc_to_pred,corrmodel=corrmodel,model=model,mse=TRUE,
       sparse=TRUE,param=as.list(c(res$param,res$fixed)))

## map of residuals
par(mfrow=c(1,3))
quilt.plot(coords,res$data,main="Residuals")
map=matrix(pr$pred,ncol=length(xx))
mapmse=matrix(pr$mse,ncol=length(xx))
image.plot(xx, xx, map,
           xlab="",ylab="",main="Residuals Kriging ")

image.plot(xx, xx, mapmse,
           xlab="",ylab="",main="Simple Kriging ")


################################################################
###
### Example 5. Spatial  simple kriging of a t
###            random field
###
###############################################################

###############################################################
model="StudentT"
```

```
df=5
corrmodel = "Wend0"
nsel=200
coords=cbind(runif(nsel),runif(nsel))

mean = 0
sill = 1
nugget = 0
power2=4
scale = 0.2

# Starting value for the estimated parameters
set.seed(32)
param=list(nugget=nugget,mean=mean, scale=scale,sill=sill,df=1/df,power2=power2)
data = GeoSim(coordx=coords,corrmodel=corrmodel, param=param,
     model=model)$data

fixed=list(nugget=nugget,power2=4,df=1/df)
start=list(mean=mean, scale=scale,sill=sill)
I=Inf
upper=list(mean=I, scale=I,sill=I)
lower=list(mean=-I, scale=0,sill=0)
# Maximum pairwise composite-likelihood fitting of the RF:
fit = GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
             maxdist=0.04,likelihood="Marginal",type="Pairwise",
             lower=lower,upper=upper,
             optimizer="nlminb", start=start,fixed=fixed, model = model)

# locations to predict
xx=seq(0,1,0.04)
loc_to_pred=as.matrix(expand.grid(xx,xx))

#optimal linear  kriging
pr=GeoKrig(data=data, coordx=coords,loc=loc_to_pred,corrmodel="Wend0",model=model,
       sparse=TRUE,param= as.list(c(fit$param,fit$fixed)))

par(mfrow=c(1,2))
#### map of simulated data
quilt.plot(coords[,1], coords[,2], data, main="Data")

map_t=matrix(pr$pred,ncol=length(xx))
image.plot(xx, xx, map_t,
           xlab="",ylab="",main="Simple Kriging ")


################################################################
########### Examples of spatio temporal kriging ############
################################################################

model="Gaussian"
# Define the spatial-coordinates of the points:
x = runif(200, 0, 1)
y = runif(200, 0, 1)
coords=cbind(x,y)
times=1:10

# Define model correlation and associated parameters
```

```
corrmodel="Wend0_Wend0"
param=list(nugget=0,mean=0,power2_s=4,power2_t=4,
        scale_s=0.15,scale_t=2,sill=1)

# Simulation of the space time Gaussian random field:
set.seed(31)
data=GeoSim(coordx=coords,coordt=times,corrmodel=corrmodel,
           param=param)$data

# Maximum pairwise likelihood fitting of the space time random field:
start = list(scale_s=0.2/3,scale_t=1,sill=1,mean=0)
fixed = list(nugget=0,power2_s=4,power2_t=4)
I=Inf
lower=list(scale_s=0,scale_t=0,sill=0,mean=-I)
upper=list(scale_s=I,scale_t=I,sill=I,mean=I)
fit = GeoFit(data, coordx=coords, coordt=times, model=model, corrmodel=corrmodel,
             likelihood='Marginal', type='Pairwise',start=start,fixed=fixed,
             optimizer="nlminb", maxdist=0.1,maxtime=1)

################################################################
###
### Example 6. Spatio temporal simple kriging of n locations
### sites and m temporal instants for a Gaussian random fields
### with estimated double exponential correlation.
###
###############################################################
param=as.list(c(fit$param,fit$fixed))

# locations to predict
xx=seq(0,1,0.03)
loc_to_pred=as.matrix(expand.grid(xx,xx))
#  Define the times to predict
times_to_pred=1:2

pr=GeoKrig(loc=loc_to_pred,time=times_to_pred,coordx=coords,coordt=times,
       sparse=TRUE,corrmodel=corrmodel, param=param,data=data,mse=TRUE)

par(mfrow=c(2,3))
#zlim=c(-2.5,2.5)
colour = rainbow(100)

for(i in 1:2) {
quilt.plot(coords,data[i,] ,col=colour,main = paste(" data  at Time=" , i))
image.plot(xx, xx, matrix(pr$pred[i,],ncol=length(xx)),col=colour,
        main = paste(" Kriging Time=" , i),ylab="")
image.plot(xx, xx, matrix(pr$mse[i,],ncol=length(xx)),col=colour,
           main = paste("Std err Time=" , i),ylab="")
}

################################################################
########### Examples of spatial bivariate cokriging ############
################################################################

################################################################
###
### Example 7. Bivariate simple cokriging  of n locations
### for a Gaussian random fields with separable Matern correlation
```

```
###
################################################################
# Define the spatial-coordinates of the points:
set.seed(12)
x = runif(300, 0, 1)
y = runif(300, 0, 1)
coords=cbind(x,y)

# Simulation of a spatial bivariate Gaussian random field
# with Matern separable covariance model


param=list(scale=0.3/3,mean_1=0,mean_2=0,sill_1=1,sill_2=1,
          nugget_1=0,nugget_2=0,pcol=0.7,smooth=0.5)

data = GeoSim(coordx=coords, corrmodel="Bi_matern_sep", param=param)$data

fixed=list(nugget_1=0,nugget_2=0,smooth=0.5,mean_1=0,mean_2=0)
start=list(sill_1=var(data[,1]),sill_2=var(data[,2]),scale=0.3/3,
      pcol=cor(data[1,],data[2,]))

# Maximum Composite likelihood fitting of the random field:
fitcl= GeoFit(data, coordx=coords, corrmodel="Bi_matern_sep",
        likelihood="Marginal",type="Pairwise",maxdist=0.1,
        optimizer="BFGS", start=start,fixed=fixed)


# locations to predict
xx=seq(0,1,0.04)
loc_to_pred=as.matrix(expand.grid(xx,xx))
colour = rainbow(100)

pr1=GeoKrig(loc=loc_to_pred,coordx=coords,corrmodel="Bi_matern_sep",
       param= as.list(c(fitcl$param,fitcl$fixed)), data=data,which=1,mse=TRUE)

pr2=GeoKrig(loc=loc_to_pred,coordx=coords,corrmodel="Bi_matern_sep",
       param= as.list(c(fitcl$param,fitcl$fixed)), data=data,which=2,mse=TRUE)

par(mfrow=c(2,3))

quilt.plot(coords,data[1,])
# simple kriging map prediction of the first variable
image.plot(xx, xx, matrix(pr1$pred,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="First Simple coKriging")

# simple kriging map prediction variance of the first variable
image.plot(xx, xx, matrix(pr1$mse,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Std error")

 quilt.plot(coords,data[2,])
 # simple kriging map prediction of the second variable
image.plot(xx, xx, matrix(pr2$pred,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Second Simple coKriging")

# simple kriging map prediction variance of the second variable
image.plot(xx, xx, matrix(pr2$mse,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Std error")
```

GeoKrigloc	*Spatial (bivariate) and spatio temporal optimal linear local prediction for Gaussian and non Gaussian RFs.*

Description

For a given set of spatial location sites and temporal instants, the function computes optimal linear prediction and associated mean square error for the Gaussian and non Gaussian case using a spatial (temporal) neighborhood computed using the function `GeoNeighborhood`

Usage

```
GeoKrigloc(data, coordx, coordy=NULL, coordt=NULL, coordx_dyn=NULL, corrmodel, distance="Eucl", gr
       loc, neighb=NULL, maxdist=NULL, maxtime=NULL, method="cholesky", model="Gaussian", n=1,nloc
       param, radius=6371, sparse=FALSE, time=NULL, type="Standard",type_mse=NULL, type_krig="Sim
          which=1, X=NULL,Xloc=NULL)
```

Arguments

`data` — A d-dimensional vector (a single spatial realisation) or a $(d \times d)$-matrix (a single spatial realisation on regular grid) or a $(t \times d)$-matrix (a single spatial-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatial-temporal realisation on regular grid) giving the data used for prediction.

`coordx` — A numeric $(d \times 2)$-matrix (where `d` is the number of spatial sites) giving 2-dimensions of spatial coordinates or a numeric d-dimensional vector giving 1-dimension of spatial coordinates used for prediction. qndd-dimensional vector giving 1-dimension of spatial coordinates. Coordinates on a sphere for a fixed radius `radius` are passed in lon/lat format expressed in decimal degrees.

`coordy` — A numeric vector giving 1-dimension of spatial coordinates used for prediction; `coordy` is interpreted only if `coordx` is a numeric vector or `grid=TRUE` otherwise it will be ignored. Optional argument, the default is `NULL` then `coordx` is expected to be numeric a $(d \times 2)$-matrix.

`coordt` — A numeric vector giving 1-dimension of temporal coordinates used for prediction; the default is `NULL` then a spatial random field is expected.

`coordx_dyn` — A list of m numeric $(d_t \times 2)$-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is `NULL`

`corrmodel` — String; the name of a correlation model, for the description see the Section **Details**.

`distance` — String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details** of `GeoFit`.

`grid` — Logical; if `FALSE` (the default) the data used for prediction are interpreted as spatial or spatial-temporal realisations on a set of non-equispaced spatial sites (irregular grid).

`loc` — A numeric $(n \times 2)$-matrix (where `n` is the number of spatial sites) giving 2-dimensions of spatial coordinates to be predicted.

`neighb` — Numeric; an optional positive integer indicating the order of the neighborhood.

`maxdist` — Numeric; an optional positive value indicating the distance in the spatial neighborhood.

`maxtime`	Numeric; an optional positive value indicating the distance in the temporal neighborhood.
`method`	String; the type of matrix decomposition used in the simulation. Default is `cholesky`. The other possible choices is `svd`.
`n`	Numeric; the number of trials in a binomial random fields. Default is 1.
`nloc`	Numeric; the number of trials of the locations sites to be predicted in a binomial random fields type II. Default is 1.
`mse`	Logical; if `TRUE` (the default) MSE of the kriging predictor is computed
`model`	String; the type of RF and therefore the densities associated to the likelihood objects. `Gaussian` is the default, see the Section **Details**.
`param`	A list of parameter values required for the correlation model.See the Section **Details**.
`radius`	Numeric: the radius of the sphere if coordinates are passed in lon/lat format;
`sparse`	Logical; if `TRUE` kriging is computed with sparse matrices algorithms using spam package. Default is FALSE. It should be used with compactly supported covariances.
`time`	A numeric $(m \times 1)$ vector (where `m` is the number of temporal instants) giving the temporal instants to be predicted; the default is `NULL` then only spatial prediction is performed.
`type`	String; if `Standard` then standard kriging is performed;if `Tapering` then kriging with covariance tapering is performed;if `Pairwise` then pairwise kriging is performed
`type_mse`	String; if `Theoretical` then theoretical MSE pairwise kriging is computed. If `SubSamp` then an estimation based on subsampling is computed.
`type_krig`	String; the type of kriging. If `Simple` (the default) then simple kriging is performed. (See the Section **Details**).
`weigthed`	Logical; if `TRUE` then decreasing weigths coming from a compactly supported correlation function with compact support `maxdist` (`maxtime`)are used in the pairwise kriging.
`which`	Numeric; In the case of bivariate (tapered) cokriging it indicates which variable to predict. It can be 1 or 2
`X`	Numeric; Matrix of spatio(temporal)covariates in the linear mean specification.
`Xloc`	Numeric; Matrix of spatio(temporal)covariates in the linear mean specification associated to predicted locations.

Details

This function use the `GeoKrig` with a spatial or spatio-temporal neighborhood computed using the function `GeoNeighborhood`. The neighborhood is specified with the option `maxdist` and `maxtime`.

Value

Returns an object of class `Kg`. An object of class `Kg` is a list containing at most the following components:

`bivariate`	`TRUE` if spatial bivariate cokriging is performed, otherwise `FALSE`;
`coordx`	A d-dimensional vector of spatial coordinates used for prediction;
`coordy`	A d-dimensional vector of spatial coordinates used for prediction;

coordt	A t-dimensional vector of temporal coordinates used for prediction;
corrmodel	String: the correlation model;
covmatrix	The covariance matrix if type is Standard. An object of class spam if type is Tapering
data	The vector or matrix or array of data used for prediction
distance	String: the type of spatial distance;
grid	TRUE if the spatial data used for prediction are observed in a regular grid, otherwise FALSE;
loc	A $(n \times 2)$-matrix of spatial locations to be predicted.
n	The number of trial for Binomial RFs
nozero	In the case of tapered simple kriging the percentage of non zero values in the covariance matrix. Otherwise is NULL.
numcoord	Numeric:he number d of spatial coordinates used for prediction;
numloc	Numeric: the number n of spatial coordinates to be predicted;
numtime	Numeric: the number d of the temporal instants used for prediction;
numt	Numeric: the number m of the temporal instants to be predicted;
model	The type of RF, see GeoFit.
param	Numeric: The covariance parameters;
pred	A $(m \times n)$-matrix of spatio or spatio temporal kriging prediction;
radius	Numeric: the radius of the sphere if coordinates are pssed in lon/lat format;
spacetime	TRUE if spatio-temporal kriging and FALSE if spatial kriging;
tapmod	String: the taper model if type is Tapering. Otherwise is NULL.
time	A m-dimensional vector of temporal coordinates to be predicted;
type	String: the type of kriging (Standard or Tapering).
type_krig	String: the type of kriging.
mse	A $(m \times n)$-matrix of spatio or spatio temporal mean square error kriging prediction;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

References

Gaetan, C. and Guyon, X. (2010) *Spatial Statistics and Modelling*. Spring Verlang, New York. Furrer R., Genton, M.G. and Nychka D. (2006). *Covariance Tapering for Interpolation of Large Spatial Datasets*. Journal of Computational and Graphical Statistics, **15-3**, 502–523.

See Also

GeoCovmatrix

Examples

```
################################################################
############### Examples of Spatial local kriging  #############
################################################################
require(GeoModels)
require(fields)
####
model="Gaussian"

# Define the spatial-coordinates of the points:
set.seed(79)
x = runif(800, 0, 1)
y = runif(800, 0, 1)
coords=cbind(x,y)
# Set the exponential cov parameters:
corrmodel = "Exponential"
mean=0
sill=1
nugget=0
scale=0.3/3
param=list(mean=mean,sill=sill,nugget=nugget,scale=scale)


# Simulation of the spatial Gaussian random field:
data = GeoSim(coordx=coords, corrmodel=corrmodel,
              param=param)$data

# locations to predict
xx=seq(0,1,0.025)
loc_to_pred=as.matrix(expand.grid(xx,xx))

################################################################
###
### Example 1. Comparing spatial kriging with local kriging for
### a Gaussian random field with exponential correlation.
###
###############################################################
pr=GeoKrig(loc=loc_to_pred,coordx=coords,corrmodel=corrmodel,
      model=model,param= param, data=data,mse=TRUE)

pr_loc=GeoKrigloc(data=data,loc=loc_to_pred,coordx=coords,corrmodel=corrmodel,
      model=model,maxdist=0.1, param= param,mse=TRUE)

colour = rainbow(100)
par(mfrow=c(2,2))
zlim=c(-2.6,2.6)
# simple kriging map prediction
image.plot(xx, xx, matrix(pr$pred,ncol=length(xx)),col=colour,
           zlim=zlim,xlab="",ylab="",
           main="Kriging with  exponential model ")

# simple kriging MSE map prediction variance
image.plot(xx, xx, matrix(pr$mse,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Std error")

# simple  kriging map prediction
```

```
image.plot(xx, xx, matrix(pr_loc$pred,ncol=length(xx)),col=colour,
           zlim=zlim,xlab="",ylab="",main="Local Kriging with exponential model")

# simple kriging MSE map prediction variance
image.plot(xx, xx, matrix(pr_loc$mse,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Std error")


############################################################
#### Example: spatio temporal  Gaussian local kriging ######
############################################################


require(GeoModels)
require(fields)
set.seed(78)
coords=cbind(runif(50),runif(50))
coordt=seq(0,5,0.25)
corrmodel="Exp_Exp"
param=list(nugget=0,mean=0,scale_s=0.2/3,scale_t=0.25/3,sill=2)

data = GeoSim(coordx=coords, coordt=coordt,corrmodel="Exp_Exp",
                         param=param)$data
##  four location to predict
loc_to_pred=matrix(runif(8),4,2)
## three temporal instants to predict
time=c(0.5,1.2,3.7)


pr=GeoKrig(data=data,loc=loc_to_pred,time=time,coordx=coords,coordt=coordt,corrmodel=corrmodel,
  model="Gaussian", param= param, mse=TRUE)
pr_loc=GeoKrigloc(data=data,loc=loc_to_pred,time=time,coordx=coords,coordt=coordt,corrmodel=corrmodel,
  maxdist=0.15,maxtime=0.5,model="Gaussian", param= param, mse=TRUE)

##  full and local prediction
pr$pred
pr_loc$pred


############################################################
### Example 7. Bivariate local cokriging  ##################
############################################################
require(GeoModels)
require(fields)
# Define the spatial-coordinates of the points:
set.seed(128)
x = runif(200, 0, 1)
y = runif(200, 0, 1)
coords=cbind(x,y)

param=list(scale=0.3/3,mean_1=0,mean_2=0,sill_1=1,sill_2=1,
          nugget_1=0,nugget_2=0,pcol=0.7,smooth=0.5)

data = GeoSim(coordx=coords, corrmodel="Bi_matern_sep", param=param)$data

fixed=list(nugget_1=0,nugget_2=0,smooth=0.5,mean_1=0,mean_2=0)
start=list(sill_1=var(data[,1]),sill_2=var(data[,2]),scale=0.3/3,
```

```
      pcol=cor(data[1,],data[2,]))
# Maximum Composite likelihood fitting of the random field:
fitcl= GeoFit(data, coordx=coords, corrmodel="Bi_matern_sep",
        likelihood="Marginal",type="Pairwise",maxdist=0.1,
        optimizer="BFGS", start=start,fixed=fixed)

neigh=GeoNeighborhood(data, coordx=coords,bivariate=TRUE,
                  loc=loc_to_pred,neighb=10)

# locations to predict

xx=seq(0,1,0.025)
loc_to_pred=as.matrix(expand.grid(xx,xx))

pr=GeoKrig(loc=loc_to_pred,coordx=coords,corrmodel="Bi_matern_sep",
       param= as.list(c(fitcl$param,fitcl$fixed)), data=data,which=1,mse=TRUE)

pr_loc=GeoKrigloc(loc=loc_to_pred,coordx=coords,corrmodel="Bi_matern_sep",maxdist=0.15,param= as.list(c(fit

par(mfrow=c(2,2))

colour = rainbow(100)
zlim=c(-2.6,2.6)
# simple kriging map prediction
image.plot(xx, xx, matrix(pr$pred,ncol=length(xx)),col=colour,
           zlim=zlim,xlab="",ylab="",
           main="CoKriging with Bivariate Matern model")

# simple kriging MSE map prediction variance
image.plot(xx, xx, matrix(pr$mse,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Std error")

# simple  kriging map prediction
image.plot(xx, xx, matrix(pr_loc$pred,ncol=length(xx)),col=colour,
           zlim=zlim,xlab="",ylab="",main="Local CoKriging with Bivariate Matern model")

# simple kriging MSE map prediction variance
image.plot(xx, xx, matrix(pr_loc$mse,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Std error")
```

GeoNeighborhood *Spatio (temporal) neighborhood selection for local kriging.*

Description

Given a set of spatio (temporal) locations and data, the procedure select a spatio (temporal) neighborhood associated to some given spatio (temporal) locations. The neighborhood is computed using a fixed spatio (temporal) threshold or including a fixed number of spatio (temporal) neighbors.

Usage

```
GeoNeighborhood(data=NULL, coordx, coordy=NULL, coordt=NULL, coordx_dyn=NULL, bivariate=FALSE,
          distance="Eucl", grid=FALSE, loc, neighb=NULL,maxdist=NULL,maxtime=NULL, radius=6371, t
```

Arguments

`data`	An optional d-dimensional vector (a single spatial realisation) or a $(d \times d)$-matrix (a single spatial realisation on regular grid) or a $(t \times d)$-matrix (a single spatial-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatial-temporal realisation on regular grid).
`coordx`	A numeric $(d \times 2)$-matrix (where `d` is the number of spatial sites) giving 2-dimensions of spatial coordinates or a numeric d-dimensional vector giving 1-dimension of spatial coordinates. Coordinates on a sphere for a fixed radius `radius` are passed in lon/lat format expressed in decimal degrees.
`coordy`	A numeric vector giving 1-dimension of spatial coordinates; `coordy` is interpreted only if `coordx` is a numeric vector or `grid=TRUE` otherwise it will be ignored. Optional argument, the default is `NULL` then `coordx` is expected to be numeric a $(d \times 2)$-matrix.
`coordt`	A numeric vector giving 1-dimension of temporal coordinates. Optional argument, the default is `NULL` then a spatial RF is expected.
`coordx_dyn`	A list of m numeric $(d_t \times 2)$-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is `NULL`
`bivariate`	If TRUE then data is considered as spatial bivariate data.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details** of `GeoFit`.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as spatial or spatial-temporal realisations on a set of non-equispaced spatial sites (irregular grid).
`loc`	A (1×2)-matrix giving the spatial coordinate of the location for which a neighborhood is computed .
`neighb`	Numeric; an optional positive integer indicating the order of neighborhood.
`maxdist`	Numeric; a positive value indicating the maximum spatial distance considered in the spatial neighborhood selection.
`maxtime`	Numeric; a positive value indicating the maximum temporal distance considered in the temporal neighborhood selection.
`radius`	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is the radius of the earth in Km (i.e. 6371)
`time`	Numeric; a value giving the temporal instant for which a neighborhood is computed.
`X`	Numeric; an optional Matrix of spatio (temporal) covariates.

Value

Returns a list containing the following informations:

`coordx`	A list of the matrix coordinates of the computed spatial neighborhood ;
`coordt`	A vector of the computed temporal neighborhood;
`data`	A list of the vector of data associated with the spatio (temporal) neighborhood;
`distance`	The type of spatial distance;
`numcoord`	The vector of numbers of location sites involved the spatial neighborhood;
`numtime`	The vector of numbers of temporal insttants involved the temporal neighborhood;

radius	The radius of the sphere if coordinates are passed in lon/lat format;
spacetime	TRUE if spatio-temporal and FALSE if spatial RF;
X	The matrix of spatio (temporal) covariates associated with the computed spatio (temporal) neighborhood;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

Examples

```
library(GeoModels)
##########################################
#### Example: spatial neighborhood  ######
##########################################
set.seed(75)
coords=cbind(runif(500),runif(500))

param=list(nugget=0,mean=0,scale=0.2,sill=1,
            power2=4,smooth=1)

data_all = GeoSim(coordx=coords, corrmodel="GenWend",
                         param=param)$data

plot(coords)
##two locations
loc_to_pred=matrix(c(0.3,0.5,0.7,0.2),2,2)

points(loc_to_pred,pch=20)
neigh=GeoNeighborhood(data_all, coordx=coords,
                  loc=loc_to_pred,neighb=8)

# two Neighborhoods
neigh$coordx
points(neigh$coordx[[1]],pch=20,col="red")
points(neigh$coordx[[2]],pch=20,col="blue")
# associated data
neigh$data


###################################################
#### Example: spatio temporal spatial neighborhood#
###################################################

set.seed(78)
coords=matrix(runif(80),40,2)
coordt=seq(0,6,0.25)

param=list(nugget=0,mean=0,scale_s=0.2/3,scale_t=0.25/3,sill=2)

data_all = GeoSim(coordx=coords, coordt=coordt,corrmodel="Exp_Exp",
                         param=param)$data
##  two location to predict
loc_to_pred=matrix(runif(4),2,2)
## three temporal instants to predict
```

```
time=c(1,2,3)

plot(coords,xlim=c(0,1),ylim=c(0,1))
points(loc_to_pred,pch=20)

neigh=GeoNeighborhood(data_all, coordx=coords,  coordt=coordt,
                  loc=loc_to_pred,time=time,maxdist=0.4,maxtime=0.25)

# first spatio-temporal neighborhoods
# with  associated data
neigh$coordx[[1]]
neigh$coordt[[1]]
neigh$data[[1]]

###################################################
#### Example: bivariate  spatial neighborhood #####
###################################################

set.seed(79)
coords=matrix(runif(100),50,2)

param=list(mean_1=0,mean_2=0,scale=0.12,smooth=0.5,
           sill_1=1,sill_2=1,nugget_1=0,nugget_2=0,pcol=0.5)

data_all = GeoSim(coordx=coords,corrmodel="Bi_matern_sep",
                 param=param)$data
##  two location to predict
loc_to_pred=matrix(runif(4),2,2)

neigh=GeoNeighborhood(data_all, coordx=coords,bivariate=TRUE,
                  loc=loc_to_pred,maxdist=0.25)

plot(coords)
points(loc_to_pred,pch=20)
points(neigh$coordx[[1]],col="red",pch=20)
points(neigh$coordx[[2]],col="red",pch=20)
```

GeoQQ *Quantile-quantile plot of residuals*

Description

The procedure plots a quantile-quantile plot for the residuals associated to a fitted model

Usage

```
GeoQQ(fit)
```

Arguments

fit	A GeoFit object obtained from GeoResiduals.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

Examples

```
library(GeoModels)
set.seed(2711)

model="StudentT";df=6
N=400 # number of location sites
# Set the coordinates of the points:
x = runif(N, 0, 1)
y = runif(N, 0, 1)
coords=cbind(x,y)


# regression parameters
mean = 5
mean1=0.8

X=cbind(rep(1,N),runif(N))
# correlation parameters:
corrmodel = "Wend0"
sill = 1
nugget = 0
scale = 0.3
power2=4


param=list(mean=mean,mean1=mean1, sill=sill, nugget=nugget,
           scale=scale,df=1/df,power2=power2)
# Simulation of the Gaussian RF:
data = GeoSim(coordx=coords, corrmodel=corrmodel, X=X,model=model,param=param)$data

start=list(mean=mean,mean1=mean1, scale=scale)
fixed=list(nugget=nugget,sill=sill,power2=power2,df=1/df)
# Maximum composite-likelihood fitting
fit = GeoFit(data,coordx=coords, corrmodel=corrmodel,model=model,X=X,
                    likelihood="Marginal",type='Pairwise',start=start,
                    fixed=fixed,neighb=2)

res=GeoResiduals(fit)

GeoQQ(res)
```

GeoResiduals *Computes fitted covariance and/or variogram*

Description

The procedure return a GeoFit object associated to the estimated residuals

Usage

```
GeoResiduals(fit)
```

Arguments

fit	A fitted object obtained from the GeoFit.

Value

A GeoFit object with the estimated residuals

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit.

Examples

```
library(GeoModels)
set.seed(211)

model="Weibull";shape=4
N=700 # number of location sites
# Set the coordinates of the points:
x = runif(N, 0, 1)
y = runif(N, 0, 1)
coords=cbind(x,y)


# regression parameters
mean = 5
mean1=0.8

X=cbind(rep(1,N),runif(N))
# correlation parameters:
corrmodel = "Wend0"
sill = 1
nugget = 0
scale = 0.3
power2=4


param=list(mean=mean,mean1=mean1, sill=sill, nugget=nugget,
           scale=scale,shape=shape,power2=power2)
# Simulation of the Gaussian RF:
data = GeoSim(coordx=coords, corrmodel=corrmodel, X=X,model=model,param=param)$data

start=list(mean=mean,mean1=mean1, scale=scale,shape=shape)
fixed=list(nugget=nugget,sill=sill,power2=power2)
# Maximum composite-likelihood fitting
fit = GeoFit(data,coordx=coords, corrmodel=corrmodel,model=model,X=X,
                    likelihood="Marginal",type='Pairwise',start=start,
```

```
                    fixed=fixed,neighb=3)

res=GeoResiduals(fit)
mean(res$data) # should be approx 1
# Empirical estimation of the variogram for the residuals:
vario = GeoVariogram(res$data,coordx=coords,maxdist=0.5)

# Plot of covariance and variogram functions:
GeoCovariogram(res, show.vario=TRUE, vario=vario,pch=20)
```

`GeoScatterplot` *h-scatterplot for space and space-time data.*

Description

The function produces h-scatterplots for the spatial, spatio-temporal and bivariate setting.

Usage

```
GeoScatterplot(data, coordx, coordy=NULL, coordt=NULL, coordx_dyn=NULL,
            distance='Eucl', grid=FALSE, maxdist=NULL,neighb=NULL,
            times=NULL, numbins=4, radius=6371, bivariate=FALSE)
```

Arguments

`data`	A d-dimensional vector (a single spatial realisation) or a $(n \times d)$-matrix (n iid spatial realisations) or a $(d \times d)$-matrix (a single spatial realisation on regular grid) or an $(d \times d \times n)$-array (n iid spatial realisations on regular grid) or a $(t \times d)$-matrix (a single spatial-temporal realisation) or an $(t \times d \times n)$-array (n iid spatial-temporal realisations) or or an $(d \times d \times t \times n)$-array (a single spatial-temporal realisation on regular grid) or an $(d \times d \times t \times n)$-array ($n$ iid spatial-temporal realisations on regular grid). See `GeoFit` for details.
`coordx`	A numeric $(d \times 2)$-matrix (where `d` is the number of spatial sites) assigning 2-dimensions of spatial coordinates or a numeric d-dimensional vector assigning 1-dimension of spatial coordinates. Coordinates on a sphere for a fixed radius `radius` are passed in lon/lat format expressed in decimal degrees.
`coordy`	A numeric vector assigning 1-dimension of spatial coordinates; `coordy` is interpreted only if `coordx` is a numeric vector or `grid=TRUE` otherwise it will be ignored. Optional argument, the default is `NULL` then `coordx` is expected to be numeric a $(d \times 2)$-matrix.
`coordt`	A numeric vector assigning 1-dimension of temporal coordinates. Optional argument, the default is `NULL` then a spatial random field is expected.
`coordx_dyn`	A list of m numeric $(d_t \times 2)$-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is `NULL`
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details** of `GeoFit`.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as spatial or spatial-temporal realisations on a set of non-equispaced spatial sites.

maxdist	A numeric value denoting the spatial maximum distance, see the Section **Details**.
neighb	Numeric; an optional positive integer indicating the order of neighborhood. See the Section **Details** for more information.
times	A numeric vector denoting the temporal instants involved **Details**.
numbins	A numeric value denoting the numbers of bins, see the Section **Details**.
radius	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is the radius of the earth in Km (i.e. 6371)
bivariate	Logical; if FALSE (the default) the data are interpreted as univariate spatial or spatial-temporal realisations. Otherwise they are intrepreted as a a realization from a bivariate field.

Details

some details

Value

The returned object is eventually a list with:

covariance	The vector of the estimated covariance function;
variogram	The vector of the estimated variogram function;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

Examples

```
library(GeoModels)

################################################################
################################################################
set.seed(514)
require(GeoModels)

NN = 500
coords = cbind(runif(NN),runif(NN))

param = list(mean=0,sill=1,nugget=0,power2=4,scale=0.4,smooth=0)

corrmodel = "GenWend"; model = "Gaussian"

data = GeoSim(coordx = coords,corrmodel = corrmodel,
                  model = model,param = param)$data


GeoScatterplot(data,coords,distance="Eucl",maxdist=.15,numbins=6)
```

GeoSim *Simulation of Gaussian and non Gaussian Random Fields.*

Description

Simulation of Gaussian and some non Gaussian spatial, spatio-temporal and spatial bivariate random fields. The function return a realization of a Random Field for a given covariance model and covariance parameters. Simulation is based on Cholesky decomposition.

Usage

```
GeoSim(coordx, coordy=NULL, coordt=NULL, coordx_dyn=NULL, corrmodel, distance="Eucl",
    GPU=NULL, grid=FALSE, local=c(1,1),method="cholesky", model='Gaussian', n=1, param,
      radius=6371, sparse=FALSE, X=NULL)
```

Arguments

coordx	A numeric $(d \times 2)$-matrix (where d is the number of spatial sites) giving 2-dimensions of spatial coordinates or a numeric d-dimensional vector giving 1-dimension of spatial coordinates. Coordinates on a sphere for a fixed radius radius are passed in lon/lat format expressed in decimal degrees.
coordy	A numeric vector giving 1-dimension of spatial coordinates; coordy is interpreted only if coordx is a numeric vector or grid=TRUE otherwise it will be ignored. Optional argument, the default is NULL then coordx is expected to be numeric a $(d \times 2)$-matrix.
coordt	A numeric vector giving 1-dimension of temporal coordinates. Optional argument, the default is NULL then a spatial RF is expected.
coordx_dyn	A list of m numeric $(d_t \times 2)$-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is NULL
corrmodel	String; the name of a correlation model, for the description see the Section **Details**.
distance	String; the name of the spatial distance. The default is Eucl, the euclidean distance. See the Section **Details** of GeoFit.
GPU	Numeric; if NULL (the default) no GPU computation is performed.
grid	Logical; if FALSE (the default) the data are interpreted as spatial or spatial-temporal realisations on a set of non-equispaced spatial sites (irregular grid).
local	Numeric; number of local work-items of the GPU
method	String; the type of matrix decomposition used in the simulation. Default is cholesky. The other possible choices is svd.
model	String; the type of RF and therefore the densities associated to the likelihood objects. Gaussian is the default, see the Section **Details**.
n	Numeric; the number of trials for binomial RFs. The number of successes in the negative Binomial RFs. Default is 1.
param	A list of parameter values required in the simulation procedure of RFs, see **Examples**.
radius	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is the radius of the earth in Km (i.e. 6371)

sparse — Logical; if TRUE then cholesky decomposition is performed using sparse matrices algorithms (spam packake). It should be used with compactly supported covariance models.FALSE is the default.

X — Numeric; Matrix of space-time covariates.

Value

Returns an object of class GeoSim. An object of class GeoSim is a list containing at most the following components:

bivariate — Logical:TRUE if the Gaussian RF is bivariate, otherwise FALSE;

coordx — A d-dimensional vector of spatial coordinates;

coordy — A d-dimensional vector of spatial coordinates;

coordt — A t-dimensional vector of temporal coordinates;

coordx_dyn — A list of dynamical (in time) spatial coordinates;

corrmodel — The correlation model; see GeoCovmatrix.

data — The vector or matrix or array of data, see GeoFit;

distance — The type of spatial distance;

model — The type of RF, see GeoFit.

n — The number of trial for Binomial RFs;the number of successes in a negative Binomial RFs;

numcoord — The number of spatial coordinates;

numtime — The number the temporal realisations of the RF;

param — The vector of parameters' estimates;

radius — The radius of the sphere if coordinates are passed in lon/lat format;

randseed — The seed used for the random simulation;

spacetime — TRUE if spatio-temporal and FALSE if spatial RF;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

Examples

```
library(GeoModels)
library(mapproj)
library(fields)

################################################################
###
### Example 1. Simulation of a spatial Gaussian RF on a regular grid
###
###############################################################

# Define the spatial-coordinates of the points:
x <- seq(0,1,0.045)
y <- seq(0,1,0.045)
set.seed(261)
# Simulation of a spatial Gaussian RF with Matern correlation function
```

```
data1 <- GeoSim(x,y,grid=TRUE, corrmodel="Matern", param=list(smooth=0.5,
             mean=0,sill=1,scale=0.4/3,nugget=0))$data

# Simulation of a spatial Gaussian RF with Generalized Wendland correlation function
# using sparse alghorithm matrices
set.seed(261)
data2 <- GeoSim(x,y,grid=TRUE, corrmodel="GenWend", param=list(smooth=0,
              power2=4,mean=0,sill=1,scale=0.4,nugget=0))$data
par(mfrow=c(1,2))
image.plot(x,y,data1,col=terrain.colors(100),main="Matern",xlab="",ylab="")
image.plot(x,y,data2,col=terrain.colors(100),main="Wendland",xlab="",ylab="")



################################################################
###
### Example 2. Simulation of a spatial binomial RF
### with  underlying exponential correlation
### on a regular grid
###
################################################################

# Define the spatial-coordinates of the points:
x <- seq(0, 1, 0.022)
y <- seq(0, 1, 0.022)
coords <- cbind(x,y)
set.seed(251)
n=5
# Simulation of a spatial Binomial RF:
sim <- GeoSim(x,y,grid=TRUE, corrmodel="Wend0",
             model="Binomial",n=n,sparse=TRUE,
             param=list(nugget=0,mean=0,scale=.2,sill=1,power2=4))

image.plot(x,y,sim$data,nlevel=n+1,col=terrain.colors(n+1),zlim=c(0,n))


################################################################
###
### Example 3. Simulation of a spatial Weibull  RF
### with  underlying exponential correlation
###
###############################################################
# Define the spatial-coordinates of the points:
x <- seq(0,1,0.032)
y <- seq(0,1,0.032)
set.seed(261)
# Simulation of a spatial Gaussian RF with Matern correlation function
data1 <- GeoSim(x,y,grid=TRUE, corrmodel="Exponential",model="Weibull",
         param=list(shape=1.2,mean=0,sill=1,scale=0.3/3,nugget=0))$data
image.plot(x,y,data1,col=terrain.colors(200),main="Weibull RF",xlab="",ylab="")



################################################################
###
### Example 4. Simulation of a spatial t  RF
### with  with  underlying exponential correlation
```

```
###
###############################################################
# Define the spatial-coordinates of the points:
x <- seq(0,1,0.03)
y <- seq(0,1,0.03)
set.seed(268)
# Simulation of a spatial Gaussian RF with Matern correlation function
data1 <- GeoSim(x,y,grid=TRUE, corrmodel="GenWend",model="StudentT", sparse=TRUE,
         param=list(df=1/4,mean=0,sill=1,scale=0.3,nugget=0,smooth=1,power2=5))$data
image.plot(x,y,data1,col=terrain.colors(100),main="Student-t RF",xlab="",ylab="")



################################################################
###
### Example 5. Simulation of a sinhasinh RF
###   with  underlying Wend0 correlation.
###
###############################################################

# Define the spatial-coordinates of the points:
x <- runif(800, 0, 2)
y <- runif(800, 0, 2)
coords <- cbind(x,y)
set.seed(261)
corrmodel="Wend0"
# Simulation of a spatial Gaussian RF:
param=list(power2=4,skew=0,tail=1,
             mean=0,sill=1,scale=0.2,nugget=0)  ## gaussian case
data0 <- GeoSim(coordx=coords, corrmodel=corrmodel,
               model="SinhAsinh", param=param,sparse=TRUE)$data
plot(density(data0),xlim=c(-7,7))

param=list(power2=4,skew=0,tail=0.7,
             mean=0,sill=1,scale=0.2,nugget=0) ## heavy tails
data1 <- GeoSim(coordx=coords, corrmodel=corrmodel,
               model="SinhAsinh", param=param,sparse=TRUE)$data
lines(density(data1),lty=2)

param=list(power2=4,skew=0.5,tail=1,
             mean=0,sill=1,scale=0.2,nugget=0)  ## asymmetry
data2 <- GeoSim(coordx=coords, corrmodel=corrmodel,
               model="SinhAsinh", param=param,sparse=TRUE)$data
lines(density(data2),lty=3)



################################################################
###
### Example 6. Simulation of a bivariate Gaussian RF
### with separable bivariate exponential correlation model
### on a regular grid.
###
###############################################################

# Define the spatial-coordinates of the points:
```

```
x <- seq(-1,1,0.08)
y <- seq(-1,1,0.08)

# Simulation of a bivariate spatial Gaussian RF:
# with a separable Bivariate Matern
set.seed(12)
param=list(mean_1=0,mean_2=0,scale=0.12,smooth=0.5,
           sill_1=1,sill_2=1,nugget_1=0,nugget_2=0,pcol=0.5)
data <- GeoSim(x,y,grid=TRUE,corrmodel="Bi_matern_sep",
              param=param)$data
par(mfrow=c(1,2))

image.plot(x,y,data[,,1],col=terrain.colors(100),main="1",xlab="",ylab="")
image.plot(x,y,data[,,2],col=terrain.colors(100),main="2",xlab="",ylab="")




################################################################
###
### Example 7. Simulation of a  spatio temporal Gaussian RF.
### observed on  dynamical location sites with double exponential correlation
###
###############################################################

# Define the dynamical spatial-coordinates of the points:

coordt=1:5
coordx_dyn=list()
maxN=40
set.seed(8)
for(k in 1:length(coordt))
{
NN=sample(1:maxN,size=1)
x <- runif(NN, 0, 1)
y <- runif(NN, 0, 1)
coordx_dyn[[k]]=cbind(x,y)
}
coordx_dyn

param<-list(nugget=0,mean=0,scale_s=0.2/3,scale_t=2/3,sill=1)
data <- GeoSim(coordx_dyn=coordx_dyn, coordt=coordt, corrmodel="Exp_Exp",
                     param=param)$data
## spatial realization at first temporal instants
data[[1]]
## spatial realization at third temporal instants
data[[3]]




################################################################
###
### Example 8. Simulation of a Gaussian RF
###  with a Wend0 correlation in the north emisphere of the planet earth
###
```

```
################################################################
distance="Geod";radius=6371

NN=1000 ## total point on the sphere on lon/lat format
set.seed(80)
coords=cbind(runif(NN,-180,180),runif(NN,0,90))
## Set the wendland parameters
corrmodel <- "Wend0"
param<-list(mean=0,sill=1,nugget=0,scale=radius*0.1,power2=3)
# Simulation of a spatial Gaussian RF on the sphere
#set.seed(2)
data <- GeoSim(coordx=coords,corrmodel=corrmodel,sparse=TRUE,
        distance=distance,radius=radius,param=param)$data



#################################################################
###
### Example 9. Simulation of a Gaussian RF
###  with Wend0 model  on USA
###
################################################################

distance="Geod";radius=6378.88
NN=40
x=seq(-125,-64,length.out=NN)
y=seq(27,50, length.out =NN)
nrow(expand.grid(x,y))
## Set the wendland parameters
corrmodel <- "Wend0"
param<-list(mean=0,sill=1,nugget=0,scale=radius*0.3,power2=3)
# Simulation of a spatial Gaussian RF on the sphere
#set.seed(2)
data <- GeoSim(x,y,grid=TRUE,corrmodel=corrmodel,sparse=TRUE,
        distance=distance,radius=radius,param=param)$data
image.plot(x,y,data,col=terrain.colors(100),xlab="",ylab="")
map("usa", add = TRUE)




#################################################################
###
### Example 10. Simulation of a Wrapped  RF
### with   underlying exponential correlation
### on a regular grid
###
################################################################

# Define the spatial-coordinates of the points:
x <- runif(200,0, 1)
y <- runif(200,0, 1)
coords <- cbind(x,y)
set.seed(251)

# Simulation of a spatial wrapped RF:
sim <- GeoSim(coordx=coords, corrmodel="Exp",
```

```
              model="Wrapped",
              param=list(nugget=0,mean=0,scale=.1,sill=1))$data

long <- 0.08;
x1 <- coords[,1] + long*cos(sim)
y1 <- coords[,2] + long*sin(sim)
eps <- 0.1
plot(0,xlim=c(0-eps,1+eps),ylim=c(0-eps,1+eps));
require(shape)
Arrows(coords[,1], coords[,2], x1, y1, arr.length = 0.2, code = 2,  arr.type = "triangle",col=1)
```

GeoTests *Statistical Hypothesis Tests for Nested Models*

Description

The function performs statistical hypothesis tests for nested models based on composite or standard likelihood versions of Wald-type and Wilks-type (likelihood ratio) statistics.

Usage

```
GeoTests(object1, object2, ..., statistic)
```

Arguments

object1	An object of class GeoFit.
object2	An object of class GeoFit that is a nested model within object1.
...	Further successively nested objects.
statistic	String; the name of the statistic used within the hypothesis test (see **Details**).

Details

The implemented hypothesis tests for nested models are based on the following statistics:

1. Wald-type (Wald);
2. Likelihood ratio or Wilks-type (Wilks under standard likelihood); For composite likelihood available variants of the basic version are:
    - Rotnitzky and Jewell adjustment (WilksRJ);
    - Satterhwaite adjustment (WilksS);
    - Chandler and Bate adjustment (WilksCB);
    - Pace, Salvan and Sartori adjustment (WilksPSS);

More specifically, consider an p-dimensional random vector $\mathbf{Y}$ with probability density function $f(\mathbf{y};\theta)$, where $\theta \in \Theta$ is a q-dimensional vector of parameters. Suppose that $\theta = (\psi, \tau)$ can be partitioned in a q'-dimensional subvector ψ and q''-dimensional subvector τ. Assume also to be interested in testing the specific values of the vector ψ. Then, one can use some statistical hypothesis tests for testing the null hypothesis $H_0 : \psi = \psi_0$ against the alternative $H_1 : \psi \neq \psi_0$. Composite likelihood versions of 'Wald' and 'score' statistics have the usual asymptotic chi-square distribution with q' degree of freedom. The Wald-type statistic is

$$W = (\hat{\psi} - \psi_0)^T (G^{\psi\psi})^{-1}(\hat{\theta})(\hat{\psi} - \psi_0),$$

where $G_{\psi\psi}$ is the $q' \times q'$ submatrix of the Godambe or Fisher information pertaining to ψ and $\hat{\theta}$ is the maximum likelihood estimator from the full model. This statistic can be called from the routine `GeoTests` assigning at the argument `statistic` the value: `Wald`.

Alternatively to the Wald-type statistic one can use the composite version of the Wilks-type or likelihood ratio statistic, given by

$$W = 2[C\ell(\hat{\theta}; \mathbf{y}) - C\ell\{\psi_0, \hat{\tau}(\psi_0); \mathbf{y}\}].$$

In the composite likelihood case, the asymptotic distribution of the composite likelihood ratio statistic is given by

$$W \dot{\sim} \sum_i \lambda_i \chi^2,$$

for $i = 1, \ldots, q'$, where χ_i^2 are q' iid copies of a chi-square one random variable and $\lambda_1, \ldots, \lambda_{q'}$ are the eigenvalues of the matrix $(H^{\psi\psi})^{-1}G^{\psi\psi}$. There exist several adjustments to the composite likelihood ratio statistic in order to get an approximated $\chi^2_{q'}$. For example, Rotnitzky and Jewell (1990) proposed the adjustment $W' = W/\bar{\lambda}$ where $\bar{\lambda}$ is the average of the eigenvalues λ_i. This statistic can be called within the routine by the value: `WilksRJ`. A better solution is proposed by Satterhwaite (1946) defining $W'' = \nu W/(q'\bar{\lambda})$, where $\nu = (\sum_i \lambda)^2 / \sum_i \lambda_i^2$ for $i = 1 \ldots, q'$, is the effective number of the degree of freedom. Note that in this case the distribution of the likelihood ratio statistic is a chi-square random variable with ν degree of freedom. This statistic can be called from the routine assigning the value: `WilksS`. For the adjustments suggested by Chandler and Bate (2007) and Pace, Salvan and Sartori (2011) we refere to the articles (see **References**), these versions can be called from the routine assigning respectively the values: `WilksCB` and `WilksPSS`.

Value

An object of class `c("data.frame")`. The object contain a table with the results of the tested models. The rows represent the responses for each model and the columns the following results:

`Num.Par`	The number of the model's parameters.
`Diff.Par`	The difference between the number of parameters of the model in the previous row and those in the actual row.
`Df`	The effective number of degree of freedom of the chi-square distribution.
`Chisq`	The observed value of the statistic.
`Pr(>chisq)`	The p-value of the quantile `Chisq` computed using a chi-squared distribution with `Df` degrees of freedom.

Author(s)

Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, `<victor.morales@uv.cl>`, https://sites.google.com/site/moralesonatevictor/

References

Chandler, R. E., and Bate, S. (2007). Inference for Clustered Data Using the Independence log-likelihood. *Biometrika*, **94**, 167–183.

Pace, L., Salvan, A. and Sartori, N. (2011). Adjusting Composite Likelihood Ratio Statistics. *Statistica Sinica*, **21**, 129–148.

Rotnitzky, A. and Jewell, N. P. (1990). Hypothesis Testing of Regression Parameters in Semiparametric Generalized Linear Models for Cluster Correlated Data. *Biometrika*, **77**, 485–497.

Satterthwaite, F. E. (1946). An Approximate Distribution of Estimates of Variance Components. *Biometrics Bulletin*, **2**, 110–114.

Varin, C., Reid, N. and Firth, D. (2011). An Overview of Composite Likelihood Methods. *Statistica Sinica*, **21**, 5–42.

See Also

GeoFit.

Examples

```
library(GeoModels)
################################################################
###
### Example 1. Parametric test of Gaussianity
### using SinhAsinh random fields using standard likelihood
###
###############################################################
set.seed(314)
model="SinhAsinh"
# Define the spatial-coordinates of the points:
NN=500
x = runif(NN, 0, 1)
y = runif(NN, 0, 1)
coords = cbind(x,y)
# Parameters
mean=0; nugget=0; sill=1
### skew and tail parameters
skew=0;tail=1   ## H0 is Gaussianity
# underlying model correlation
corrmodel="Wend0"
power2=4;c_supp=0.2

# simulation
param=list(power2=power2,skew=skew,tail=tail,
             mean=mean,sill=sill,scale=c_supp,nugget=nugget)
data = GeoSim(coordx=coords, corrmodel=corrmodel,model=model, param=param)$data


##### H1
fixed=list(power2=power2,nugget=nugget,mean=mean)
start=list(scale=c_supp,skew=skew,tail=tail,sill=sill)

lower=list(scale=0,skew=-1 ,tail=0.5,sill=0)
upper=list(scale=2,skew=1,tail=1.5,sill=5)
# Maximum pairwise composite-likelihood fitting of the RF:
fitH1 = GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                   likelihood="Full",type="Standard",varest=TRUE,
                   lower=lower,upper=upper,
                   optimizer="nlminb",
```

```
                    start=start,fixed=fixed)

fitH1$param

##### H0: Gaussianity (i.e tail1=1, skew=0 fixed)
fixed=list(power2=power2,nugget=nugget,mean=mean,tail=1,skew=0)
start=list(scale=c_supp,sill=sill)
lower=list(scale=0,sill=0)
upper=list(scale=2,sill=5)
# Maximum pairwise composite-likelihood fitting of the RF:
fitH0 = GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                    likelihood="Full",type="Standard",varest=TRUE,
                      lower=lower,upper=upper,
                   optimizer="nlminb",
                    start=start,fixed=fixed)


#  Wald statistic test
GeoTests(fitH1, fitH0 ,statistic='Wald')
# likelihood ratio statistic test
GeoTests(fitH1, fitH0 , statistic='Wilks')

################################################################
###
### Example 2. Composite likelihood-based hypothesis testing
### for a Gaussian  RF
### Testing significance of a regression parameter
###
###############################################################
set.seed(31)
# Define the spatial-coordinates of the points:
N=1500
x = runif(N, 0, 1)
y = runif(N, 0, 1)
X=cbind(rep(1,N),runif(N))
coords=cbind(x,y)

# Set the model's parameters:
corrmodel = "Exp"
mean = 1;  mean1=0.3 ## H0: mean1=0
sill = 1
nugget = 0
scale = 0.15/3

model="Gaussian"

# Simulation of the spatial Gaussian random field:
data = GeoSim(coordx=coords, corrmodel=corrmodel, model=model,
      param=list(mean=mean,mean1=mean1,sill=sill,
                   nugget=nugget,scale=scale),X=X)$data

# Pairwise-likelihood fitting of the random field, full model:
start=list(mean=mean,mean1=mean1,scale=scale,sill=sill)
fixed=list(nugget=nugget)
fitH1 = GeoFit(data=data,coordx=coords,corrmodel=corrmodel, maxdist=0.05,model=model,
                    varest=TRUE,likelihood="Marginal",type="Pairwise",
                    fixed=fixed,start=start,X=X)
```

```
# Pairwise-likelihood fitting of the random field, with a nested model:
start=list(mean=mean,scale=scale,sill=sill)
fixed=list(nugget=nugget,mean1=0) # setting H0 restriction
fitH0 = GeoFit(data=data,coordx=coords,corrmodel=corrmodel, maxdist=0.05,model=model,
                    varest=TRUE,likelihood="Marginal",type="Pairwise",
                    fixed=fixed,start=start,X=X)


# Hypothesis testing results:
# composite  Wald statistic test:
GeoTests(fitH1, fitH0 ,statistic='Wald')
# composite likelihood ratio statistic test with PSS adjustment:
GeoTests(fitH1, fitH0, statistic='WilksPSS')
```

GeoVarestbootstrap — *Update a* GeoFit *object using parametric bootstrap for std error estimation*

Description

The procedure update a GeoFit object estimating stderr estimation using parametric bootstrap.

Usage

```
GeoVarestbootstrap(fit,K=100,sparse=FALSE,GPU=NULL,local=c(1,1),optimizer="Nelder-Mead",
                      lower=NULL, upper=NULL, memdist=TRUE,seed=1)
```

Arguments

fit	A fitted object obtained from the GeoFit.
K	The number of simulations in the parametric bootstrap.
sparse	Logical; if TRUE then cholesky decomposition is performed using sparse matrices algorithms (spam packake).
GPU	Numeric; if NULL (the default) no OpenCL computation is performed. The user can choose the device to be used. Use DeviceInfo() function to see available devices, only double precision devices are allowed
local	Numeric; number of local work-items of the OpenCL setup
optimizer	The type of optimization algorithm. See GeoFit for details.
lower	An optional named list giving the values for the lower bound of the space parameter when the optimizer is L-BFGS-B or nlminb or optimize.
upper	An optional named list giving the values for the upper bound of the space parameter when the optimizer is L-BFGS-B or nlminb or optimize.
memdist	Logical; if TRUE then the distances in the composite likelihood are computed before the optimization.
seed	Numeric; The seed used in the n-fold kriging cross-validation. Default is 1.

Details

The function update a `GeoFit` object estimating stderr estimation using parametric bootstrap.

Value

Returns an object of class `GeoFit`.

Author(s)

Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`,`https://sites.google.com/a/uv.cl/moreno-bevilacqua/home`, Víctor Morales Oñate, `<victor.morales@uv.cl>`, `https://sites.google.com/site/moralesonatevictor/`

See Also

`GeoFit`.

`GeoVariogram` *Empirical Variogram estimation*

Description

The function returns an empirical estimate of the variogram for spatio (temporal) and bivariate random fields.

Usage

```
GeoVariogram(data, coordx, coordy=NULL, coordt=NULL, coordx_dyn=NULL, cloud=FALSE,
           distance='Eucl', grid=FALSE, maxdist=NULL,neighb=NULL,
           maxtime=NULL, numbins=NULL, radius=6371,
           type='variogram',bivariate=FALSE)
```

Arguments

`data` A d-dimensional vector (a single spatial realisation) or a $(n \times d)$-matrix (n iid spatial realisations) or a $(d \times d)$-matrix (a single spatial realisation on regular grid) or an $(d \times d \times n)$-array (n iid spatial realisations on regular grid) or a $(t \times d)$-matrix (a single spatial-temporal realisation) or an $(t \times d \times n)$-array (n iid spatial-temporal realisations) or or an $(d \times d \times t \times n)$-array (a single spatial-temporal realisation on regular grid) or an $(d \times d \times t \times n)$-array ($n$ iid spatial-temporal realisations on regular grid). See `GeoFit` for details.

`coordx` A numeric $(d \times 2)$-matrix (where `d` is the number of spatial sites) assigning 2-dimensions of spatial coordinates or a numeric d-dimensional vector assigning 1-dimension of spatial coordinates. Coordinates on a sphere for a fixed radius `radius` are passed in lon/lat format expressed in decimal degrees.

`coordy` A numeric vector assigning 1-dimension of spatial coordinates; `coordy` is interpreted only if `coordx` is a numeric vector or `grid=TRUE` otherwise it will be ignored. Optional argument, the default is `NULL` then `coordx` is expected to be numeric a $(d \times 2)$-matrix.

`coordt` A numeric vector assigning 1-dimension of temporal coordinates. Optional argument, the default is `NULL` then a spatial random field is expected.

`coordx_dyn`	A list of m numeric ($d_t \times 2$)-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is `NULL`
`cloud`	Logical; if `TRUE` the variogram cloud is computed, otherwise if `FALSE` (the default) the empirical (binned) variogram is returned.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details** of `GeoFit`.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as spatial or spatial-temporal realisations on a set of non-equispaced spatial sites.
`maxdist`	A numeric value denoting the spatial maximum distance, see the Section **Details**.
`neighb`	Numeric; an optional positive integer indicating the order of neighborhood in the composite likelihood computation. See the Section **Details** for more information.
`maxtime`	A numeric value denoting the temporal maximum distance, see the Section **Details**.
`numbins`	A numeric value denoting the numbers of bins, see the Section **Details**.
`radius`	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is the radius of the earth in Km (i.e. 6371)
`type`	A String denoting the type of variogram. The option available is : `variogram`.
`bivariate`	Logical; if `FALSE` (the default) the data are interpreted as univariate spatial or spatial-temporal realisations. Otherwise they are intrepreted as a a realization from a bivariate field.

Details

We briefly report the definitions of semi-variogram used in this function. In the case of a spatial Gaussian random field the sample `variogram` estimator is defined by

$$\hat{\gamma}(h) = 0.5 \sum_{x_i, x_j \in N(h)} (Z(x_i) - Z(x_j))^2 / |N(h)|$$

where $N(h)$ is the set of all the sample pairs whose distances fall into a tolerance region with size h (equispaced intervalls are considered).

In the case of a spatio-temporal Gaussian random field the sample `variogram` estimator is defined by

$$\hat{\gamma}(h, u) = 0.5 \sum_{(x_i, l), (x_j, k) \in N(h, u)} (Z(x_i, l) - Z(x_j, k))^2 / |N(h, u)|$$

where $N(h, u)$ is the set of all the sample pairs whose spatial distances fall into a tolerance region with size h and $|k - l| = u$. Note, that $Z(x_i, l)$ is the observation at site x_i and time l.

The `numbins` parameter indicates the number of adjacent intervals to consider in order to grouped distances with which to compute the (weighted) lest squares.

The `maxdist` parameter indicates the maximum spatial distance below which the shorter distances will be considered in the calculation of the (weigthed) least squares.

The `maxtime` parameter indicates the maximum temporal distance below which the shorter distances will be considered in the calculation of the (weigthed) least squares.

Value

Returns an object of class `Variogram`. An object of class `Variogram` is a list containing at most the following components:

- `bins` Adjacent intervals of grouped spatial distances if `cloud=FALSE`. Otherwise if `cloud=TRUE` all the spatial pairwise distances;
- `bint` Adjacent intervals of grouped temporal distances if `cloud=FALSE`. Otherwise if `cloud=TRUE` all the temporal pairwise distances;
- `cloud` If the variogram cloud is returned (`TRUE`) or the empirical variogram (`FALSE`);
- `centers` The centers of the spatial bins;
- `distance` The type of spatial distance;
- `lenbins` The number of pairs in each spatial bin;
- `lenbinst` The number of pairs in each spatial-temporal bin;
- `lenbint` The number of pairs in each temporal bin;
- `maxdist` The maximum spatial distance used for the calculation of the variogram. If no spatial distance is specified then it is NULL;
- `maxtime` The maximum temporal distance used for the calculation of the variogram. If no temporal distance is specified then it is NULL;
- `spacetime_dyn` If the space-time variogram is obtained using dynamical coordinates then it is(`TRUE`).
- `variograms` The empirical spatial variogram;
- `variogramst` The empirical spatial-temporal variogram;
- `variogramt` The empirical temporal variogram;
- `type` The type of estimated variogram

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

References

Cressie, N. A. C. (1993) *Statistics for Spatial Data*. New York: Wiley.

Gaetan, C. and Guyon, X. (2010) *Spatial Statistics and Modelling*. Spring Verlang, New York.

See Also

GeoFit

Examples

```
library(GeoModels)

################################################################
###
### Example 1. Empirical estimation of the semi-variogram from a
### spatial Gaussian random field with exponential correlation.
###
###############################################################
```

```
set.seed(514)
# Set the coordinates of the sites:
x = runif(200, 0, 1)
y = runif(200, 0, 1)
coords = cbind(x,y)
# Set the model's parameters:
corrmodel = "exponential"
mean = 0
sill = 1
nugget = 0
scale = 0.3/3

# Simulation of the spatial Gaussian random field:
data = GeoSim(coordx=coords, corrmodel=corrmodel, param=list(mean=mean,
              sill=sill, nugget=nugget, scale=scale))$data

# Empirical spatial semi-variogram estimation:
fit = GeoVariogram(coordx=coords,data=data,maxdist=0.6)

# Results:
plot(fit$centers, fit$variograms, xlab='h', ylab=expression(gamma(h)),
     ylim=c(0, max(fit$variograms)), pch=20,
     main="Semi-variogram")


################################################################
###
### Example 2. Empirical estimation of the variogram from a
### spatio-temporal Gaussian random fields with Gneiting
### correlation function.
###
###############################################################

set.seed(331)
# Define the temporal sequence:
# Set the coordinates of the sites:
x = runif(200, 0, 1)
y = runif(200, 0, 1)
coords = cbind(x,y)
times = seq(1,10,1)

# Simulation of a spatio-temporal Gaussian random field:
data = GeoSim(coordx=coords, coordt=times, corrmodel="gneiting",
              param=list(mean=0,scale_s=0.08,scale_t=0.4,sill=1,
              nugget=0,power_s=1,power_t=1,sep=0.5))$data

# Empirical spatio-temporal semi-variogram estimation:
fit = GeoVariogram(data=data, coordx=coords, coordt=times, maxtime=7,maxdist=0.5)

# Results: Marginal spatial empirical  semi-variogram
par(mfrow=c(2,2), mai=c(.5,.5,.3,.3), mgp=c(1.4,.5, 0))
plot(fit$centers, fit$variograms, xlab='h', ylab=expression(gamma(h)),
     ylim=c(0, max(fit$variograms)), xlim=c(0, max(fit$centers)),
     pch=20,main="Marginal spatial semi-variogram",cex.axis=.8)

# Results: Marginal temporal empirical  semi-variogram
plot(fit$bint, fit$variogramt, xlab='t', ylab=expression(gamma(t)),
```

```
     ylim=c(0, max(fit$variogramt)),xlim=c(0,max(fit$bint)),
     pch=20,main="Marginal temporal semi-variogram",cex.axis=.8)

# Building space-time semi-variogram
st.vario = matrix(fit$variogramst,length(fit$centers),length(fit$bint))
st.vario = cbind(c(0,fit$variograms), rbind(fit$variogramt,st.vario))

# Results: 3d Spatio-temporal semi-variogram
require(scatterplot3d)
st.grid = expand.grid(c(0,fit$centers),c(0,fit$bint))
scatterplot3d(st.grid[,1], st.grid[,2], c(st.vario),
              highlight.3d=TRUE, xlab="h",ylab="t",
              zlab=expression(gamma(h,t)), pch=20,
              main="Space-time semi-variogram",cex.axis=.7,
              mar=c(2,2,2,2), mgp=c(0,0,0),
              cex.lab=.7)

# A smoothed version
par(mai=c(.2,.2,.2,.2),mgp=c(1,.3, 0))
persp(c(0,fit$centers), c(0,fit$bint), st.vario,
      xlab="h", ylab="u", zlab=expression(gamma(h,u)),
      ltheta=90, shade=0.75, ticktype="detailed", phi=30,
      theta=30,main="Space-time semi-variogram",cex.axis=.8,
      cex.lab=.8)

################################################################
###
### Example 3. Empirical estimation of the (cross) semivariograms
### from a bivariate Gaussian random fields with Matern
### correlation function.
###
###############################################################
# Simulation of a bivariate spatial Gaussian random field:
set.seed(293)
# Define the spatial-coordinates of the points:
x = runif(400, 0, 1)
y = runif(400, 0, 1)
coords=cbind(x,y)

# Simulation of a bivariate Gaussian Random field
# with matern (cross)  covariance function
param=list(mean_1=0,mean_2=0,scale_1=0.1/3,scale_2=0.15/3,scale_12=0.15/3,
           sill_1=1,sill_2=1,nugget_1=0,nugget_2=0,
           smooth_1=0.5,smooth_12=0.5,smooth_2=0.5,pcol=0.3)
data = GeoSim(coordx=coords, corrmodel="Bi_matern", param=param)$data

# Empirical  semi-(cross)variogram estimation:
biv_vario=GeoVariogram(data,coordx=coords, bivariate=TRUE,maxdist=c(0.5,0.5,0.5))

# Variograms plots
par(mfrow=c(2,2))
plot(biv_vario$centers,biv_vario$variograms[1,],pch=20,xlab="h",ylim=c(0,1.2),
                                      ylab="",main=expression(gamma[11](h)))
plot(biv_vario$centers,biv_vario$variogramst,pch=20,xlab="h",
                                      ylab="",main=expression(gamma[12](h)))
plot(biv_vario$centers,biv_vario$variogramst,pch=20,xlab="h",ylab="",
                                      main=expression(gamma[21](h)))
```

```
plot(biv_vario$centers,biv_vario$variograms[2,],pch=20,xlab="h",,ylim=c(0,1.2),
                                                 ylab="",main=expression(gamma[22](h)))
```

GeoWLS *WLS of Random Fields*

Description

the function returns the parameters' estimates and the estimates' variances of a random field obtained by the weigthed least squares estimator.

Usage

```
GeoWLS(data, coordx, coordy=NULL, coordt=NULL, coordx_dyn=NULL, corrmodel,
                 distance="Eucl", fixed=NULL, grid=FALSE, maxdist=NULL,neighb=NULL,
                 maxtime=NULL,  model='Gaussian', optimizer='Nelder-Mead',
                 numbins=NULL, radius=6371, start=NULL, weighted=FALSE)
```

Arguments

`data` A d-dimensional vector (a single spatial realisation) or a $(d \times d)$-matrix (a single spatial realisation on regular grid) or an $(d \times d \times n)$-array (n iid spatial realisations on regular grid) or a $(t \times d)$-matrix (a single spatial-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatial-temporal realisation on regular grid). See `GeoFit` for details.

`coordx` A numeric $(d \times 2)$-matrix (where `d` is the number of spatial sites) giving 2-dimensions of spatial coordinates or a numeric d-dimensional vector giving 1-dimension of spatial coordinates.

`coordy` A numeric vector giving 1-dimension of spatial coordinates; `coordy` is interpreted only if `coordx` is a numeric vector or `grid=TRUE` otherwise it will be ignored. Optional argument, the default is `NULL` then `coordx` is expected to be numeric a $(d \times 2)$-matrix.

`coordt` A numeric vector giving 1-dimension of temporal coordinates. Optional argument, the default is `NULL` then a spatial random field is expected.

`coordx_dyn` A list of m numeric $(d_t \times 2)$-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is `NULL`

`corrmodel` String; the name of a correlation model, for the description (see `GeoFit`).

`distance` String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details** of `GeoFit`.

`fixed` A named list giving the values of the parameters that will be considered as known values. The listed parameters for a given correlation function will be not estimated, i.e. if `list(nugget=0)` the nugget effect is ignored.

`grid` Logical; if `FALSE` (the default) the data are interpreted as a vector or a $(n \times d)$-matrix, instead if `TRUE` then $(d \times d \times n)$-matrix is considered.

`maxdist` A numeric value denoting the maximum distance, see **Details** in `GeoFit`.

`neighb` Numeric; an optional positive integer indicating the order of neighborhood. See **Details** and `GeoFit`

maxtime	Numeric; an optional positive value indicating the maximum temporal lag considered.See **Details** and GeoFit.
model	String; the type of random field. Gaussian is the default, see GeoFit for the different types.
optimizer	String; the optimization algorithm (see optim for details). 'Nelder-Mead' is the default.
numbins	A numeric value denoting the numbers of bins, see the Section **Details**
radius	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is the radius of the earth in Km (i.e. 6371)
start	A named list with the initial values of the parameters that are used by the numerical routines in maximization procedure. NULL is the default (see GeoFit).
weighted	Logical; if TRUE then the weighted least square estimator is considered. If FALSE (the default) then the classic least square is used.

Details

The numbins parameter indicates the number of adjacent intervals to consider in order to grouped distances with which to compute the (weighted) lest squares.

The maxdist parameter indicates the maximum distance below which the shorter distances will be considered in the calculation of the (weigthed) least squares.

Value

Returns an object of class WLS. An object of class WLS is a list containing at most the following components:

bins	Adjacent intervals of grouped distances;
bint	Adjacent intervals of grouped temporal separations
centers	The centers of the bins;
coordx	The vector or matrix of spatial coordinates;
coordy	The vector of spatial coordinates;
coordt	The vector of temporal coordinates;
convergence	A string that denotes if convergence is reached;
corrmodel	The correlation model;
data	The vector or matrix of data;
distance	The type of spatial distance;
fixed	The vector of fixed parameters;
iterations	The number of iteration used by the numerical routine;
maxdist	The maximum spatial distance used for the calculation of the variogram used in least square estimation. If no spatial distance is specified then it is NULL;
maxtime	The maximum temporal distance used for the calculation of the variogram used in least square estimation. If no temporal distance is specified then it is NULL;
message	Extra message passed from the numerical routines;
model	The type of random fields;
numcoord	The number of spatial coordinates;
numtime	The number the temporal realisations of the random field;

param	The vector of parameters' estimates;
variograms	The empirical spatial variogram;
variogramt	The empirical temporal variogram;
variogramst	The empirical spatial-temporal variogram;
weighted	A logical value indicating if its the weighted method;
wls	The value of the least squares at the minimum.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

References

Cressie, N. A. C. (1993) *Statistics for Spatial Data*. New York: Wiley.

Gaetan, C. and Guyon, X. (2010) *Spatial Statistics and Modelling*. Spring Verlang, New York.

See Also

GeoFit, optim

Examples

```
library(GeoModels)


# Set the coordinates of the sites:

set.seed(211)
x <- runif(200, 0, 1)
set.seed(98)
y <- runif(200, 0, 1)
coords <- cbind(x,y)

################################################################
###
### Example 1. Least square fitting of a Gaussian random field
### with exponential correlation.
###
###############################################################

# Set the model's parameters:
corrmodel <- "Exponential"
mean <- 0
sill <- 1
nugget <- 0
scale <- 0.15/3
param <- list(mean=0,sill=sill, nugget=nugget, scale=scale)
# Simulation of the Gaussian random field:
set.seed(2)
data <- GeoSim(coordx=coords, corrmodel=corrmodel, param=param)$data

fixed=list(nugget=0,mean=mean)
start=list(scale=scale,sill=sill)
```

```
# Least square fitting of the random field:
fit <- GeoWLS(data=data,coordx=coords, corrmodel=corrmodel,
         fixed=fixed,start=start,maxdist=0.5)

# Results:
print(fit)



################################################################
###
### Example 3. Least square fitting of a spatio-temporal
### Gaussian random field with double exponential correlation.
###
###############################################################

# Define the temporal sequence:
time <- seq(1, 10, 1)
mean <- 0
sill <- 1
scale_s <- 0.15/3
scale_t <- 2/3
param <- list(mean=0,scale_s=scale,scale_t=scale_t,sill=sill,nugget=nugget)
# Simulation of the Gaussian random field:
set.seed(35)
data <- GeoSim(coordx=coords,coordt=time, corrmodel="exp_exp",
             param=param)$data

fixed<-list(nugget=nugget,mean=0)
start<-list(scale_s=scale_s,scale_t=scale_t,sill=1)
# Weighted least square estimation:
fit <- GeoWLS(data=data, coordx=coords,coordt=time, corrmodel="exp_exp",
                 ,maxdist=0.5,maxtime=3,fixed=fixed,start=start)

# Results
print(fit)
```

Lik *Optimizes the Log Likelihood*

Description

Subroutine called by GeoFit. The procedure estimates the model parameters by maximization of the log-likelihood.

Usage

```
Lik(bivariate,coordx,coordy,coordt,coordx_dyn,corrmodel,data,fixed,flagcor,flagnuis,
            grid,lower,mdecomp,model,namescorr,namesnuis,namesparam,numcoord,
        numpairs,numparamcor,numtime,optimizer,onlyvar,parallel,param,radius,setup,
            spacetime,sparse,varest,taper,type,upper,ns,X)
```

Arguments

bivariate	Logical; if TRUE then the data come froma a bivariate random field. Otherwise from a univariate random field.
coordx	A numeric ($d \times 2$)-matrix (where d is the number of spatial sites) assigning 2-dimensions of spatial coordinates or a numeric d-dimensional vector assigning 1-dimension of spatial coordinates.
coordy	A numeric vector assigning 1-dimension of spatial coordinates; coordy is interpreted only if coordx is a numeric vector or grid=TRUE otherwise it will be ignored. Optional argument, the default is NULL then coordx is expected to be numeric a ($d \times 2$)-matrix.
coordt	A numeric vector assigning 1-dimension of temporal coordinates. Optional argument, the default is NULL then a spatial random field is expected.
coordx_dyn	A list of m numeric ($d_t \times 2$)-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is NULL
corrmodel	Numeric; the id of the correlation model.
data	A numeric vector or a ($n \times d$)-matrix or ($d \times d \times n$)-matrix of observations.
flagcor	A numeric vector of flags denoting which correlation parameters have to be estimated.
flagnuis	A numeric verctor of flags denoting which nuisance parameters have to estimated.
fixed	A numeric vector of parameters that will be considered as known values.
grid	Logical; if FALSE (the default) the data are interpreted as a vector or a ($n \times d$)-matrix, instead if TRUE then ($d \times d \times n$)-matrix is considered.
lower	An optional named list giving the values for the lower bound of the space parameter when the optimizer is L-BFGS-B or nlminb or optimize. The names of the list must be the same of the names in the start list.
model	Numeric; the id value of the density associated to the likelihood objects.
namescorr	String; the names of the correlation parameters.
namesnuis	String; the names of the nuisance parameters.
namesparam	String; the names of the parameters to be maximised.
numcoord	Numeric; the number of coordinates.
numpairs	Numeric; the number of pairs.
numparamcor	Numeric; the number of the correlation parameters.
numtime	Numeric; the number of temporal observations.
mdecomp	String; the type of matrix decomposition used in the simulation. Default is cholesky. The other possible choices is svd (Singular values decomposition).
optimizer	String; the optimization algorithm (see optim for details). Nelder-Mead is the default. Other possible choices are ucminf,nlm, BFGS L-BFGS-B and nlminb. In these last two cases upper and lower bounds can be passed by the user. In the case of one-dimensional optimization, the function optimize is used.
onlyvar	Logical; if TRUE (and varest is TRUE) only the variance covariance matrix is computed without optimizing. FALSE is the default.
parallel	Logical; if TRUE optmization is performed using optimParallel using the maximum number of cores, when optimizer is L-BFGS-B.FALSE is the default.

`param`	A numeric vector of parameters.
`sparse`	Logical; if `TRUE` then maximum likelihood is computed using sparse matrices algorithms.`FALSE` is the default.
`radius`	Numeric; the radius of the sphere when considering data on a sphere.
`ns`	Numeric: vector of number of location sites for each temporal instants
`setup`	A List of useful components for the estimation based on the maximum tapered likelihood.
`spacetime`	Logical; if the random field is spatial (`FALSE`) or spatio-temporal (`TRUE`).
`varest`	Logical; if `TRUE` the estimate' variances and standard errors are returned. `FALSE` is the default.
`taper`	String; the name of the taper correlation function.
`type`	String; the type of the likelihood objects. If `Pairwise` (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods.
`upper`	An optional named list giving the values for the upper bound of the space parameter when the optimizer is or `L-BFGS-B` or `nlminb` or `optimize`. The names of the list must be the same of the names in the `start` list.
`X`	Numeric; Matrix of spatio(temporal)covariates in the linear mean specification.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

`GeoFit`

`MatDecomp` *Matrix decomposition*

Description

Matrix decomposition.

Usage

```
MatDecomp(mtx, method)
```

Arguments

`mtx`	numeric; a square positive or semipositive definite matrix.
`method`	string; the type of matrix decomposition. Two possible choices: `cholesky` and `svd`.

Details

Decomposition of a square positive or positive semidefinite matrix.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

MatSqrt, MatInv, MatLogDet
Square root, inverse and log determinant of a (semi)positive definite matrix, given a matrix decomposition.

Description

Square root, inverse and log determinant of a (semi)positive definite matrix, given a matrix decomposition.

Usage

```
MatSqrt(mat.decomp,method)
MatInv(mat.decomp,method)
MatLogDet(mat.decomp,method)
```

Arguments

`mat.decomp`	numeric; a matrix decomposition.
`method`	string; the type of matrix decomposition. Two possible choices: `cholesky` and `svd`.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

MatDecomp

Examples

```
library(GeoModels)
################################################################
###
### Example 1. Inverse of Covariance matrix associated to
### a Matern correlation model
###
###############################################################
# Define the spatial-coordinates of the points:
x <- runif(15, 0, 1)
y <- runif(15, 0, 1)
coords <- cbind(x,y)
# Matern Parameters
param=list(smooth=0.5,sill=1,scale=0.2,nugget=0)
a=matrix <- GeoCovmatrix(coordx=coords, corrmodel="Matern", param=param)
```

```
## decomposition with cholesky method
b=MatDecomp(a$covmat,method="cholesky")
## inverse of covariance matrix
inverse=MatInv(b,method="cholesky")
```

NuisParam *Lists the Nuisance Parameters of a Random Field*

Description

The procedure returns a list with the nuisance parameters of a given random field model.

Usage

```
NuisParam(model, bivariate=FALSE,num_betas=c(1,1))
```

Arguments

model	String; the name of a random field.
bivariate	Logical; if FALSE (the default) the correlation model is univariate spatial or spatial-temporal. Otherwise is bivariate.
num_betas	Numerical; the nunber of mean parameters in the linear specification (default is 1)

Details

The function returns a list with the nuisance parameters of a given random field model.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit

Examples

```
library(GeoModels)

NuisParam("Gaussian")

NuisParam("Binomial")

NuisParam("Weibull",num_betas=2)

NuisParam("SkewGaussian", num_betas=3)

NuisParam("SinhAsinh")

NuisParam("StudentT")
```

```
## note that in the bivariate case sill and nugget are considered as correlation parameteres
NuisParam("Gaussian", bivariate=TRUE)
```

Prscores   *Computation of three predictive scores: RMSE, LSCORE, CRPS for spatial, spatiotemporal and bivariate Gaussian RF.*

Description

The function computes RMSE, LSCORE, CRPS predictive scores.

Usage

```
Prscores(data,  method="cholesky", matrix)
```

Arguments

data	A d-dimensional vector (a single spatial realisation) or a a$(t \times d)$-matrix (a single spatial-temporal realisation). or a a$(2 \times d)$-matrix (a single bivariate realisation).
method	String; the type of matrix decomposition used in the computation of the predictive scores. Default is cholesky. The other possible choices is svd.
matrix	An object of class matrix. See the Section **Details**.

Details

For a given covariance matrix object (GeoCovmatrix) and a given spatial, spatiotemporal or bivariare realization from a Gaussian random field, the function computes three predictive scores.

Value

Returns a list containing the following informations:

RMSE	Root-mean-square error predictive score
MAE	Mean absolute error predictive score
LSCORE	Logarithmic predictive score
CRPS	Continuous ranked probability predictive score

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

References

Zhang H. and Wang Y. (2010). *Kriging and cross-validation for massive spatial data*. Environmetrics, **21**, 290–304. Gneiting T. and Raftery A. *Strictly Proper Scoring Rules, Prediction, and Estimation*. Journal of the American Statistical Association, **102**

See Also

GeoCovmatrix

Examples

```
library(GeoModels)
library(fields)

################################################################
######### Examples of predictive score computation  ############
################################################################

  # Define the spatial-coordinates of the points:
x <- runif(500, 0, 2)
y <- runif(500, 0, 2)
coords=cbind(x,y)
matrix1 <- GeoCovmatrix(coordx=coords, corrmodel="Matern", param=list(smooth=0.5,
                    sill=1,scale=0.2,nugget=0))

data <- GeoSim(coordx=coords, corrmodel="Matern", param=list(mean=0,smooth=0.5,
                    sill=1,scale=0.2,nugget=0))$data

Pr_scores <- Prscores(data,matrix=matrix1)

Pr_scores
```

StartParam   *Initializes the Parameters for Estimation Procedures*

Description

Subroutine called by the fitting procedures. The procedure initializes the parameters for the fitting procedure.

Usage

```
StartParam(coordx, coordy, coordt,coordx_dyn, corrmodel, data, distance, fcall,
          fixed, grid,likelihood,  maxdist, neighb, maxtime, model, n,  param,
          parscale, paramrange, radius,  start, taper, tapsep,
          type,typereal,varest, vartype, weighted, winconst,
          winstp, winconst_t, winstp_t,X,memdist)
```

Arguments

coordx	A numeric ($d \times 2$)-matrix (where d is the number of points) assigning 2-dimensions of coordinates or a numeric vector assigning 1-dimension of coordinates.
coordy	A numeric vector assigning 1-dimension of coordinates; coordy is interpreted only if coordx is a numeric vector otherwise it will be ignored.
coordt	A numeric vector assigning 1-dimension of temporal coordinates.
coordx_dyn	A list of m numeric ($d_t \times 2$)-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is NULL
corrmodel	String; the name of a correlation model.
data	A numeric vector or a ($n \times d$)-matrix or ($d \times d \times n$)-matrix of observations.

distance	String; the name of the spatial distance. The default is Eucl, the euclidean distance. See the Section **Details**.
fcall	String; "fitting" to call the fitting procedure and "simulation" to call the simulation procedure.
fixed	A named list giving the values of the parameters that will be considered as known values.
grid	Logical; if FALSE (the default) the data are interpreted as a vector or a $(n \times d)$-matrix, instead if TRUE then $(d \times d \times n)$-matrix is considered.
likelihood	String; the configuration of the composite likelihood.
maxdist	Numeric; an optional positive value indicating the maximum spatial distance considered in the composite-likelihood computation.
neighb	Numeric; an optional positive integer indicating the order of neighborhood in the composite likelihood computation. See the Section **Details** for more information.
maxtime	Numeric; an optional positive value indicating the maximum temporal lag considered in the composite-likelihood computation.
radius	Numeric; the radius of the sphere in the case of lon-lat coordinates. The default is 6371, the radius of the earth.
model	String; the density associated to the likelihood objects. Gaussian is the default.
n	Numeric; number of trials for binomial random fields.
param	A numeric vector of parameter values required in the simulation procedure of random fields.
parscale	A numeric vector of scaling factor to improve the maximizing procedure, see optim.
paramrange	A numeric vector of parameters ranges, see optim.
start	A named list with the initial values of the parameters that are used by the numerical routines in maximization procedure.
taper	String; the name of the type of covariance matrix. It can be Standard (the default value) or Tapering for taperd covariance matrix.
tapsep	Numeric; an optional value indicating the separabe parameter in the space time adaptive taper (see **Details**).
type	String; the type of likelihood objects. Temporary value set to be "WLeastSquare" (weigthed least-square) in order to compute the starting values.
typereal	String; the real type of likelihood objects. See GeoFit.
varest	Logical; if TRUE the estimates' variances and standard errors are returned. FALSE is the default.
vartype	String; the type of estimation method for computing the estimate variances, see the Section **Details**.
weighted	Logical; if TRUE the likelihood objects are weighted, see GeoFit.
winconst	Numeric; a positive value for computing the spatial sub-window in the subsampling procedure.
winstp	Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the spatial sub-sampling procedure.
winconst_t	Numeric; a positive value for computing the temporal sub-window in the subsampling procedure.

winstp_t	Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the temporal sub-sampling procedure.
X	Numeric; Matrix of space-time covariates.
memdist	Logical; if TRUE then the distances in the composite likelihood are computed before the optmization.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

GeoFit

---

winds *Irish Daily Wind Speeds*

Description

A matrix containing daily wind speeds, in kilometers per hour, from 1961 to 1978 at 12 sites in Ireland.

Usage

```
data(irishwinds)
```

Format

A (6574×11)-matrix containing wind speed observations.

Source

Haslett, J. and Raftery, A. E. (1989), Space-time modelling with long-memory dependence: assessing Ireland's wind-power resource (with discussion), *Applied Statistics*, 38, 1–50.

---

winds.coords *Weather Stations of the Irish Daily Wind Speeds*

Description

A data frame containing information about the weather stations where the data are recorded in Ireland.

Usage

```
data(irishwinds)
```

Format

A data frame containing site - the name of the city (character), abbr - the abbrevation (character), elev - the elevation (numeric), lat - latitude (numeric) and lon - longitude.

Source

Haslett, J. and Raftery, A. E. (1989), Space-time modelling with long-memory dependence: assessing Ireland's wind-power resource (with discussion), *Applied Statistics*, 38, 1–50.

WlsStart *Computes Starting Values based on Weighted Least Squares*

Description

Subroutine called by GeoFit. The function returns opportune starting values for the composite-likelihood fitting procedure based on weigthed least squares.

Usage

```
WlsStart(coordx, coordy, coordt, coordx_dyn, corrmodel, data, distance, fcall,
        fixed, grid,likelihood, maxdist, neighb,maxtime, model, n, param,
        parscale, paramrange, radius,start, taper, tapsep, type, varest,
        vartype, weighted, winconst,winconst_t, winstp_t, winstp,X,memdist)
```

Arguments

`coordx`	A numeric ($d \times 2$)-matrix (where d is the number of points) assigning 2-dimensions of coordinates or a numeric vector assigning 1-dimension of coordinates.
`coordy`	A numeric vector assigning 1-dimension of coordinates; `coordy` is interpreted only if `coordx` is a numeric vector otherwise it will be ignored.
`coordt`	A numeric vector assigning 1-dimension of temporal coordinates.
`coordx_dyn`	A list of m numeric ($d_t \times 2$)-matrices containing dynamical (in time) spatial coordinates. Optional argument, the default is `NULL`
`corrmodel`	String; the name of a correlation model, for the description.
`data`	A numeric vector or a ($n \times d$)-matrix or ($d \times d \times n$)-matrix of observations.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details**.
`fcall`	String; "fitting" to call the fitting procedure and "simulation" to call the simulation procedure.
`fixed`	A named list giving the values of the parameters that will be considered as known values.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as a vector or a ($n \times d$)-matrix, instead if `TRUE` then ($d \times d \times n$)-matrix is considered.
`likelihood`	String; the configuration of the composite likelihood.
`maxdist`	Numeric; an optional positive value indicating the maximum spatial distance considered in the composite-likelihood computation.

neighb	Numeric; an optional positive integer indicating the order of neighborhood in the composite likelihood computation. See the Section **Details** for more information.
maxtime	Numeric; an optional positive value indicating the maximum temporal separation considered in the composite-likelihood computation.
model	String; the name of the model. Here the default is `NULL`.
n	Numeric; number of trials in a binomial random field.
param	A numeric vector of parameter values required in the simulation procedure of random fields.
parscale	A numeric vector with scaling values for improving the maximisation routine.
paramrange	A numeric vector with the range of the parameter space.
radius	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is the radius of the earth in Km (i.e. 6371)
start	A numeric vector with starting values.
taper	String; the name of the type of covariance matrix. It can be `Standard` (the default value) or `Tapering` for taperd covariance matrix.
tapsep	Numeric; an optional value indicating the separabe parameter in the space time quasi taper (see **Details**).
type	String; the type of estimation method.
varest	Logical; if `TRUE` the estimates' variances and standard errors are returned. `FALSE` is the default.
vartype	String; the type of estimation method for computing the estimate variances, see the Section **Details**.
weighted	Logical; if `TRUE` the likelihood objects are weighted, see `GeoFit`.
winconst	Numeric; a positive value for computing the spatial sub-window in the sub-sampling procedure.
winstp	Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the spatial sub-sampling procedure.
winconst_t	Numeric; a positive value for computing the temporal sub-window in the sub-sampling procedure.
winstp_t	Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the temporal sub-sampling procedure.
X	Numeric; Matrix of spatio(temporal)covariates in the linear mean specification.
memdist	Logical; if `TRUE` then the distances in the composite likelihood are computed before the optmization.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

See Also

`GeoFit`.

Index

∗ **Composite**
CheckBiv, 3
CheckDistance, 4
CheckSph, 5
CheckST, 5
CkCorrModel, 6
CkInput, 7
CkLikelihood, 8
CkModel, 9
CkType, 10
CkVarType, 10
CompLik, 11
CompLik2, 13
CorrelationPar, 15
CorrParam, 16
GeoCovariogram, 18
GeoCV, 34
GeoFit, 36
GeoKrig, 51
GeoKrigloc, 61
GeoQQ, 69
GeoResiduals, 70
GeoVarestbootstrap, 84
Lik, 93
MatDecomp, 95
MatSqrt, MatInv, MatLogDet, 96
NuisParam, 97
StartParam, 99
∗ **Devices**
DeviceInfo, 17
∗ **LeastSquare**
GeoWLS, 90
WlsStart, 102
∗ **Predictive scores**
Prscores, 98
∗ **Scatterplot**
GeoScatterplot, 72
∗ **Simulation**
GeoCovmatrix, 24
GeoSim, 74
∗ **Sparsness pattern**
GeoCovDisplay, 23
∗ **Variogram**
GeoVariogram, 85
∗ **datasets**
anomalies, 2
austemp, 3
winds, 101
winds.coords, 101
∗ **spatial**
GeoTests, 80

anomalies, 2
austemp, 3

CheckBiv, 3
CheckDistance, 4
CheckSph, 5
CheckST, 5
CkCorrModel, 6
CkInput, 7
CkLikelihood, 8
CkModel, 9
CkType, 10
CkVarType, 10
CompLik, 11
CompLik2, 13
CorrelationPar, 15
CorrParam, 16, *25*, *40*, *52*

DeviceInfo, 17

GeoCovariogram, 18
GeoCovDisplay, 23
GeoCovmatrix, *4*, *6*, *16*, *23*, 24, *39, 40*, *52–54*, *63*, *75*, *98*
GeoCV, 34
GeoFit, *7–15*, *18, 19*, *24, 25*, *31*, *34*, 36, *51*, *53*, *61*, *63*, *67*, *71, 72*, *74, 75*, *82*, *84–87*, *90–92*, *95*, *97*, *100, 101*, *103*
GeoKrig, *31*, *34, 35*, 50, *62*
GeoKrigloc, 61
GeoNeighborhood, *61, 62*, 66
GeoQQ, 69
GeoResiduals, *69*, 70
GeoScatterplot, 72
GeoSim, *8*, *31*, 74

GeoTests, 80
GeoVarestbootstrap, 84
GeoVariogram, *18*, 85
GeoWLS, *18*, 90

Lik, 93

MatDecomp, 95, *96*
MatInv (MatSqrt, MatInv, MatLogDet), 96
MatLogDet (MatSqrt, MatInv, MatLogDet), 96
MatSqrt (MatSqrt, MatInv, MatLogDet), 96
MatSqrt, MatInv, MatLogDet, 96

NuisParam, *40*, 97

optim, *8*, *12*, *14*, *37*, *91*, *92*, *94*, *100*

print.GeoFit (GeoFit), 36
print.GeoSim (GeoSim), 74
print.GeoWLS (GeoWLS), 90
Prscores, 98

StartParam, 99

winds, 101
winds.coords, 101
WlsStart, 102