Package 'GeoModels'

May 7, 2026

Version 2.2.3

Date 2026-03-31

Depends R (>= 4.1.0), fields, mapproj, shape,progressr,future.apply

Suggests numDeriv, memuse

Description

Functions for Gaussian and Non Gaussian (bivariate) spatial and spatio-temporal data analysis are provided for a) (fast) simulation of random fields, b) inference for random fields using standard likelihood and a likelihood approximation method called weighted composite likelihood based on pairs and b) prediction using (local) best linear unbiased prediction. Weighted composite likelihood can be very efficient for estimating massive datasets. Both regression and spatial (temporal) dependence analysis can be jointly performed. Flexible covariance models for spatial and spatial-temporal data on Euclidean domains and spheres are provided. There are also many useful functions for plotting and performing diagnostic analysis. Different non Gaussian random fields can be considered in the analysis. Among them, random fields with marginal distributions such as Skew-Gaussian, Student-t, Tukey-h, Sin-Arcsin, Two-piece, Weibull, Gamma, Log-Gaussian, Binomial, Negative Binomial and Poisson. See the URL for the papers associated with this package, as for instance, Bevilacqua and Gaetan (2015) <doi:10.1007/s11222-014-9460-6>, Bevilacqua et al. (2016) <doi:10.1007/s13253-016-0256-3>, Vallejos et al. (2020) <doi:10.1007/978-3-030-56681-4>, Bevilacqua et. al (2020) <doi:10.1002/env.2632>, Bevilacqua et. al (2021) <doi:10.1111/sjos.12447>, Bevilacqua et al. (2022) <doi:10.1016/j.jmva.2022.104949>, Morales-Navarrete et al. (2023) <doi:10.1080/01621459.2022.2140053>, and a large class of examples and tutorials.

Title Procedures for Gaussian and Non Gaussian Geostatistical (Large) Data Analysis

License GPL (>= 3)

Encoding UTF-8

Imports methods, spam, scatterplot3d, dotCall64, FastGP, plotrix, pracma, pbivnorm, sn, sp, nabor, hypergeo, VGAM, foreach, future, doFuture,minqa,withr

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

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

NeedsCompilation yes

Author Moreno Bevilacqua [aut, cre, cph],
Víctor Morales-Oñate [ctb],
Francisco Cuevas-Pacheco [ctb],
Christian Caamaño-Carrillo [ctb]

Maintainer Moreno Bevilacqua <moreno.bevilacqua89@gmail.com>

Repository CRAN

Date/Publication 2026-04-06 14:20:08 UTC

Contents

`anomalies` *Annual Precipitation Anomalies in the U.S.*

Description

A numerical matrix of dimension 7252×3 containing longitude, latitude, and yearly total precipitation anomalies registered at 7,352 location sites in the USA.

Usage

```
data(anomalies)
```

Format

A numeric matrix with 7,252 rows and 3 columns:

Column 1 Longitude

Column 2 Latitude

Column 3 Annual precipitation anomaly

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.

CheckBiv *Checking Bivariate Covariance Models*

Description

Checks whether the correlation model is bivariate.

Usage

```
CheckBiv(numbermodel)
```

Arguments

`numbermodel` A numeric value; the number associated with a given correlation model.

Details

This function checks whether the correlation model is bivariate.

Value

A logical value: `TRUE` if the correlation model is bivariate, and `FALSE` otherwise.

Author(s)

Moreno Bevilacqua <moreno.bevilacqua89@gmail.com>
https://sites.google.com/view/moreno-bevilacqua/home
Víctor Morales Oñate <victor.morales@uv.cl>
https://sites.google.com/site/moralesonatevictor/
Christian Caamaño-Carrillo <chcaaman@ubiobio.cl>
https://www.researchgate.net/profile/Christian-Caamano

Examples

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

CheckDistance *Checking Distance Type*

Description

Checks the validity and type of the specified distance.

Usage

```
CheckDistance(distance)
```

Arguments

distance	A character string indicating the type of distance. Available options are: "Eucl" (Euclidean), "Geod" (Geodesic), and "Chor" (Chordal). See also GeoCovmatrix.

Details

This function checks whether the specified distance type is valid.

Value

An integer:

- 0 for Euclidean distance
- 1 for Geodesic distance
- 2 for Chordal distance

If the input is not recognized, the function returns NULL.

Author(s)

Moreno Bevilacqua <moreno.bevilacqua89@gmail.com>
https://sites.google.com/view/moreno-bevilacqua/home
Víctor Morales Oñate <victor.morales@uv.cl>
https://sites.google.com/site/moralesonatevictor/
Christian Caamaño-Carrillo <chcaaman@ubiobio.cl>
https://www.researchgate.net/profile/Christian-Caamano

`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.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

`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.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

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.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

---

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, coordz,coordt, coordx_dyn,corrmodel, data, distance,
           fcall, fixed, grid,likelihood, maxdist, maxtime,
            model, n,  optimizer, param, radius,
           start, taper, tapsep,  type, varest,
           weighted,copula,X)
```

Arguments

coordx	A numeric $(d \times 2)$-matrix or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.
coordz	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is `NULL`.
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.
`weighted`	Logical; if `TRUE` the likelihood objects are weighted. If `FALSE` (the default) the composite likelihood is not weighted.
`copula`	String; the type of copula. It can be "Clayton" or "Gaussian"
`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.

Value

A list with the type of error associated with the input parameters.

Author(s)

Moreno Bevilacqua, `<moreno.bevilacqua89@gmail.com>`,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, `<victor.morales@uv.cl>`, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, `<chcaaman@ubiobio.cl>`,https://www.researchgate.net/profile/Christian-Caamano

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.

Value

The function returns a numeric positive integer, or NULL if the likelihood is invalid.

Author(s)

Moreno Bevilacqua, `<moreno.bevilacqua89@gmail.com>`,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, `<victor.morales@uv.cl>`, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, `<chcaaman@ubiobio.cl>`,https://www.researchgate.net/profile/Christian-Caamano

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 with the likelihood objects. `Gaussian` is the default.

Details

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

Value

The function returns a numeric positive integer, or `NULL` if the model is invalid.

Author(s)

Moreno Bevilacqua <moreno.bevilacqua89@gmail.com> https://sites.google.com/view/moreno-bevilacqua/home \ Víctor Morales Oñate <victor.morales@uv.cl> https://sites.google.com/site/moralesonatevictor/ \ Christian Caamaño-Carrillo <chcaaman@ubiobio.cl> https://www.researchgate.net/profile/Christian-Caamano

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.

Value

The function returns a numeric positive integer, or `NULL` if the type of likelihood is invalid.

Author(s)

Moreno Bevilacqua <moreno.bevilacqua89@gmail.com> https://sites.google.com/view/moreno-bevilacqua/home \ Víctor Morales Oñate <victor.morales@uv.cl> https://sites.google.com/site/moralesonatevictor/ \ Christian Caamaño-Carrillo <chcaaman@ubiobio.cl> https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit

CompIndLik2 *Optimizes the Composite indipendence log-likelihood*

Description

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

Usage

```
CompIndLik2(bivariate, coordx, coordy ,coordz,coordt,
coordx_dyn, data, flagcorr, flagnuis, fixed,grid,
              lower, model, n, namescorr, namesnuis,
              namesparam,
              numparam, optimizer, onlyvar,
              param, spacetime, type,
              upper, namesupper, varest, ns, X,
              sensitivity,copula,MM,score)
```

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 or $(d \times 3)$-matrix 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; Optional argument, the default is NULL.
coordz	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is NULL.
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
data	A numeric vector or a $(n \times d)$-matrix or $(d \times d \times n)$-matrix of observations.

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.
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.
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.
optimizer	String; the optimization algorithm (see optim for details). Nelder-Mead is the default. Other possible choices are 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.
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.
namesupper	String; the names of the upper limit of the parameters.
varest	Logical; if TRUE the estimate variances and standard errors are returned. FALSE is the default.
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
copula	String; the type of copula. It can be "Clayton" or "Gaussian"
MM	Numeric;a non constant fixed mean
score	Logical; should score function be computed?

Value

Return a list from an optim call.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit

CompLik *Optimizes the Composite Log-likelihood*

Description

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

Usage

```
CompLik(copula, bivariate, coordx, coordy, coordz, coordt,
        coordx_dyn, corrmodel, data, distance, flagcorr,
        flagnuis, fixed, grid, likelihood,  lower,
        model, n, namescorr, namesnuis, namesparam,
        numparam, numparamcorr, optimizer,
        onlyvar,  param,
        spacetime, type, upper, varest,
        weigthed, ns, X, sensitivity, MM, aniso,score)
```

Arguments

copula	String; the type of copula. It can be "Clayton" or "Gaussian".
bivariate	Logical; if TRUE then the data come from a bivariate random field, otherwise from a univariate random field.
coordx	A numeric $d \times 2$ or $d \times 3$ matrix. Coordinates on a sphere for a fixed radius radius are passed in lon/lat format expressed in decimal degrees.
coordy	A numeric vector giving one dimension of spatial coordinates; optional, default is NULL.
coordz	A numeric vector giving one dimension of spatial coordinates; optional, default is NULL.
coordt	A numeric vector giving one dimension of temporal coordinates; optional, default is NULL (in which case a spatial random field is assumed).
coordx_dyn	A list of m numeric $d_t \times 2$ matrices containing dynamic (in time) spatial coordinates; optional, default is NULL.
corrmodel	Numeric; the ID of the correlation model.

`data`	A numeric vector, or a $n \times d$ matrix, or a $d \times d \times n$ array of observations.
`distance`	String; the name of the spatial distance. Default is `"Eucl"` (Euclidean distance). See Details.
`flagcorr`	Numeric vector of binary values indicating which parameters of the correlation function will be estimated.
`flagnuis`	Numeric vector of binary values indicating which nuisance parameters will be estimated.
`fixed`	Numeric vector of parameters considered as known values.
`grid`	Logical; if `FALSE` (default), data are interpreted as vector or $n \times d$ matrix; if `TRUE`, then a $d \times d \times n$ array is considered.
`likelihood`	String; configuration of the composite likelihood (see `GeoFit`).
`lower`	Named list; optional lower bounds for parameters when using optimizers `L-BFGS-B`, `nlminb`, or `optimize`. Names must match those in the `start` list.
`model`	Numeric; ID of the density associated with the likelihood objects.
`n`	Numeric; number of trials in binomial random fields.
`namescorr`	Character vector; names of the correlation parameters.
`namesnuis`	Character vector; names of the nuisance parameters.
`namesparam`	Character vector; names of the parameters to be maximized.
`numparam`	Numeric; number of parameters to be maximized.
`numparamcorr`	Numeric; number of correlation parameters.
`optimizer`	String; optimization algorithm (see `optim`). Default is `"Nelder-Mead"`. Other options: `"nlm"`, `"BFGS"`, `"L-BFGS-B"`, `"nlminb"`. For `"L-BFGS-B"` and `"nlminb"` bounds can be provided. For 1D optimization, `optimize` is used.
`onlyvar`	Logical; if `TRUE` (and `varest` is `TRUE`), only the variance-covariance matrix is computed without optimizing. Default is `FALSE`.
`param`	Numeric vector of parameter values.
`spacetime`	Logical; if `TRUE`, the random field is spatio-temporal, otherwise spatial.
`type`	String; type of likelihood object. Default is `"Pairwise"` (marginal composite likelihood formed by pairwise marginal likelihoods).
`upper`	Named list; optional upper bounds for parameters when using optimizers `L-BFGS-B`, `nlminb`, or `optimize`. Names must match those in the `start` list.
`varest`	Logical; if `TRUE`, variance estimates and standard errors are returned. Default is `FALSE`.
`weigthed`	Logical; if `TRUE`, decreasing weights from a compactly supported correlation function with compact support `maxdist` (`maxtime`) are used.
`ns`	Numeric; number of (dynamic) temporal instants.
`X`	Numeric; matrix of space-time covariates in the linear mean specification.
`sensitivity`	Logical; if `TRUE`, the sensitivity matrix is computed.
`MM`	Numeric; a non-constant fixed mean.
`aniso`	Logical; whether anisotropy should be considered.
`score`	Logical; should score function be computed?

Details

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

Value

Returns a list from an optim call.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>, https://sites.google.com/view/moreno-bevilacqua/home,
Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/,
Christian Caamaño-Carrillo, <chcaaman@ubiobio.cl>, https://www.researchgate.net/profile/Christian-Caamano

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(copula,bivariate, coordx, coordy ,coordz,coordt,
coordx_dyn,corrmodel, data, distance, flagcorr, flagnuis,
         fixed,grid,likelihood, lower,
         model, n, namescorr, namesnuis, namesparam,
         numparam, numparamcorr, optimizer, onlyvar,
          param, spacetime, type,
         upper, varest, weigthed, ns, X,sensitivity,
         colidx,rowidx,neighb,MM,aniso,score)
```

Arguments

copula	String; the type of copula. It can be "Clayton" or "Gaussian"
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 or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.
`coordz`	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is `NULL`.
`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.
`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`.
`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 `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.
`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.
weigthed	Logical; if TRUE then decreasing weigths coming from a compactly supported correlation function with compact support maxdist (maxtime)are used.
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.
MM	Numeric;a non constant fixed mean
aniso	Logical; should anisotropy be considered?
score	Logical; should score function be computed?

Value

Return a list from an optim call.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

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

Value

Return a vector string of correlation parameters.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit

CorrParam — *Lists the Parameters of a Correlation Model*

Description

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

Usage

```
CorrParam(corrmodel)
```

Arguments

corrmodel String: the name associated to a given correlation model.

Details

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

Value

Return a vector string of correlation parameters.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

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 Cauchy  model
###
###############################################################

CorrParam("GenCauchy")


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

CorrParam("Gneiting")


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

```
###           Note that in the bivariate case variance paramters are
###           included
###############################################################

CorrParam("Bi_Matern_sep")
```

`corrsas` *Correlation Function for Sinh-Arcsinh Random Fields*

Description

Computes the correlations f for a random field transformed via the sinh-arcsinh (SAS) distribution. This transformation introduces flexible skewness and tail behavior to an underlying Gaussian field. The resulting correlation is derived via an infinite Hermite expansion, as described in Equation (16) of Blasi et al. (2022).

Usage

```
corrsas(corr, skew, tail, max_coeff = NULL)
```

Arguments

`corr`	A numeric vector of correlation values of the underlying standard Gaussian random field.
`skew`	A numeric value representing the skewness parameter α of the sinh-arcsinh transformation. Positive values induce right-skewness, negative values left-skewness.
`tail`	A positive numeric value representing the tailweight parameter κ. Values less than 1 yield heavier tails than Gaussian, while values greater than 1 produce lighter tails.
`max_coeff`	Optional integer. The maximum number of Hermite coefficients used in the infinite series expansion. If `NULL`, a default truncation value is used internally.

Details

The correlation of the sinh-arcsinh transformed field is computed as:

$$\rho_{SAS}(h) = \sum_{j=1}^{\infty} \frac{\xi_j^2(\alpha, \kappa)}{j!} \rho(h)^j$$

where $\rho(h)$ is the correlation function of the underlying Gaussian field and $\xi_j(\alpha, \kappa)$ are Hermite coefficients depending on the skewness and tail parameters. This series is truncated at `max_coeff` terms for computational feasibility.

See Equation (16) in Blasi et al. (2022) for the full derivation.

Value

A numeric vector of adjusted correlation values corresponding to the SAS-transformed process.

References

Blasi, F., Caamaño-Carrillo, C., Bevilacqua, M., Furrer, R. (2022). A selective view of climatological data and likelihood estimation. *Spatial Statistics*, 50, 100596. doi:10.1016/j.spasta.2022.100596

Examples

```
# Example usage:
rho <- seq(0, 1, length.out = 50)
rho_sas <- corrsas(rho, skew = 0.5, tail = 0.8, max_coeff = 20)
plot(rho, rho_sas, type = "l", main = "SAS Correlation",
         xlab = "Original Correlation", ylab = "Transformed Correlation")
```

GeoAniso — *Spatial Anisotropy correction*

Description

Transforms or back-transforms a set of coordinates according to the geometric anisotropy parameters.

Usage

```
GeoAniso(coords, anisopars=c(0,1), inverse = FALSE)
```

Arguments

`coords`	An n x 2 matrix with the coordinates to be transformed.
`anisopars`	A bivariate vector with the the anisotropy angle and the anisotropy ratio, respectively. The angle must be given in radians in [0,pi] and the anisotropy ratio must be greater or equal than 1.
`inverse`	Logical: Default to FALSE. If TRUE the reverse transformation is performed.

Details

Geometric anisotropy is defined by a linear tranformation from the anisotropic space to the isotropic space that is

$$Y = XRS$$

where X is a matrix with original coordinates (anisotropic space), and Y is a matrix with transformed coordinates (isotropic space). Here R is a rotation matrix with associated anisotropy angle parameter (in $[0, pi]$) and a S is a shrinking matrix with associated anisotropy ratio parameter (greeater or equal than one). The two parameters are specified in the anisopars argument as a bivariate numeric vector. The case $(., 1)$ corresponds to the isotropic case.

Value

Returns a matrix of transformed coordinates

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

---

`GeoCorrFct` *Spatial and Spatio-temporal correlation or covariance of (non) Gaussian random fields*

Description

The function computes the correlations of a spatial (or spatio-temporal or bivariate spatial) Gaussian or non Gaussian randomm field for a given correlation model and a set of spatial (temporal) distances.

Usage

```
GeoCorrFct(x,t=NULL,corrmodel, model="Gaussian",
distance="Eucl", param, radius=6371,n=1,
covariance=FALSE,variogram=FALSE)
```

Arguments

x	A set of spatial distances.
t	A set of (optional) temporal distances.
corrmodel	String; the name of a correlation model, for the description see GeoCovmatrix.
model	String; the type of RF. See GeoFit.
distance	String; the name of the spatial distance. The default is Eucl, the euclidean distance. 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)
n	Numeric; the number of trials in a (negative) binomial random fields. Default is 1.
covariance	Logic; if TRUE then the covariance is returned. Default is FALSE
variogram	Logic; if FALSE then the covariance/correlation is returned. Otherwise the associated semivariogram is returned

Value

Returns correlations or covariances values associated to a given parametric spatial and temporal correlation models.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian, Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

Examples

```
library(GeoModels)

################################################################
###
### Example 1. Covariance of a Gaussian random field with underlying
### Matern correlation model with nugget
###
###############################################################
# Define the spatial distances
x = seq(0,1,0.002)
# Correlation Parameters for Matern model
CorrParam("Matern")
NuisParam("Gaussian")
# Matern Parameters
param=list(sill=2,smooth=0.5,scale=0.2/3,nugget=0.2,mean=0)
cc= GeoCorrFct(x=x, corrmodel="Matern", covariance=TRUE,
  param=param,model="Gaussian")
plot(cc,ylab="Corr",lwd=2,main="Matern correlation",type="l")

################################################################
###
### Example 2. Covariance of a Gaussian random field with underlying
### Generalized Wendland-Matern correlation model
###
###############################################################
CorrParam("GenWend_Matern")
NuisParam("Gaussian")
# GenWend Matern Parameters
param=list(sill=2,smooth=1,scale=0.1,nugget=0,power2=1/4,mean=0)
cc= GeoCorrFct(x=x, corrmodel="GenWend_Matern", param=param,model="Gaussian",covariance=FALSE)
plot(cc,ylab="Cov",lwd=2,,main="GenWend covariance",type="l")

################################################################
###
### Example 3. Semivariogram of a Tukeyh random field with underlying
### Generalized Wendland correlation model
###
###############################################################
CorrParam("GenWend")
NuisParam("Tukeyh")
x = seq(0,1,0.005)
param=list(sill=1,smooth=1,scale=0.5,nugget=0,power2=5,tail=0.1,mean=0)
cc= GeoCorrFct(x=x, corrmodel="GenWend", param=param,model="Tukeyh",variogram=TRUE)
plot(cc,ylab="Corr",lwd=2,main="Tukey semivariogram",type="l")
```

```
################################################################
###
### Example 4. Semi-Variogram of a LoggGaussian random field with underlying
### Kummer correlation model
###
###############################################################
CorrParam("Kummer")
NuisParam("LogGaussian")
# GenWend Matern Parameters
param=list(smooth=1,sill=0.5,scale=0.1,nugget=0,power2=1,mean=0)
cc= GeoCorrFct(x=x, corrmodel="Kummer", param=param,model="LogGaussian",
       ,covariance=TRUE,variogram=TRUE)
plot(cc,ylab="Semivario",lwd=2,
  main="LogGaussian semivariogram",type="l")


################################################################
###
### Example 5. Covariance of Poisson random field with underlying
### Matern correlation model
###
###############################################################
CorrParam("Matern")
NuisParam("Poisson")
x = seq(0,1,0.005)
param=list(scale=0.6/3,nugget=0,smooth=0.5,mean=2)
cc= GeoCorrFct(x=x, corrmodel="Matern", param=param,model="Poisson",covariance=TRUE)
plot(cc,ylab="Cov",lwd=2,
  main="Poisson covariance",type="l")

################################################################
###
### Example 6.  Space time  semivariogram of a Gaussian random field
### with  separable Matern correlation model
###
###############################################################

## spatial and temporal distances
h<-seq(0,3,by=0.04)
times<-seq(0,3,by=0.04)

# Correlation Parameters for the space time separable Matern model
CorrParam("Matern")
NuisParam("Gaussian")
# Matern Parameters
param=list(sill=1,scale_s=0.6/3,scale_t=0.5,nugget=0,mean=0,smooth_s=1.5,smooth_t=0.5)
cc= GeoCorrFct(x=h,t=times,corrmodel="Matern_Matern", param=param,
        model="Gaussian",variogram=TRUE)
plot(cc,lwd=2,type="l")


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

```
###
### Example 7. Correlation of a bivariate Gaussian random field
### with underlying  separable bivariate  Matern correlation model
###
################################################################
# Define the spatial distances
x = seq(0,1,0.005)
 #Correlation Parameters for the bivariate sep Matern model
CorrParam("Bi_Matern")
 #Matern Parameters
param=list(sill_1=1,sill_2=1,smooth_1=0.5,smooth_2=1,smooth_12=0.75,
           scale_1=0.2/3, scale_2=0.2/3, scale_12=0.2/3,
           mean_1=0,mean_2=0,nugget_1=0,nugget_2=0,pcol=-0.2)
cc= GeoCorrFct(x=x, corrmodel="Bi_Matern", param=param,model="Gaussian")
plot(cc,ylab="corr",lwd=2,type="l")
```

GeoCorrFct_Cop *Spatial and Spatio-temporal correlation or covariance of (non) Gaussian random fields (copula models)*

Description

The function computes the correlations of a spatial or spatio-temporal or a bivariate spatial Gaussian or non Gaussian copula randomm field with a given covariance model and a set of spatial (temporal) distances.

Usage

```
GeoCorrFct_Cop(x,t=NULL,corrmodel,
model="Gaussian",copula="Gaussian",
distance="Eucl", param, radius=6371,
n=1,covariance=FALSE,variogram=FALSE)
```

Arguments

`x`	A set of spatial distances.
`t`	A set of (optional) temporal distances.
`corrmodel`	String; the name of a correlation model, for the description see the Section **Details**.
`model`	String; the type of RF. See `GeoFit`.
`copula`	String; the type of copula. The two options are Gaussian and Clayton.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. 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)

n	Numeric; the number of trials in a (negative) binomial random fields. Default is 1.
covariance	Logic; if TRUE then the covariance is returned. Default is FALSE
variogram	Logic; if FALSE then the covariance/coorelation is returned. Otherwise the associated semivariogram is returned

Value

Returns a vector of correlations or covariances values associated to a given parametric spatial and temporal correlation models.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

Examples

```
library(GeoModels)

################################################################
###
### Example 1. Correlation of a (mean reparametrized) beta random field with underlying
### Matern correlation model using Gaussian and Clayton copulas
###
###############################################################

# Define the spatial distances
x = seq(0,0.4,0.01)

# Correlation Parameters for Matern model
CorrParam("Matern")
NuisParam("Beta2")
# corr Gaussian copula
param=list(smooth=0.5,sill=1,scale=0.2/3,nugget=0,mean=0,min=0,max=1,shape=0.5)
corr1= GeoCorrFct_Cop(x=x, corrmodel="Matern", param=param,copula="Gaussian",model="Beta2")

plot(corr1,ylab="corr",main="Gauss copula correlation",lwd=2)

# corr Clayton copula
param=list(smooth=0.5,sill=1,scale=0.2/3,nugget=0,mean=0,min=0,max=1,shape=0.5,nu=2)
corr2= GeoCorrFct_Cop(x=x, corrmodel="Matern", param=param,copula="Clayton",model="Beta2")
lines(x,corr2$corr,ylim=c(0,1),lty=2)

plot(corr1,ylab="corr",main="Clayton copula correlation",lwd=2)
```

GeoCovariogram *Computes the fitted variogram model.*

Description

The procedure computes and plots estimated covariance or semivariogram models of a Gaussian or a non Gaussian spatial (temporal or bivariate spatial) random field. It 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, invisible=FALSE, ...)
```

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.
invisible	Logical;If TRUE then a statistic the (sum of the squared diffeence between the empirical semivariogram and the estimated semivariogram) is computed.
...	other optional parameters which are passed to plot functions.

Details

The function computes the fitted variogram model

Value

Produces a plot. No values are returned.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

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 semivariogram 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
I=Inf
start=list(mean=0,scale=scale,sill=sill)
lower=list(mean=-I,scale=0,sill=0)
upper=list(mean= I,scale=I,sill=I)
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,
            lower=lower,upper=upper,
            optimizer="nlminb", fixed=fixed,neighb=3)

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

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

################################################################
###
### Example 2. Plot of fitted covariance and fitted
### and empirical semivariogram from a  Bernoulli
###  RF with Genwend  correlation.
###
###############################################################
set.seed(2111)

model="Binomial";n=1
# 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 = "GenWend"
mean = 0
nugget = 0
scale = 0.2
smooth=0
power=4
param=list(mean=mean, nugget=nugget, scale=scale,smooth=0,power2=4)
# Simulation of the Gaussian RF:
data = GeoSim(coordx=coords, corrmodel=corrmodel, model=model,param=param,n=n)$data

start=list(mean=0,scale=scale)
fixed=list(nugget=nugget,power2=4,smooth=0)
```

```
# Maximum composite-likelihood fitting of the Binomial random field:
fit = GeoFit(data,coordx=coords, corrmodel=corrmodel,model=model,
             likelihood="Marginal",type='Pairwise',start=start,n=n,
             optimizer="BFGS", fixed=fixed,neighb=4)

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

# Plot of covariance and variogram functions:
GeoCovariogram(fit, show.vario=TRUE, vario=vario,pch=20,ylim=c(0,0.3))


################################################################
###
### Example 3. Plot of fitted covariance and fitted
### and empirical semivariogram from a  Weibull  RF
###   with Wend0 correlation.
###
###############################################################
set.seed(111)

model="Weibull";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
nugget = 0
scale = 0.4
power2=4


param=list(mean=mean, 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)
I=Inf
lower=list(mean=-I,scale=0,shape=0)
upper=list(mean= I,scale=I,shape=I)
fixed=list(nugget=nugget,power2=power2)

fit = GeoFit(data,coordx=coords, corrmodel=corrmodel,model=model,
             likelihood="Marginal",type='Pairwise',start=start,
             lower=lower,upper=upper,
             optimizer="nlminb", fixed=fixed,neighb=3)

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

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

################################################################
###
### Example 4. Plot of  fitted  and empirical semivariogram
### from a space time Gaussian random fields
### with double Matern 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, 10, 1)


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

fixed=list(nugget=0, mean=0, smooth_s=0.5,smooth_t=0.5)
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="Matern_Matern", maxtime=1,
             neighb=3, likelihood="Marginal", type="Pairwise",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  fitted  and empirical semivariogram
### from a bivariate Gaussian random fields
### with  Matern correlation.
###
###############################################################
set.seed(92)
# Define the spatial-coordinates of the points:
x <- runif(600, 0, 2)
y <- runif(600, 0, 2)
coords <- cbind(x,y)
```

```
# Simulation of a bivariate spatial Gaussian RF:
# with a  Bivariate Matern
set.seed(12)
param=list(mean_1=4,mean_2=2,smooth_1=0.5,smooth_2=0.5,smooth_12=0.5,
           scale_1=0.12,scale_2=0.1,scale_12=0.15,
           sill_1=1,sill_2=1,nugget_1=0,nugget_2=0,pcol=-0.5)
data <- GeoSim(coordx=coords,corrmodel="Bi_matern",
              param=param)$data



# selecting fixed and estimated parameters
fixed=list(mean_1=4,mean_2=2,nugget_1=0,nugget_2=0,
      smooth_1=0.5,smooth_2=0.5,smooth_12=0.5)
start=list(sill_1=var(data[1,]),sill_2=var(data[2,]),
           scale_1=0.1,scale_2=0.1,scale_12=0.1,
           pcol=cor(data[1,],data[2,]))


# Maximum marginal pairwise likelihood
fitcl<- GeoFit(data=data, coordx=coords, corrmodel="Bi_Matern",
                     likelihood="Marginal",type="Pairwise",
                     optimizer="BFGS" , start=start,fixed=fixed,
                     neighb=4)
print(fitcl)

# Empirical estimation of spatio-temporal covariance:
vario = GeoVariogram(data,coordx=coords,maxdist=0.4,bivariate=TRUE)
GeoCovariogram(fitcl,vario=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 GeoCovmatrix. 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

Produces a plot. No values are returned.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

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 random field with given underlying covariance model and a set of spatial location sites (and temporal instants).

Usage

```
GeoCovmatrix(estobj = NULL, coordx, coordy = NULL, coordz = NULL, coordt = NULL,
             coordx_dyn = NULL, corrmodel, distance = "Eucl", grid = FALSE,
             model = "Gaussian", n = 1, param, anisopars = NULL, radius = 1,
             sparse = FALSE, copula = NULL, X = NULL, spobj = NULL)
```

Arguments

estobj	An object of class Geofit that includes information about data, model and estimates.
coordx	A numeric $(d \times 2)$-matrix or $(d \times 3)$-matrix. 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; optional argument, the default is NULL.
coordz	A numeric vector giving 1-dimension of spatial coordinates; optional argument, the default is NULL.
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.
n	Numeric; the number of trials in a binomial random field. Default is 1.
model	String; the type of RF. See GeoFit.
param	A list of parameter values required for the covariance model.
anisopars	A list of two elements "angle" and "ratio", i.e. the anisotropy angle and the anisotropy ratio, respectively.
radius	Numeric; a value indicating the radius of the sphere when using covariance models valid for the great circle distance. Default value is 1.
sparse	Logical; if TRUE the function returns an object of class spam. This option should be used when a parametric compactly supported covariance is used. Default is FALSE.
copula	String; the type of copula. It can be "Clayton" or "Gaussian".
X	Numeric; matrix of space-time covariates.
spobj	An object of class sp or spacetime.

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 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 $d = 3$ or a point (given in lon/lat degree format) on a sphere of arbitrary radius.

A list with all implemented spatial, space-time and bivariate correlation models is given below. 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 parameters' names 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. `GenCauchy` (generalised Cauchy in Gneiting and Schlather 2004) defined as:

     $$R(h) = (1 + h^\alpha)^{-\beta/\alpha}$$

     If h is the geodesic distance then $\alpha \in (0, 1]$.
  2. `Matern` defined as:

     $$R(h) = 2^{1-\kappa}\Gamma(\kappa)^{-1}h^\kappa K_\kappa(h)$$

     If h is the geodesic distance then $\kappa \in (0, 0.5]$.
  3. `Kummer` (Kummer hypergeometric in Ma and Bhadra 2022) defined as:

     $$R(h) = \Gamma(\kappa + \alpha)U(\alpha, 1 - \kappa, 0.5h^2)/\Gamma(\kappa + \alpha)$$

     where $U(., ., .)$ is the Kummer hypergeometric function. If h is the geodesic distance then $\kappa \in (0, 0.5]$.
  4. `Kummer_Matern` It is a rescaled version of the `Kummer` model, i.e. h must be divided by $(2(1 + \alpha))^{0.5}$. When α goes to infinity it is the Matern model.
  5. `Wave` defined as:

     $$R(h) = \sin(h)/h$$

     This model is valid only for dimensions less than or equal to 3.

6. GenWend (Generalized Wendland in Bevilacqua et al. 2019) defined as:

$$R(h) = A(1-h^2)^{\beta+\kappa}F(\beta/2,(\beta+1)/2,2\beta+\kappa+1,1-h^2)1(h\in[0,1])$$

where $\mu \geq 0.5(d+1)+\kappa$ and $A = (\Gamma(\kappa)\Gamma(2\kappa+\beta+1))/(\Gamma(2\kappa)\Gamma(\beta+1-\kappa)2^{\beta+1})$ and $F(.,.,.)$ is the Gaussian hypergeometric function. The cases $\kappa = 0,1,2$ correspond to the Wend0, Wend1 and Wend2 models respectively.
7. GenWend_Matern (Generalized Wendland Matern in Bevilacqua et al. 2022). It is defined as a rescaled version of the Generalized Wendland, i.e. h must be divided by $(\Gamma(\beta+2\kappa+1)/\Gamma(\beta))^{1/(1+2\kappa)}$. When β goes to infinity it is the Matern model.
8. GenWend_Matern2 (Generalized Wendland Matern second parametrisation). It is defined as a rescaled version of the Generalized Wendland, i.e. h must be multiplied by β and the smoothness parameter is $\kappa - 0.5$. When β goes to infinity it is the Matern model.
9. Hypergeometric (Hypergeometric model in Bevilacqua et al. 2025).
10. Hypergeometric_Matern (Hypergeometric model first parametrisation).
11. Hypergeometric_Matern2 (Hypergeometric model second parametrisation).
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. F_Sphere (F family in Alegria et al. 2021) defined as:

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

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:

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

    2. Gneiting_GC defined as:

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

    where h can be either Euclidean or geodesic distance.
    3. Iacocesare defined as:

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

    4. Porcu defined as:

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

5. `Porcu1` defined as:

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

6. `Stein` defined as:

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

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

7. `Gneiting_mat_S` defined as:

$$R(h,u) = \phi(u)^{\tau_t}\mathrm{Mat}(h\phi(u)^{-\beta}, \nu_s)$$

where $\phi(u) = (1+u^{0.5\alpha_t})$, $\tau_t \geq 3.5 + \nu_s$, $\beta \in [0,1]$.

8. `Gneiting_mat_T` defined by interchanging h with u in `Gneiting_mat_S`.
9. `Gneiting_wen_S` defined as:

$$R(h,u) = \phi(u)^{\tau_t}\mathrm{GenWend}(h\phi(u)^{\beta}, \nu_s, \mu_s)$$

where $\phi(u) = (1+u^{0.5\alpha_t})$, $\tau_t \geq 2.5 + 2\nu_s$, $\beta \in [0,1]$.

10. `Gneiting_wen_T` defined by interchanging h with u in `Gneiting_wen_S`.
11. `Matern_Matern_nosep` defined as:

$$R(h,u) = \frac{\mathrm{Matern}(h;\nu_s)\,\mathrm{Matern}(u;\nu_t)}{1+\lambda\,h^2u^2/(1+\lambda)}$$

with $\nu_s, \nu_t > 0$ and $\lambda \in [0,1]$.

12. `GenWend_GenWend_nosep` defined as:

$$R(h,u) = \frac{\mathrm{GenWend}(h;\nu_s,\delta_s)\,\mathrm{GenWend}(u;\nu_t,\delta_t)}{1+\lambda\,h^2u^2/(1+\lambda)}$$

with $\nu_x \geq -0.5$, $\delta_x > (d+1)/2 + \nu_x$ for $x = s,t$, and $\lambda \in [0,1]$.

13. `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.

14. `Sinpower_st` defined as:

$$R(h,u) = (\exp(\alpha_s\cos(h)\psi(u)/a_s)(1+\alpha_s\cos(h)\psi(u)/a_s))/k$$

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

– Separable models:
Space-time separable correlation models are easily 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:
$$R(h,u) = \text{Matern}(h;\kappa_s)\text{Matern}(u;\kappa_t)$$
3. `GenWend_GenWend` defined as:
$$R(h,u) = \text{GenWend}(h;\kappa_s,\mu_s)\text{GenWend}(u;\kappa_t,\mu_t)$$
4. `Stable_Stable` defined as:
$$R(h,u) = \exp(-h^{\alpha_s})\exp(-u^{\alpha_t})$$

Note that some models are nested (e.g. `Exp_Exp` within `Matern_Matern`).

- Spatial bivariate correlation models:
  1. `Bi_Matern` (Bivariate full Matern model)
  2. `Bi_Matern_contr` (Bivariate Matern model with constraints)
  3. `Bi_Matern_sep` (Bivariate separable Matern model)
  4. `Bi_LMC` (Bivariate linear model of coregionalization)
  5. `Bi_LMC_contr` (Bivariate LMC with constraints)
  6. `Bi_Wendx` (Bivariate full Wendland model)
  7. `Bi_Wendx_contr` (Bivariate Wendland model with constraints)
  8. `Bi_Wendx_sep` (Bivariate separable Wendland model)
  9. `Bi_F_Sphere` (Bivariate full F model on the unit sphere)

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 names of the 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 σ^2 if $h = 0$ and
$$C(h) = \sigma^2(1-\tau^2)R(h/a,\ldots), \quad h > 0$$
with nugget parameter τ^2 between 0 and 1.

Similarly, if $R(h,u)$ is a spatio-temporal correlation model given in standard notation, then the covariance model is σ^2 if $h = 0$ and $u = 0$ and
$$C(h,u) = \sigma^2(1-\tau^2)R(h/a_s, u/a_t,\ldots), \quad h > 0, u > 0$$

Here '...' stands for additional parameters.

The bivariate models implemented are the following:

1. `Bi_Matern` defined as:

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

   where $\rho = \rho_{12} = \rho_{21}$ is the colocated correlation 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_GenWend` defined as:

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

   where $\rho = \rho_{12} = \rho_{21}$ is the colocated correlation parameter and $\rho_{ii} = 1$. The model `Bi_GenWend_sep` (separable GenWendland) is a special case when $a = a_{11} = a_{12} = a_{22}$ and $\mu = \mu_{11} = \mu_{12} = \mu_{22}$. The model `Bi_GenWend_contr` (constrained GenWendland) 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, are obtained with $R(h)$ equal to the exponential correlation model.

Value

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

`bivariate`	Logical: `TRUE` if the Gaussian random field 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 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 on a regular grid, otherwise `FALSE`.
`nozero`	In the case of tapered matrix the percentage of non-zero values in the covariance matrix; otherwise `NULL`.
`n`	The number of trials for binomial RFs.
`namescorr`	String: the names of the correlation parameters.
`numcoord`	Numeric: the number of spatial coordinates.

numtime	Numeric: the number of temporal coordinates.
model	The type of RF, see GeoFit.
param	Numeric: the covariance parameters.
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.bevilacqua89@gmail.com>, https://sites.google.com/view/moreno-bevilacqua/home

Víctor Morales-Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

Christian Caamaño-Carrillo, <chcaaman@ubiobio.cl>, https://www.researchgate.net/profile/Christian-Caamano

References

Alegria, A., Cuevas-Pacheco, F., Diggle, P. and Porcu, E. (2021). The F-family of covariance functions: A Matérn analogue for modeling random fields on spheres. *Spatial Statistics* 43, 100512.

Bevilacqua, M., Faouzi, T., Furrer, R. and Porcu, E. (2019). Estimation and prediction using generalized Wendland functions under fixed domain asymptotics. *Annals of Statistics* 47(2), 828–856.

Bevilacqua, M., Caamaño-Carrillo, C. and Porcu, E. (2022). Unifying compactly supported and Matérn covariance functions in spatial statistics. *Journal of Multivariate Analysis* 189, 104949.

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

Emery, X. and Alegria, A. (2022). The Gauss hypergeometric covariance kernel for modeling second-order stationary random fields in Euclidean spaces: its compact support, properties and spectral representation. *Stochastic Environmental Research and Risk Assessment* 36, 2819–2834.

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

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

Ma, P. and Bhadra, A. (2022). Beyond Matérn: on a class of interpretable confluent hypergeometric covariance functions. *Journal of the American Statistical Association*, 1–14.

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.

Gneiting, T. and Schlather, M. (2004). Stochastic models that separate fractal dimension and the Hurst effect. *SIAM Review* 46, 269–282.

See Also

GeoKrig, GeoSim, GeoFit

Examples

```
library(GeoModels)
################################################################
###
### Example 1. Estimated spatial covariance matrix associated to
### a WCL estimates using the Matern correlation model
###
###############################################################

set.seed(3)
N <- 300  # number of location sites
x <- runif(N, 0, 1)
y <- runif(N, 0, 1)
coords <- cbind(x, y)
# Set the covariance model's parameters:
corrmodel <- "Matern"
mean <- 0.5
sill <- 1
nugget <- 0
scale <- 0.2/3
smooth <- 0.5

param <- list(mean = mean, sill = sill, nugget = nugget, scale = scale, smooth = smooth)
data <- GeoSim(coordx = coords, corrmodel = corrmodel, param = param)$data
fixed <- list(nugget = nugget, smooth = smooth)
start <- list(mean = mean, scale = scale, sill = sill)

fit0 <- GeoFit(data = data, coordx = coords, corrmodel = corrmodel,
               neighb = 3, likelihood = "Conditional", optimizer = "BFGS",
               type = "Pairwise", start = start, fixed = fixed)
print(fit0)

# estimated covariance matrix using Geofit object
mm <- GeoCovmatrix(fit0)$covmatrix
# estimated covariance matrix
mm1 <- GeoCovmatrix(coordx = coords, corrmodel = corrmodel,
                    param = c(fit0$param, fit0$fixed))$covmatrix
sum(mm - mm1)

################################################################
###
### Example 2. Spatial covariance matrix associated to
### the Generalized Wendland-Matern correlation model
###
###############################################################

# Correlation Parameters for Gen Wendland model
CorrParam("GenWend_Matern")
# Gen Wendland Parameters
param <- list(sill = 1, scale = 0.04, nugget = 0, smooth = 0, power2 = 1/1.5)

matrix2 <- GeoCovmatrix(coordx = coords, corrmodel = "GenWend_Matern",
```

```
                        param = param, sparse = TRUE)

# Percentage of non-zero values
matrix2$nozero

################################################################
###
### Example 3. Spatial covariance matrix associated to
### the Kummer correlation model
###
###############################################################

# Correlation Parameters for Kummer model
CorrParam("Kummer")
param <- list(sill = 1, scale = 0.2, nugget = 0, smooth = 0.5, power2 = 1)

matrix3 <- GeoCovmatrix(coordx = coords, corrmodel = "Kummer", param = param)

matrix3$covmatrix[1:4, 1:4]

################################################################
###
### Example 4. Covariance matrix associated to
### the space-time double Matern correlation model
###
###############################################################

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

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

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

# Simulation of a spatial Gaussian random field:
matrix4 <- GeoCovmatrix(coordx = coords, coordt = times,
                        corrmodel = "Matern_Matern", param = param)

dim(matrix4$covmatrix)

################################################################
###
### Example 5. Spatial covariance matrix associated to
### a skew Gaussian RF with Matern correlation model
###
###############################################################

param <- list(sill = 1, scale = 0.3/3, nugget = 0, skew = 4, smooth = 0.5)
# Simulation of a spatial Gaussian random field:
matrix5 <- GeoCovmatrix(coordx = coords, corrmodel = "Matern", param = param,
                        model = "SkewGaussian")
```

```
# covariance matrix
matrix5$covmatrix[1:4, 1:4]

################################################################
###
### Example 6. Spatial covariance matrix associated to
### a Weibull RF with GenWend correlation model
###
###############################################################

param <- list(scale = 0.3, nugget = 0, shape = 4, mean = 0, smooth = 1, power2 = 5)
# Simulation of a spatial Gaussian random field:
matrix6 <- GeoCovmatrix(coordx = coords, corrmodel = "GenWend", param = param,
                        sparse = TRUE, model = "Weibull")

# Percentage of non-zero values
matrix6$nozero

################################################################
###
### Example 7. Spatial covariance matrix associated to
### a binomial Gaussian RF with Generalized Wendland correlation model
###
###############################################################

param <- list(mean = 0.2, scale = 0.2, nugget = 0, power2 = 4, smooth = 0)
# Simulation of a spatial Gaussian random field:
matrix7 <- GeoCovmatrix(coordx = coords, corrmodel = "GenWend", param = param,
                        n = 5, sparse = TRUE, model = "Binomial")

as.matrix(matrix7$covmatrix)[1:4, 1:4]

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

set.seed(8)
# Define the spatial coordinates of the points:
x <- runif(4, 0, 1)
y <- runif(4, 0, 1)
coords <- cbind(x, y)

# Parameters
param <- list(mean_1 = 0, mean_2 = 0, sill_1 = 1, sill_2 = 2,
              scale_1 = 0.1, scale_2 = 0.1, scale_12 = 0.1,
              smooth_1 = 0.5, smooth_2 = 0.5, smooth_12 = 0.5,
              nugget_1 = 0, nugget_2 = 0, pcol = -0.25)

# Covariance matrix
```

```
matrix8 <- GeoCovmatrix(coordx = coords, corrmodel = "Bi_matern", param = param)$covmatrix

matrix8
```

GeoCV *n-fold kriging Cross-validation*

Description

The procedure use the GeoKrig or GeoKrigloc 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, estimation=TRUE, optimizer=NULL,
     lower=NULL, upper=NULL, n.fold=0.05,local=FALSE,
    neighb=NULL, maxdist=NULL,maxtime=NULL,sparse=FALSE,
    type_krig="Simple", which=1, parallel=TRUE, ncores=NULL,progress=TRUE)
```

Arguments

fit	An object of class GeoFit.
K	The number of iterations in cross-validation.
estimation	Logical; if TRUE then an estimation is performed at each iteration and the estimates are used in the prediction. Otherwise the estimates in the object fit are used.
optimizer	The type of optimization algorithm if estimation is TRUE. See GeoFit for details. If NULL then the optimization algorithm of the object fit is chosen.
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 if estimation is TRUE.
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 if estimation is TRUE.
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 if local kriging is performed.
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 and simulation are computed with sparse matrices algorithms using spam package. Default is FALSE. It should be used with compactly supported covariances.

`type_krig` String; the type of kriging. If `Simple` (the default) then simple kriging is performed. If `Optim` then optimal kriging is performed for some non-Gaussian RFs

`which` Numeric; In the case of bivariate cokriging it indicates which variable to predict. It can be 1 or 2

`parallel` Logical; if `TRUE` then the estimation step is parallelized

`ncores` Numeric; number of cores involved in parallelization.

`progress` Logic; If TRUE then a progress bar is shown.

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;

`mad` The vector of median absolute error in the CV procedure;

`brie` The vector of brie score 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.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoKrig.

Examples

```
 library(GeoModels)

################################################################
########### Examples of spatial kriging ############
################################################################

model="Gaussian"
set.seed(79)
x = runif(400, 0, 1)
y = runif(400, 0, 1)
```

```
coords=cbind(x,y)
# Set the exponential cov parameters:
corrmodel = "GenWend"
mean=0; sill=5; nugget=0
scale=0.2;smooth=0;power2=4

param=list(mean=mean,sill=sill,nugget=nugget,scale=scale,smooth=smooth,power2=power2)

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

## estimation with pairwise likelihood
fixed=list(nugget=nugget,smooth=0,power2=power2)
start=list(mean=0,scale=scale,sill=1)
I=Inf
lower=list(mean=-I,scale=0,sill=0)
upper=list(mean= I,scale=I,sill=I)
# Maximum pairwise likelihood fitting :
fit = GeoFit(data, coordx=coords, corrmodel=corrmodel,model=model,
                    likelihood='Marginal', type='Pairwise',neighb=3,
                    optimizer="nlminb", lower=lower,upper=upper,
                    start=start,fixed=fixed)

#a=GeoCV(fit,K=100,estimation=TRUE,parallel=TRUE)
#mean(a$rmse)
```

GeoDistances *Compute Distance Matrices (Euclidean, Chordal, or Geodesic)*

Description

Compute a full distance matrix between coordinates in 2D or 3D, using Euclidean, Chordal, or Geodesic distance. The computation is done in C for efficiency, with parameters passed from R via `dotCall64`.

Usage

```
GeoDistances(coordx = NULL, coordy = NULL, coordz = NULL,
             distance = c("Eucl", "Chor", "Geod"), radius = 1)
```

Arguments

`coordx`	A numeric matrix of coordinates. Must have 2 or 3 columns (for 2D or 3D) (if coordy and coordz are not provided). Otherwise coordinates can be given via `coordx`, `coordy`, and optionally `coordz`.
`coordy`	Optional numeric vector of coordinates.
`coordz`	Optional numeric vector of coordinates (for 3D Euclidean).

`distance`	Type of distance to compute: "Eucl" for Euclidean, "Chor" for chordal, "Geod" for geodesic.
`radius`	Radius of the sphere for geodesic or chordal distances. Defaults to 1. Use Earth radius (e.g., 6371 Km) for geographic distances.

Details

- Euclidean distance: straight-line distance in 2D or 3D space.
- Chordal distance: Euclidean distance between points projected on the unit sphere, scaled by the given radius.
- Geodesic distance: shortest path along the surface of a sphere of radius `radius`.

Value

A symmetric numeric matrix of size `n x n`, where `n` is the number of points. Each entry contains the distance between the corresponding pair of coordinates.

Examples

```
# Example with Euclidean distance
coords <- cbind(c(0, 10), c(0, 10))
GeoDistances(coords, distance = "Eucl")

# Example with geodesic distance (approx distance between Rome and New York)
rome <- c(12.4964, 41.9028)     #
ny   <- c(-74.0060, 40.7128)
coords <- rbind(rome, ny)
GeoDistances(coords, distance = "Geod", radius = 6371)
```

`GeoDoScores` *Computation of drop-one predictive scores*

Description

The function computes RMSE, MAE, LSCORE, CRPS predictive scores based on drop-one prediction for a spatial, spatiotemporal and bivariate Gaussian RFs

Usage

```
GeoDoScores(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 GeoCovmatrix. 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 four predictive scores based on drop-one prediction.

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.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

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)

################################################################
######### Examples of predictive score computation  ############
################################################################
set.seed(8)
  # 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 <- GeoDoScores(data,matrix=matrix1)
```

Pr_scores

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

Description

Maximum weighted composite-likelihood fitting for Gaussian and some Non-Gaussian univariate spatial, spatio-temporal and bivariate spatial random fields. The function allows fixing any of the parameters and setting upper/lower bounds in the optimization. Different optimization methods can be used.

Usage

```
GeoFit(data, coordx, coordy=NULL, coordz=NULL, coordt=NULL,
       coordx_dyn=NULL, copula=NULL, corrmodel=NULL, distance="Eucl",
       fixed=NULL, anisopars=NULL, est.aniso=c(FALSE,FALSE),
       grid=FALSE, likelihood="Marginal", lower=NULL, maxdist=Inf,
       neighb=NULL, p_neighb=1, maxtime=Inf, memdist=TRUE,
       method="cholesky", model="Gaussian", n=1, onlyvar=FALSE,
       optimizer="Nelder-Mead", radius=1, score=FALSE,
       sensitivity=FALSE, sparse=FALSE, start=NULL,
       thin_method="iid",type="Pairwise", upper=NULL,
       varest=FALSE, weighted=FALSE,
       X=NULL, spobj=NULL, spdata=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 spatio-temporal realisation) or a $(d \times d \times t \times n)$-array (a single spatio-temporal realisation on regular grid). For the description see the Section **Details**.
coordx	A numeric $(d \times 2)$-matrix or $(d \times 3)$-matrix. 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; optional argument, default is NULL.
coordz	A numeric vector giving 1-dimension of spatial coordinates; optional argument, default is NULL.
coordt	A numeric vector assigning 1-dimension of temporal coordinates. Optional argument, default is NULL: if 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, default is NULL.
copula	String; the type of copula. It can be "Clayton" or "Gaussian".

`corrmodel`	String; the name of a correlation model, for the description see the Section **Details**.
`distance`	String; the name of the spatial distance. Default is `"Eucl"` (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 model/correlation function will not be estimated.
`anisopars`	A list of two elements: `"angle"` and `"ratio"`, i.e. the anisotropy angle and the anisotropy ratio, respectively.
`est.aniso`	A bivariate logical vector providing which anisotropy parameters must be estimated.
`grid`	Logical; if `FALSE` (default) the data are interpreted as spatial or spatio-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**.
`lower`	An optional named list giving lower bounds for parameters when the optimizer is `L-BFGS-B`, `nlminb`, `bobyqa` or `optimize`. Names must match those in `start`.
`maxdist`	Numeric; an optional positive value indicating the maximum spatial distance considered in the composite 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.
`p_neighb`	Numeric; a value in $(0, 1]$ specifying the expected fraction of nearest-neighbor pairs retained through stochastic thinning. If `1` (default), no thinning is applied and all nearest-neighbor pairs are used. If `<1`, pairs are randomly retained (Bernoulli sampling).
`maxtime`	Numeric; an optional positive integer indicating the order of temporal neighborhood in the composite likelihood computation.
`memdist`	Logical; if `TRUE` then all distances useful in the composite likelihood estimation are computed before the optimization. `FALSE` is deprecated.
`method`	String; the type of matrix decomposition/linear algebra backend used in likelihood computations. Default is `"cholesky"`. Another possible choice is `"svd"` (when available).
`model`	String; the type of random field (and associated density) used in the likelihood objects. Default is `"Gaussian"`; see the Section **Details**.
`n`	Numeric; number of trials in a binomial random field; number of successes in a negative binomial random field.
`onlyvar`	Logical; if `TRUE` (and `varest=TRUE`) only the variance-covariance matrix is computed without optimizing. Default is `FALSE`.
`optimizer`	String; the optimization algorithm (see `optim` for details). Default is `"Nelder-Mead"`. Other possible choices are `"nlm"`, `"BFGS"`, `"SANN"`, `"L-BFGS-B"`, `"nlminb"`, `"bobyqa"`. For `"L-BFGS-B"`, `"nlminb"` and `"bobyqa"` bounds can be passed via `lower` and `upper`. In the one-dimensional case, `optimize` is used.

radius	Numeric; the radius of the sphere in the case of lon-lat coordinates. Default is 1.
score	Logical; if TRUE the score function is computed. Default is FALSE.
sensitivity	Logical; if TRUE the sensitivity matrix is computed. Default is FALSE.
sparse	Logical; if TRUE then maximum likelihood / composite likelihood may exploit sparse-matrix algorithms (e.g., **spam**). Typically used with compactly supported covariance models. Default is FALSE.
start	An optional named list with initial values for parameters to be estimated. Default is NULL (see **Details**).
thin_method	String; thinning scheme used when p_neighb < 1. Default is "iid" (independent Bernoulli thinning).
type	String; the type of likelihood objects. If "Pairwise" (default) the composite likelihood is formed by pairwise components (see **Details**).
upper	An optional named list giving upper bounds for parameters when the optimizer is L-BFGS-B, nlminb, bobyqa or optimize. Names must match those in start.
varest	Logical; if TRUE the estimates' variances and standard errors are returned. For composite likelihood estimation it is deprecated. Use sensitivity=TRUE and update the object using GeoVarestbootstrap. Default is FALSE.
weighted	Logical; if TRUE the likelihood objects are weighted; see the Section **Details**. Default is FALSE.
X	Numeric; matrix of spatio(temporal) covariates in the linear mean specification.
spobj	An object of class sp or spacetime.
spdata	Character; the name of the data component in the sp or spacetime object.

Details

GeoFit provides weighted composite likelihood estimation based on pairs and independence composite likelihood estimation for Gaussian and non-Gaussian random fields. Specifically, marginal and conditional pairwise likelihoods are available for each type of random field.

The optimization method is specified using optimizer. The default method is Nelder-Mead; other available methods are nlm, BFGS, SANN, L-BFGS-B, bobyqa, and nlminb. In the last three cases, bounds can be specified using lower and upper.

Depending on the dimension of data and on the name of the correlation model, the observations are assumed to be a realization of a spatial, spatio-temporal or bivariate random field. Specifically, with data, coordx, coordy, coordt:

- If data is a numeric d-dimensional vector and coordx, coordy are two numeric d-dimensional vectors (or coordx is a $(d \times 2)$-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 and coordt is a numeric t-dimensional vector, then the data are interpreted as a single spatio-temporal realisation observed on d sites and t times;
- If data is a numeric $(2 \times d)$-matrix, then the data are interpreted as a single bivariate spatial realisation observed on d spatial sites;
- If data is a list, coordx_dyn is a list and coordt is a numeric t-dimensional vector, then the data are interpreted as a spatio-temporal realisation observed on dynamical spatial sites (different locations for each time) and for t times.

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

- In the spatial case, X must be a $(d \times k)$ matrix associated to data (a d-vector);
- In the spatio-temporal case, X must be a $(N \times k)$ matrix associated to data (a $t \times d$-matrix), where $N = t \times d$;
- In the bivariate case, X must be a $(N \times k)$ matrix associated to data (a $2 \times d$-matrix), where $N = 2 \times d$.

The distance parameter allows different kinds of spatial distances:

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

The likelihood parameter represents the composite-likelihood configuration:

1. Conditional, composite likelihood formed by conditionals;
2. Marginal, composite likelihood formed by marginals (default);
3. Full, standard likelihood.

It must be coupled with type:

1. Pairwise, composite likelihood based on pairs;
2. Independence, composite likelihood based on independence;
3. Standard, standard likelihood.

Stochastic thinning of nearest-neighbor pairs can be enabled via p_neighb<1. The argument thin_method controls the thinning scheme (default "iid").

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 random field is bivariate, otherwise FALSE.
clic	The composite information criterion; if the full likelihood is considered then it coincides with AIC.
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.
conf.int	Confidence intervals for standard maximum likelihood estimation.
convergence	A string that denotes if convergence is reached.
copula	The type of copula.
corrmodel	The correlation model.
data	The vector/matrix/array (or list) of data.

distance	The type of spatial distance.
fixed	A list of fixed parameters.
iterations	The number of iterations 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 weighted composite likelihood (or NULL).
maxtime	The order of temporal neighborhood in the composite likelihood computation.
message	Extra message passed from the numerical routines.
model	The density associated to the likelihood objects.
missp	TRUE if a misspecified Gaussian model is used in the composite likelihood.
n	The number of trials/successes for (negative) binomial models.
neighb	The order of spatial neighborhood in the composite likelihood computation.
ns	The number of (different) location sites in the bivariate case.
numcoord	The number of spatial coordinates.
numtime	The number of temporal realisations.
param	A list of parameter estimates.
radius	The radius of the sphere in the case of great-circle distance.
stderr	Standard errors for standard maximum likelihood estimation.
sensmat	The sensitivity matrix.
varcov	The variance-covariance matrix of the estimates.
type	The type of likelihood objects.
X	The matrix of covariates.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

References

General 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.

Non Gaussian random fields:

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*.

Blasi F., Caamaño C., Bevilacqua M., Furrer R. (2022) A selective view of climatological data and likelihood estimation *Spatial Statistics* 10.1016/j.spasta.2022.100596

Bevilacqua M., Caamaño C., Arellano-Valle R. B., Camilo Gomez C. (2022) A class of random fields with two-piece marginal distributions for modeling point-referenced data with spatial outliers. *Test* 10.1007/s11749-021-00797-5

Morales-Navarrete D., Bevilacqua M., Caamaño C., Castro L.M. (2022) Modelling Point Referenced Spatial Count Data: A Poisson Process Approach *TJournal of the American Statistical Association* To appear

Caamaño C., Bevilacqua M., López C., Morales-Oñate V. (2023) Nearest neighbours weighted composite likelihood based on pairs for (non-)Gaussian massive spatial data with an application to Tukey-hh random fields estimation *Computational Statistics and Data Analysis* To appear

Bevilacqua M., Alvarado E., Caamaño C., (2023) A flexible Clayton-like spatial copula with application to bounded support data *Journal of Multivariate Analysis* To appear

Weighted Composite-likelihood for (non-)Gaussian random fields:

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.

Caamaño C., Bevilacqua M., López C., Morales-Oñate V. (2023) Nearest neighbours weighted composite likelihood based on pairs for (non-)Gaussian massive spatial data with an application to Tukey-hh random fields estimation *Computational Statistics and Data Analysis* To appear

Examples

```
library(GeoModels)

################################################################
############ Examples of spatial Gaussian random fieldss ################
################################################################


# Define the spatial-coordinates of the points:
set.seed(3)
N=300  # number of location sites
x <- runif(N, 0, 1)
y <- runif(N, 0, 1)
coords <- cbind(x,y)
```

```
# Define spatial matrix covariates and regression parameters
X=cbind(rep(1,N),runif(N))
mean <- 0.2
mean1 <- -0.5

# Set the covariance model's parameters:
corrmodel <- "Matern"
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 random fields:
data <- GeoSim(coordx=coords,corrmodel=corrmodel, param=param,X=X)$data



################################################################
###
### Example 0. Maximum independence composite likelihood fitting of
### a Gaussian random fields (no dependence parameters)
###
###############################################################
# setting starting parameters to be estimated
start<-list(mean=mean,mean1=mean1,sill=sill)

fit1 <- GeoFit(data=data,coordx=coords,likelihood="Marginal",
                    type="Independence", start=start,X=X)
print(fit1)


################################################################
###
### Example 1. Maximum conditional pairwise likelihood fitting of
### a Gaussian random fields using Nelder-Mead
###
###############################################################
# setting fixed and starting parameters to be estimated
fixed<-list(nugget=nugget,smooth=smooth)
start<-list(mean=mean,mean1=mean1,scale=scale,sill=sill)

fit1 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
                    neighb=3,likelihood="Conditional",optimizer="Nelder-Mead",
                    type="Pairwise", start=start,fixed=fixed,X=X)
print(fit1)

################################################################
###
### Example 2. Standard Maximum likelihood fitting of
### a Gaussian random fields using nlminb
```

```
###
###############################################################
# Define the spatial-coordinates of the points:
set.seed(3)
N=250  # number of location sites
x <- runif(N, 0, 1)
y <- runif(N, 0, 1)
coords <- cbind(x,y)

param<-list(mean=mean,sill=sill,nugget=nugget,scale=scale,smooth=smooth)

data <- GeoSim(coordx=coords,corrmodel=corrmodel, param=param)$data

# setting fixed and parameters to be estimated
fixed<-list(nugget=nugget,smooth=smooth)
start<-list(mean=mean,scale=scale,sill=sill)

I=Inf
lower<-list(mean=-I,scale=0,sill=0)
upper<-list(mean=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)
print(fit2)


###############################################################
############ Examples of spatial non-Gaussian random fieldss #############
###############################################################


################################################################
###
### Example 3. Maximum pairwise likelihood fitting of a Weibull  random fields
### with Generalized Wendland correlation with Nelder-Mead
###
###############################################################
set.seed(524)
# 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
shape=2
scale=0.2
smooth=0

model="Weibull"
corrmodel="GenWend"
```

```
param=list(mean=mean,mean1=mean1,scale=scale,
                     shape=shape,nugget=nugget,power2=4,smooth=smooth)
# Simulation of a  non stationary weibull random fields:
data <- GeoSim(coordx=coords, corrmodel=corrmodel,model=model,X=X,
           param=param)$data


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

# Maximum independence likelihood:
fit <- GeoFit(data=data, coordx=coords, X=X,
           likelihood="Marginal",type="Independence", corrmodel=corrmodel,
         ,model=model, start=start, fixed=fixed)
print(unlist(fit$param))

## estimating  dependence parameter fixing vector mean   parameter
Xb=as.numeric(X %*% c(mean,mean1))
fixed<-list(nugget=nugget,power2=4,smooth=smooth,mean=Xb)
start<-list(scale=scale,shape=shape)

# Maximum conditional composite-likelihood fitting of the random fields:
fit1 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                    neighb=3,likelihood="Conditional",type="Pairwise",
                    optimizer="Nelder-Mead",
                    start=start,fixed=fixed)
print(unlist(fit1$param))



### joint estimation  of the dependence parameter and  mean   parameters
fixed<-list(nugget=nugget,power2=4,smooth=smooth)
start<-list(mean=mean,mean1=mean1,scale=scale,shape=shape)
fit2 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                    neighb=3,likelihood="Conditional",type="Pairwise",X=X,
                    optimizer="Nelder-Mead",
                    start=start,fixed=fixed)
print(unlist(fit2$param))



################################################################
###
### Example 4. Maximum pairwise likelihood fitting of
### a Skew-Gaussian spatial  random fields with Wendland correlation
###
###############################################################
set.seed(261)
model="SkewGaussian"
# 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=-4.5
power2=4
c_supp=0.2

# model parameters
param=list(power2=power2,skew=skew,
             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,mean=mean,sill=sill)
lower=list(scale=0,skew=-I,mean=-I,sill=0)
upper=list(scale=I,skew=I,mean=I,sill=I)
# Maximum marginal pairwise likelihood:
fit1 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                    neighb=3,likelihood="Marginal",type="Pairwise",
                    optimizer="bobyqa",lower=lower,upper=upper,
                    start=start,fixed=fixed)
print(unlist(fit1$param))


#################################################################
###
### Example 5. Maximum pairwise likelihood fitting of
### a Bernoulli random fields with exponential correlation
###
################################################################

set.seed(422)
N=250
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,nugget=0,scale=0.2,power2=4)
# Simulation of the spatial Binomial-Gaussian random fields:
data <- GeoSim(coordx=coords, corrmodel=corrmodel, model="Binomial", n=1,X=X,
             param=param)$data


## estimating the marginal parameters using independence cl
fixed <- list(power2=4,scale=0.2,nugget=0)
start <- list(mean=mean,mean1=mean1,mean2=mean2)

# Maximum independence likelihood:
fit <- GeoFit(data=data, coordx=coords,n=1, X=X,
```

```
           likelihood="Marginal",type="Independence",corrmodel=corrmodel,
         ,model="Binomial", start=start, fixed=fixed)

print(fit)


## estimating  dependence parameter fixing vector mean   parameter
Xb=as.numeric(X %*% c(mean,mean1,mean2))
fixed <- list(nugget=0,power2=4,mean=Xb)
start <- list(scale=0.2)

# Maximum conditional pairwise likelihood:
fit1 <- GeoFit(data=data, coordx=coords, corrmodel=corrmodel,n=1,
          likelihood="Conditional",type="Pairwise",  neighb=3
         ,model="Binomial", start=start, fixed=fixed)

print(fit1)


## estimating jointly marginal   and dependnce parameters
fixed <- list(nugget=0,power2=4)
start <- list(mean=mean,mean1=mean1,mean2=mean2,scale=0.2)

# Maximum conditional pairwise likelihood:
fit2 <- GeoFit(data=data, coordx=coords, corrmodel=corrmodel,n=1, X=X,
          likelihood="Conditional",type="Pairwise",  neighb=3
         ,model="Binomial", start=start, fixed=fixed)

print(fit2)


################################################################
######### Examples of Gaussian spatio-temporal random fieldss ###########
################################################################
set.seed(52)
# Define the temporal sequence:
time <- seq(1, 9, 1)

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

# Set the covariance model's parameters:
scale_s=0.2/3;scale_t=1
smooth_s=0.5;smooth_t=0.5
sill=1
nugget=0
mean=0

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

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

################################################################
###
### Example 6. Maximum pairwise likelihood fitting of a
### space time Gaussian random fields with double-exponential correlation
###
###############################################################
# Fixed parameters
fixed<-list(nugget=nugget,smooth_s=smooth_s,smooth_t=smooth_t)
# 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 random fields:
fit <- GeoFit(data=data,coordx=coords,coordt=time,
                    corrmodel="Matern_Matern",maxtime=1,neighb=3,
                    likelihood="Marginal",type="Pairwise",
                     start=start,fixed=fixed)
print(fit)



###############################################################
######### Examples of a bivariate Gaussian  random fields ###########
###############################################################

################################################################
### Example 7. Maximum pairwise  likelihood fitting of a
### bivariate Gaussian random fields with separable Bivariate  matern
### (cross) correlation model
###############################################################

# Define the spatial-coordinates of the points:
set.seed(89)
x <- runif(300, 0, 1)
y <- runif(300, 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 bivariate Gaussian random fields:
data <- GeoSim(coordx=coords, corrmodel="Bi_Matern_sep",
              param=param)$data

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

```
# Maximum marginal pairwise likelihood
fitcl<- GeoFit(data=data, coordx=coords, corrmodel="Bi_Matern_sep",
                     likelihood="Marginal",type="Pairwise",
                      start=start,fixed=fixed,
                     neighb=3)
print(fitcl)
```

GeoFit2 *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. A first preliminary estimation is performed using independence composite-likelihood for the marginal parameters of the model. The estimates are then used as starting values in the second final estimation step. The function allows fixing any of the parameters and setting upper/lower bounds in the optimization.

Usage

```
GeoFit2(data, coordx, coordy=NULL, coordz=NULL, coordt=NULL,
        coordx_dyn=NULL, copula=NULL, corrmodel, distance="Eucl",
        fixed=NULL, anisopars=NULL, est.aniso=c(FALSE,FALSE),
        grid=FALSE, likelihood="Marginal", lower=NULL, maxdist=Inf,
        neighb=NULL, p_neighb=1, maxtime=Inf, memdist=TRUE,
        method="cholesky", model="Gaussian", n=1, onlyvar=FALSE,
        optimizer="Nelder-Mead", radius=1, score=FALSE,
        sensitivity=FALSE, sparse=FALSE, start=NULL,
        thin_method="iid", type="Pairwise", upper=NULL,
        varest=FALSE, weighted=FALSE, X=NULL,
        spobj=NULL, spdata=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 or $(d \times 3)$-matrix. 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; optional argument, default is `NULL`.
`coordz`	A numeric vector giving 1-dimension of spatial coordinates; optional argument, default is `NULL`.
`coordt`	A numeric vector assigning 1-dimension of temporal coordinates. Optional argument, default is `NULL`: if `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, default is NULL.

copula — String; the type of copula. It can be "Clayton" or "Gaussian".

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

distance — String; the name of the spatial distance. Default is "Eucl" (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 not be estimated.

anisopars — A list of two elements: "angle" and "ratio", i.e. the anisotropy angle and the anisotropy ratio, respectively.

est.aniso — A bivariate logical vector providing which anisotropic parameters must be estimated.

grid — Logical; if FALSE (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**.

lower — An optional named list giving lower bounds for parameters when the optimizer is L-BFGS-B, nlminb, bobyqa or optimize. Names must match those in start.

maxdist — Numeric; an optional positive value indicating the maximum spatial distance considered in the composite 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.

p_neighb — Numeric; a value in $(0, 1]$ specifying the expected fraction of nearest-neighbor pairs retained through stochastic thinning. If 1 (default), no thinning is applied and all nearest-neighbor pairs are used. If <1, pairs are randomly retained using independent Bernoulli sampling.

maxtime — Numeric; an optional positive integer indicating the order of temporal neighborhood in the composite likelihood computation.

memdist — Logical; if TRUE then all 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 likelihood computation. Default is "cholesky". Another possible choice is "svd" (when available).

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=TRUE) only the variance-covariance matrix is computed without optimizing. Default is FALSE.

optimizer
: String; the optimization algorithm (see optim for details). "Nelder-Mead" is the default. Other possible choices are "nlm", "BFGS", "SANN", "L-BFGS-B", "nlminb", "bobyqa". In these last three cases upper and lower bounds can be passed by the user. In the one-dimensional case, optimize is used.

radius
: Numeric; the radius of the sphere in the case of lon-lat coordinates. Default value is 1.

score
: Logical; should score function be computed? Default is FALSE.

sensitivity
: Logical; if TRUE then the sensitivity matrix is computed.

sparse
: Logical; if TRUE then maximum likelihood is computed using sparse matrix algorithms (e.g., **spam**). It should be used with compactly supported covariance models. Default is FALSE.

start
: An optional named list with initial values for parameters used by the numerical routines in the maximization procedure. Default is NULL (see **Details**).

thin_method
: String; thinning scheme used when p_neighb<1. Default is "iid" (independent Bernoulli thinning).

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

upper
: An optional named list giving upper bounds for parameters when the optimizer is L-BFGS-B, nlminb, bobyqa or optimize. Names must match those in start.

varest
: Logical; if TRUE the estimates' variances and standard errors are returned. For composite likelihood estimation it is deprecated. Use sensitivity=TRUE and update the object using GeoVarestbootstrap. Default is FALSE.

weighted
: Logical; if TRUE the likelihood objects are weighted; see the Section **Details**. Default is FALSE.

X
: Numeric; matrix of spatio(temporal) covariates in the linear mean specification.

spobj
: An object of class sp or spacetime.

spdata
: Character; the name of data in the sp or spacetime object.

Details

The function GeoFit2 is similar to GeoFit. However, GeoFit2 performs a preliminary estimation using maximum independence composite likelihood for the marginal parameters of the model and then uses the obtained estimates as starting values in the final weighted composite likelihood estimation (that includes both marginal and dependence parameters). This provides robust starting values for the marginal parameters in the optimization algorithm.

Stochastic thinning of nearest-neighbor pairs can be enabled via p_neighb<1. The argument thin_method controls the thinning scheme (default "iid").

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 AIC.
`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.
`conf.int`	Confidence intervals for standard maximum likelihood estimation.
`convergence`	A string that denotes if convergence is reached.
`copula`	The type of copula.
`corrmodel`	The correlation model.
`data`	The vector or matrix or array (or list) of data.
`distance`	The type of spatial distance.
`fixed`	A list of fixed parameters.
`iterations`	The number of iterations 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 weighted composite likelihood (or `NULL`).
`maxtime`	The order of temporal neighborhood in the composite likelihood computation.
`message`	Extra message passed from the numerical routines.
`model`	The density associated to the likelihood objects.
`missp`	`TRUE` if a misspecified Gaussian model is used in the composite likelihood.
`n`	The number of trials in a binomial RF; the number of successes in a negative binomial RF.
`neighb`	The order of spatial neighborhood in the composite likelihood computation.
`ns`	The number of (different) location sites in the bivariate case.
`numcoord`	The number of spatial coordinates.
`numtime`	The number of temporal realisations of the RF.
`param`	A list of parameter estimates.
`radius`	The radius of the sphere in the case of great-circle distance.
`stderr`	Standard errors for standard maximum likelihood estimation.
`sensmat`	The sensitivity matrix.
`varcov`	The variance-covariance matrix of the estimates.
`type`	The type of the likelihood objects.
`X`	The matrix of covariates.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>, https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian Caamaño-Carrillo, <chcaaman@ubiobio.cl>, https://www.researchgate.net/profile/Christian-Caamano

Examples

```
library(GeoModels)


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

#################################################################
###
### Example 1 : Maximum pairwise conditional likelihood fitting
###  of a Gaussian RF with Matern correlation
###
################################################################
model="Gaussian"
# 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,model=model,corrmodel=corrmodel, param=param,X=X)$data

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

#################################################################
###
### Maximum pairwise likelihood fitting of
### Gaussian RFs with exponential correlation.
###
################################################################
fit1 <- GeoFit2(data=data,coordx=coords,corrmodel=corrmodel,
                    optimizer="BFGS",neighb=3,likelihood="Conditional",
                    type="Pairwise", start=start,fixed=fixed,X=X)
```

```
print(fit1)



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


################################################################
###
### Example 2. Maximum pairwise likelihood fitting of
### a LogGaussian  RF with Generalized Wendland correlation
###
###############################################################
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
sill=0.5
scale=0.2
smooth=0

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

fixed<-list(nugget=nugget,power2=4,smooth=smooth)
start<-list(mean=mean,mean1=mean1,scale=scale,sill=sill)
I=Inf
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 <- GeoFit2(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                    neighb=3,likelihood="Conditional",type="Pairwise",X=X,
                    optimizer="nlminb",lower=lower,upper=upper,
                    start=start,fixed=fixed)
print(unlist(fit$param))

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

```
### Example 3. Maximum pairwise likelihood fitting of
### SinhAsinh   RFs with Wendland0 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 <- GeoFit2(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                    neighb=3,likelihood="Marginal",type="Pairwise",
                    start=start,fixed=fixed)
print(unlist(fit1$param))
```

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

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(estobj=NULL,data, coordx, coordy=NULL, coordz=NULL, coordt=NULL,
coordx_dyn=NULL, corrmodel,distance="Eucl",
    grid=FALSE, loc,
```

```
method="cholesky", model="Gaussian", n=1,nloc=NULL,mse=FALSE,
 param, anisopars=NULL,radius=1, sparse=FALSE,
 time=NULL,  type_krig="Simple",weigthed=TRUE,which=1,
copula=NULL, X=NULL,Xloc=NULL,Mloc=NULL,spobj=NULL,spdata=NULL,varcov=NULL)
```

Arguments

- `estobj` An object of class Geofit that includes information about data, model and estimates.
- `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 or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.
- `coordz` A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is `NULL`.
- `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.
- `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. 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**.
- `anisopars` A list of two elements: "angle" and "ratio" i.e. the anisotropy angle and the anisotropy ratio, respectively.

radius	Numeric: the radius of the sphere if coordinates are passed in lon/lat format;Default value is 1.
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_krig	String; the type of kriging. If Simple (the default) then simple kriging is performed. Otherwise Universal.
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 cokriging it indicates which variable to predict. It can be 1 or 2
copula	String; the type of copula. It can be "Clayton" or "Gaussian"
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.
Mloc	Numeric; Vector of spatio(temporal) estimated means associated to predicted locations.
spobj	An object of class sp or spacetime
spdata	Character:The name of data in the sp or spacetime object
varcov	Inverse of the covariance matrix estimates.

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).

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;

coordz	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.
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 sparse 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;
time	A m-dimensional vector of temporal coordinates to be predicted;
type	String: the type of kriging (Standard).
type_krig	String: the type of kriging (simple or universal)
mse	A $(m \times n)$-matrix of spatio or spatio temporal mean square error kriging prediction;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

References

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

See Also

GeoCovmatrix

Examples

```
library(GeoModels)
################################################################
########### Examples of spatial kriging ############
################################################################

################################################################
###
### Example 1. Spatial  kriging of a
### Gaussian random fields with Gen wendland correlation.
###
################################################################

model="Gaussian"
set.seed(79)
x = runif(300, 0, 1)
y = runif(300, 0, 1)
coords=cbind(x,y)
# Set the exponential cov parameters:
corrmodel = "GenWend"
mean=0; sill=5; nugget=0
scale=0.2;smooth=0;power2=4

param=list(mean=mean,sill=sill,nugget=nugget,scale=scale,smooth=smooth,power2=power2)

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

## estimation with pairwise likelihood
fixed=list(nugget=nugget,smooth=0,power2=power2)
start=list(mean=0,scale=scale,sill=1)
I=Inf
lower=list(mean=-I,scale=0,sill=0)
upper=list(mean= I,scale=I,sill=I)
# Maximum pairwise likelihood fitting :
fit = GeoFit(data, coordx=coords, corrmodel=corrmodel,model=model,
             likelihood='Marginal', type='Pairwise',neighb=3,
             optimizer="nlminb", lower=lower,upper=upper,
             start=start,fixed=fixed)

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

## first option
#param=append(fit$param,fit$fixed)
#pr=GeoKrig(loc=loc_to_pred,coordx=coords,corrmodel=corrmodel,
#       model=model,param=param,data=data,mse=TRUE)

## second option using object GeoFit
pr=GeoKrig(fit,loc=loc_to_pred,mse=TRUE)
```

```
colour = rainbow(100)

opar=par(no.readonly = TRUE)
par(mfrow=c(1,3))
quilt.plot(coords,data,col=colour)
# simple kriging map prediction
image.plot(xx, xx, matrix(pr$pred,ncol=length(xx)),col=colour,
           xlab="",ylab="",
           main=" Kriging ")

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

################################################################
###
### Example 2. Spatial  kriging of a Skew
### Gaussian random fields with Matern correlation.
###
################################################################
model="SkewGaussian"
set.seed(79)
x = runif(300, 0, 1)
y = runif(300, 0, 1)
coords=cbind(x,y)
# Set the exponential cov parameters:
corrmodel = "Matern"
mean=0
sill=2
nugget=0
scale=0.1
smooth=0.5
skew=3
param=list(mean=mean,sill=sill,nugget=nugget,scale=scale,smooth=smooth,skew=skew)

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

fixed=list(nugget=nugget,smooth=smooth)
start=list(mean=0,scale=scale,sill=1,skew=skew)
I=Inf
lower=list(mean=-I,scale=0,sill=0,skew=-I)
upper=list(mean= I,scale=I,sill=I,skew=I)
# Maximum pairwise likelihood fitting :
fit = GeoFit2(data, coordx=coords, corrmodel=corrmodel,model=model,
              likelihood='Marginal', type='Pairwise',neighb=3,
              optimizer="nlminb", lower=lower,upper=upper,
              start=start,fixed=fixed)
```

```
# locations to predict
xx=seq(0,1,0.03)
loc_to_pred=as.matrix(expand.grid(xx,xx))
## optimal linear kriging
pr=GeoKrig(fit,loc=loc_to_pred,mse=TRUE)

colour = rainbow(100)

opar=par(no.readonly = TRUE)
par(mfrow=c(1,3))
quilt.plot(coords,data,col=colour)
# simple kriging map prediction
image.plot(xx, xx, matrix(pr$pred,ncol=length(xx)),col=colour,
           xlab="",ylab="",
           main=" Kriging ")

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

################################################################
###
### Example 3. Spatial  kriging  of a
###   Gamma random field with mean spatial regression
###
###############################################################
set.seed(312)
model="Gamma"
corrmodel = "GenWend"
# Define the spatial-coordinates of the points:
NN=300
coords=cbind(runif(NN),runif(NN))
## matrix covariates
a0=rep(1,NN)
a1=runif(NN,0,1)
X=cbind(a0,a1)
##Set model parameters
shape=2
## regression parameters
mean = 1;mean1= -0.2
# correlation parameters
nugget = 0;power2=4
scale = 0.3;smooth=0

## simulation
param=list(shape=shape,nugget=nugget,mean=mean,mean1=mean1,
           scale=scale,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)
```

```
start=list(mean=mean,mean1=mean1, scale=scale,shape=shape)

## estimation with pairwise likelihood
fit2 = GeoFit(data=data,coordx=coords,corrmodel=corrmodel,X=X,
              neighb=3,likelihood="Conditional",type="Pairwise",
              start=start,fixed=fixed, model = model)

# locations to predict with associated covariates
xx=seq(0,1,0.03)
loc_to_pred=as.matrix(expand.grid(xx,xx))
NP=nrow(loc_to_pred)
a0=rep(1,NP)
a1=runif(NP,0,1)
Xloc=cbind(a0,a1)

#optimal linear  kriging
pr=GeoKrig(fit2,loc=loc_to_pred,Xloc=Xloc,sparse=TRUE,mse=TRUE)

## map
opar=par(no.readonly = TRUE)
par(mfrow=c(1,3))
quilt.plot(coords,data,main="Data")
map=matrix(pr$pred,ncol=length(xx))
mapmse=matrix(pr$mse,ncol=length(xx))
image.plot(xx, xx, map,
           xlab="",ylab="",main="Kriging ")

image.plot(xx, xx, mapmse,
           xlab="",ylab="",main="MSE")
par(opar)


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

################################################################
###
### Example 4. Spatio temporal simple kriging of n locations
### sites and m temporal instants for a Gaussian random fields
### with estimated double Wendland correlation.
###
###############################################################
model="Gaussian"
# Define the spatial-coordinates of the points:
x = runif(300, 0, 1)
y = runif(300, 0, 1)
coords=cbind(x,y)
times=1:4

# Define model correlation modek and associated parameters
corrmodel="Wend0_Wend0"
param=list(nugget=0,mean=0,power2_s=4,power2_t=4,
```

```
           scale_s=0.2,scale_t=2,sill=1)

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

# Maximum pairwise likelihood fitting of the space time random field:
start = list(scale_s=0.15,scale_t=2,sill=1,mean=0)
fixed = list(nugget=0,power2_s=4,power2_t=4)

fit = GeoFit(data, coordx=coords, coordt=times, model=model, corrmodel=corrmodel,
             likelihood='Conditional', type='Pairwise',start=start,fixed=fixed,
             neighb=3,maxtime=1)

# locations to predict
xx=seq(0,1,0.04)
loc_to_pred=as.matrix(expand.grid(xx,xx))
#  Define the times to predict
times_to_pred=2

pr=GeoKrig(fit,loc=loc_to_pred,time=times_to_pred,sparse=TRUE,mse=TRUE)

opar=par(no.readonly = TRUE)
par(mfrow=c(1,3))
zlim=c(-2.5,2.5)
colour = rainbow(100)


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


par(opar)


################################################################
###
### Example r. Spatial bivariate  simple cokriging of n locations
### sites  for a bivariate Gaussian random fields
### with estimated Matern correlation.
###
###############################################################
#set.seed(6)
#NN=1500 # number of spatial locations
#x = runif(NN, 0, 1);
#y = runif(NN, 0, 1)
#coords=cbind(x,y)

## setting parameters
```

```
#mean_1 = 2; mean_2= -1
#nugget_1 =0;nugget_2=0
#sill_1 =0.5; sill_2 =1;

### correlation parameters
#CorrParam("Bi_Matern")
#scale_1=0.2/3; scale_2=0.15/3; scale_12=0.5*(scale_2+scale_1)
#smooth_1=smooth_2=smooth_12=0.5
#pcol = -0.4
#param= list(nugget_1=nugget_1,nugget_2=nugget_2,
#          sill_1=sill_1,sill_2=sill_2,
#          mean_1=mean_1,mean_2=mean_2,
#          smooth_1=smooth_1, smooth_2=smooth_2,smooth_12=smooth_12,
#          scale_1=scale_1, scale_2=scale_2,scale_12=scale_12,
#          pcol=pcol)

## simulation
#data = GeoSim(coordx=coords, corrmodel="Bi_Matern",model=model,param=param)$data

#fixed=list(mean_1=mean_1,mean_2=mean_2, nugget_1=nugget_1,nugget_2=nugget_2,
#       smooth_1=smooth_1, smooth_2=smooth_2,smooth_12=smooth_12)

#start=list( sill_1=sill_1,sill_2=sill_2,
#        scale_1=scale_1,scale_2=scale_2,scale_12=scale_12, pcol=pcol)

## estimation with maximum likelihood
#fit = GeoFit(data=data,coordx=coords, corrmodel="Bi_Matern",
  #likelihood="Marginal",type="Pairwise",optimizer="BFGS",neighb=5,
  #start=start,fixed=fixed)

###### co-kriging for the fist component ##############
#xx=seq(0,1,0.022)
#loc_to_pred=as.matrix(expand.grid(xx,xx))
#pr1 = GeoKrig(fit,which=1,mse=TRUE,loc=loc_to_pred)

#opar=par(no.readonly = TRUE)
#par(mfrow=c(1,2))
#zlim=c(-2.5,2.5)
#colour = rainbow(100)
#quilt.plot(coords,data[1,] ,col=colour,main = paste(" Fist component"))
#quilt.plot(loc_to_pred,pr1$pred,col=colour,
#           main = paste(" Kriging first component"),ylab="")
#par(opar)
```

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 optmal local linear prediction and the associated mean squared error for the Gaussian and non Gaussian case using a spatial (temporal) neighborhood computed using the function GeoNeighborhood

Usage

```
GeoKrigloc(estobj=NULL,data, coordx, coordy=NULL, coordz=NULL,coordt=NULL,
    coordx_dyn=NULL, corrmodel, distance="Eucl", grid=FALSE,
    loc, neighb=NULL, maxdist=NULL,
    maxtime=NULL, method="cholesky",
    model="Gaussian", n=1,nloc=NULL, mse=FALSE,
    param, anisopars=NULL,radius=1,
    sparse=FALSE, time=NULL, type="Standard",
    type_krig="Simple",weigthed=TRUE,
    which=1, copula=NULL,X=NULL,Xloc=NULL,Mloc=NULL,varcov=NULL,
    spobj=NULL,spdata=NULL,parallel=FALSE,ncores=NULL,progress=TRUE)
```

Arguments

estobj	An object of class Geofit that includes information about data, model and estimates.
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 or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.
coordz	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is `NULL`.
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 integer value indicating the order of 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 the binomial random field. If missing then a rounded mean of n is considered.
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**.
anisopars	A list of two elements: "angle" and "ratio" i.e. the anisotropy angle and the anisotropy ratio, respectively.
radius	Numeric: the radius of the sphere if coordinates are passed in lon/lat format;Default value is 1.
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_krig	String; the type of kriging. If Simple (the default) then simple kriging is performed. Otherwise Universal
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
copula	String; the type of copula. It can be "Clayton" or "Gaussian"
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.
Mloc	Numeric; Vector of spatio(temporal) estimated means associated to predicted locations.
varcov	Inverse of the covariance matrix estimates.

spobj	An object of class sp or spacetime
spdata	Character:The name of data in the sp or spacetime object
parallel	Logical; if TRUE then the prediction computation is parallelized
ncores	Numeric; number of cores involved in parallelization.
progress	If TRUE then a progress bar is shown.

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;
coordz	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: Simple or Universal
mse	A $(m \times n)$-matrix of spatio or spatio temporal mean square error kriging prediction;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

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)
####
model="Gaussian"

# Define the spatial-coordinates of the points:
set.seed(759)
x = runif(1000, 0, 1)
y = runif(1000, 0, 1)
coords=cbind(x,y)
# Set the exponential cov parameters:
corrmodel = "GenWend"
mean=0; sill=1
nugget=0; scale=0.2
param=list(mean=mean,sill=sill,nugget=nugget,smooth=0,
scale=scale,power2=4)

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

# Maximum pairwise likelihood fitting of the space time random field:
```

```
start=list(scale=scale,sill=sill,mean=mean)
fixed=list(power2=4,smooth=0,nugget=0)
fit = GeoFit(data, coordx=coords, corrmodel=corrmodel,
              start=start,fixed=fixed,
             likelihood='Conditional', type='Pairwise',
              neighb=3)

# locations to predict
loc_to_pred=matrix(runif(8),4,2)
#################################################################
###
### Example 1. Comparing spatial kriging with local kriging for
### a Gaussian random field with GenWend correlation.
###
################################################################
param=append(fit$param,fit$fixed)
pr=GeoKrig(fit,loc=loc_to_pred,mse=TRUE)

pr_loc=GeoKrigloc(fit,loc=loc_to_pred,neighb=100,mse=TRUE)

pr$pred;
pr_loc$pred


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


require(GeoModels)
require(fields)
set.seed(78)
coords=cbind(runif(100),runif(100))
coordt=seq(0,5,0.25)
corrmodel="Matern_Matern"
param=list(nugget=0,mean=0,scale_s=0.2/3,scale_t=0.25/3,sill=2,
  smooth_s=0.5,smooth_t=0.5)

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


# Maximum pairwise likelihood fitting of the space time random field:
start = list(scale_s=0.2/3,scale_t=0.25,sill=2,mean=0)
fixed = list(smooth_s=0.5,smooth_t=0.5,nugget=0)
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=coordt, model=model, corrmodel=corrmodel,
             likelihood='Conditional', type='Pairwise',start=start,fixed=fixed,
             optimizer="nlminb",lower=lower,upper=upper,
              neighb=3,maxtime=1)
```

```
##  four location to predict
loc_to_pred=matrix(runif(8),4,2)
## three temporal instants to predict
time=c(0.5,1.5,3.5)


pr=GeoKrig(fit,loc=loc_to_pred,time=time,mse=TRUE)
pr_loc=GeoKrigloc(fit,loc=loc_to_pred,time=time,
  neigh=25,maxtime=1, mse=TRUE)

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



############################################################
#### Example: spatio bivariate  Gaussian local cokriging ######
############################################################
#set.seed(6)
#NN=1500 # number of spatial locations
#x = runif(NN, 0, 1);
#y = runif(NN, 0, 1)
#coords=cbind(x,y)

## setting parameters
#mean_1 = 2; mean_2= -1
#nugget_1 =0;nugget_2=0
#sill_1 =0.5; sill_2 =1;

### correlation parameters
#CorrParam("Bi_Matern")
#scale_1=0.2/3; scale_2=0.15/3; scale_12=0.5*(scale_2+scale_1)
#smooth_1=smooth_2=smooth_12=0.5
#pcol = -0.4
#param= list(nugget_1=nugget_1,nugget_2=nugget_2,
#          sill_1=sill_1,sill_2=sill_2,
#          mean_1=mean_1,mean_2=mean_2,
#          smooth_1=smooth_1, smooth_2=smooth_2,smooth_12=smooth_12,
#          scale_1=scale_1, scale_2=scale_2,scale_12=scale_12,
#          pcol=pcol)

## simulation
#data = GeoSim(coordx=coords, corrmodel="Bi_Matern",model=model,param=param)$data

#fixed=list(mean_1=mean_1,mean_2=mean_2, nugget_1=nugget_1,nugget_2=nugget_2,
#       smooth_1=smooth_1, smooth_2=smooth_2,smooth_12=smooth_12)

#start=list( sill_1=sill_1,sill_2=sill_2,
#        scale_1=scale_1,scale_2=scale_2,scale_12=scale_12, pcol=pcol)

## estimation with maximum likelihood
```

```
#fit = GeoFit(data=data,coordx=coords, corrmodel="Bi_Matern",
 # likelihood="Marginal",type="Pairwise",optimizer="BFGS",neighb=5,
  #start=start,fixed=fixed)

###### co-kriging for the fist component ##############
#xx=seq(0,1,0.022)
#loc_to_pred=as.matrix(expand.grid(xx,xx))
#pr1 = GeoKrigloc(fit,which=1,mse=TRUE,loc=loc_to_pred,neighb=100)

#opar=par(no.readonly = TRUE)
#par(mfrow=c(1,2))
#zlim=c(-2.5,2.5)
#colour = rainbow(100)
#quilt.plot(coords,data[1,] ,col=colour,main = paste(" Fist component"))
#quilt.plot(loc_to_pred,pr1$pred,col=colour,
#          main = paste(" Kriging first component"),ylab="")
#par(opar)
```

GeoKriglocWeights — *Compute kriging weights (and related quantities) for Gaussian / non-Gaussian spatial, spatio-temporal or multivariate random fields*

Description

Given a set of spatial locations (and possibly temporal instants) the function returns the vector (or matrix) of kriging weights, the covariance matrix used in the kriging system and, optionally, the inverse of the left-hand-side matrix dor a specified neighborhhod

Usage

```
GeoKriglocWeights(coordx, coordy=NULL, coordz=NULL, coordt=NULL, coordx_dyn=NULL,
 corrmodel, distance="Eucl", grid=FALSE, loc, neighb=NULL,
 maxdist=NULL, maxtime=NULL, method="cholesky", model="Gaussian",
 n=1, nloc=NULL, param, anisopars=NULL,
 radius=1, sparse=FALSE, time=NULL, which=1,
 copula=NULL, X=NULL, Xloc=NULL, Mloc=NULL,parallel=TRUE)
```

Arguments

`coordx`	Numeric $(d \times 2)$ or $(d \times 3)$ matrix of spatial coordinates. Coordinates on a sphere are accepted (lon/lat in decimal degrees) when `distance = "Sphere"`.
`coordy`	Optional numeric vector giving an additional spatial coordinate dimension. Ignored if `coordx` is already a matrix.
`coordz`	Optional numeric vector giving a third spatial coordinate dimension.
`coordt`	Optional numeric vector of temporal coordinates (length t). If missing, a purely spatial random field is assumed.

coordx_dyn	List of m matrices ($d_t \times 2$) containing time-varying spatial coordinates. See GeoKrig.
corrmodel	Character string naming a valid correlation model. See GeoCovmatrix for admissible choices.
distance	Character string specifying the spatial distance. Default is "Eucl" (Euclidean). See GeoFit.
grid	Logical. If TRUE, coordinates are interpreted as defining a regular grid (see GeoKrig).
loc	Numeric ($n \times 2$) matrix of locations for which the kriging weights are required.
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 integer value indicating the order of the temporal neighborhood.
method	Character string indicating the matrix factorisation used to solve the kriging system: "cholesky" (default) or "svd".
model	Character string specifying the random field type (e.g.\ "Gaussian", "SkewGaussian", "Gamma", ...). See GeoFit.
n	Integer. Number of trials for Binomial random fields (default 1).
nloc	Integer. Number of trials for the prediction locations in Binomial random fields (default 1).
param	Named list of covariance and mean parameters. See CorrParam and GeoCovmatrix.
anisopars	List with components angle and ratio defining geometric anisotropy (optional).
radius	Positive numeric value: sphere radius when coordinates are lon/lat (default 1).
sparse	Logical. If TRUE, sparse-matrix algorithms (package **spam**) are employed. Only effective with compactly supported covariance functions.
time	Numeric vector of length m giving the temporal instants for which weights are required. Ignored if coordt is missing.
which	Integer (1 or 2) selecting the variable whose weights are returned in the bivariate case.
copula	Character string naming a copula when a non-Gaussian dependence structure is used ("Clayton" or "Gaussian").
X	Numeric matrix of spatio-temporal covariates at data locations.
Xloc	Numeric matrix of spatio-temporal covariates at prediction locations.
Mloc	Numeric; Vector of spatio(temporal) estimated means associated to predicted locations.
parallel	Logical; if TRUE then parallelization is performed

Details

The function builds the kriging system

$$\Sigma \mathbf{w} = \boldsymbol{\sigma}_0$$

where Σ is the covariance matrix between observed locations and $\boldsymbol{\sigma}_0$ the vector of covariances between observed and prediction locations. The solution $\mathbf{w}$ is returned together with Σ and, optionally, Σ^{-1}. Universal kriging with covariates is supported; in that case the generalised least squares estimate of the mean parameters is appended. No actual prediction is carried out; for full kriging use GeoKrig.

Value

An object of class KgWeights, a list containing:

weights	Numeric vector (or matrix) of kriging weights.
covmat	Covariance matrix Σ used in the kriging system.
invcov	Inverse of Σ (only if sparse = FALSE).
beta	Estimated regression coefficients when covariates are supplied.
type	Character string: "simple" or "universal".
model	Input model.
corrmodel	Input corrmodel.
spacetime	Logical: TRUE for spatio-temporal case, FALSE otherwise.
sparse	Logical: was a sparse algorithm used?
loc	Copy of the loc argument.
time	Copy of the time argument (if any).

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uai.cl>, Víctor Morales-Oñate, <victor.morales@uv.cl>, Christian Caamaño-Carrillo, <chcaaman@ubiobio.cl>

References

Gaetan, C. and Guyon, X. (2010) *Spatial Statistics and Modeling*. Springer-Verlag, New York.

See Also

GeoKrig for full prediction, GeoKrigloc for local prediction, GeoCovmatrix for covariance model details, `GeoKrigWeights` *Compute kriging weights (and related quantities) for Gaussian / non-Gaussian spatial, spatio-temporal or multivariate random fields*

Description

Given a set of spatial locations (and possibly temporal instants) the function returns the vector (or matrix) of kriging weights, the covariance matrix used in the kriging system and, optionally, the inverse of the left-hand-side matrix.

Usage

```
GeoKrigWeights(
    coordx, coordy = NULL, coordz = NULL, coordt = NULL,
    coordx_dyn = NULL, corrmodel, distance = "Eucl",
    grid = FALSE, loc, method = "cholesky", model = "Gaussian",
    n = 1, nloc = NULL, param, anisopars = NULL, radius = 1,
    sparse = FALSE, time = NULL, which = 1, copula = NULL,
    X = NULL, Xloc = NULL, Mloc=NULL)
```

Arguments

`coordx`	Numeric $(d \times 2)$ or $(d \times 3)$ matrix of spatial coordinates. Coordinates on a sphere are accepted (lon/lat in decimal degrees) when `distance = "Sphere"`.
`coordy`	Optional numeric vector giving an additional spatial coordinate dimension. Ignored if `coordx` is already a matrix.
`coordz`	Optional numeric vector giving a third spatial coordinate dimension.
`coordt`	Optional numeric vector of temporal coordinates (length t). If missing, a purely spatial random field is assumed.
`coordx_dyn`	List of m matrices $(d_t \times 2)$ containing time-varying spatial coordinates. See `GeoKrig`.
`corrmodel`	Character string naming a valid correlation model. See `GeoCovmatrix` for admissible choices.
`distance`	Character string specifying the spatial distance. Default is `"Eucl"` (Euclidean). See `GeoFit`.
`grid`	Logical. If `TRUE`, coordinates are interpreted as defining a regular grid (see `GeoKrig`).
`loc`	Numeric $(n \times 2)$ matrix of locations for which the kriging weights are required.
`method`	Character string indicating the matrix factorisation used to solve the kriging system: `"cholesky"` (default) or `"svd"`.
`model`	Character string specifying the random field type (e.g.\ `"Gaussian"`, `"SkewGaussian"`, `"Gamma"`, . . .). See `GeoFit`.
`n`	Integer. Number of trials for Binomial random fields (default `1`).

nloc	Integer. Number of trials for the prediction locations in Binomial random fields (default 1).
param	Named list of covariance and mean parameters. See CorrParam and GeoCovmatrix.
anisopars	List with components angle and ratio defining geometric anisotropy (optional).
radius	Positive numeric value: sphere radius when coordinates are lon/lat (default 1).
sparse	Logical. If TRUE, sparse-matrix algorithms (package **spam**) are employed. Only effective with compactly supported covariance functions.
time	Numeric vector of length m giving the temporal instants for which weights are required. Ignored if coordt is missing.
which	Integer (1 or 2) selecting the variable whose weights are returned in the bivariate case.
copula	Character string naming a copula when a non-Gaussian dependence structure is used ("Clayton" or "Gaussian").
X	Numeric matrix of spatio-temporal covariates at data locations.
Xloc	Numeric matrix of spatio-temporal covariates at prediction locations.
Mloc	Numeric; Vector of spatio(temporal) estimated means associated to predicted locations.

Details

The function builds the kriging system

$$\Sigma \mathbf{w} = \boldsymbol{\sigma}_0$$

where Σ is the covariance matrix between observed locations and $\boldsymbol{\sigma}_0$ the vector of covariances between observed and prediction locations. The solution $\mathbf{w}$ is returned together with Σ and, optionally, Σ^{-1}. Universal kriging with covariates is supported; in that case the generalised least squares estimate of the mean parameters is appended. No actual prediction is carried out; for full kriging use GeoKrig.

Value

An object of class KgWeights, a list containing:

weights	Numeric vector (or matrix) of kriging weights.
covmat	Covariance matrix Σ used in the kriging system.
invcov	Inverse of Σ (only if sparse = FALSE).
beta	Estimated regression coefficients when covariates are supplied.
type	Character string: "simple" or "universal".
model	Input model.
corrmodel	Input corrmodel.
spacetime	Logical: TRUE for spatio-temporal case, FALSE otherwise.
sparse	Logical: was a sparse algorithm used?
loc	Copy of the loc argument.
time	Copy of the time argument (if any).

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uai.cl>, Víctor Morales-Oñate, <victor.morales@uv.cl>, Christian Caamaño-Carrillo, <chcaaman@ubiobio.cl>

References

Gaetan, C. and Guyon, X. (2010) *Spatial Statistics and Modeling*. Springer-Verlag, New York.

See Also

GeoKrig for full prediction, GeoKrigloc for local prediction, GeoCovmatrix for covariance model details,

Examples

```
library(GeoModels)
################################################################

################################################################
###
### Example 1. Spatial kriging weights for
### Gaussian random fields with Gen wendland correlation.
###
################################################################

model="Gaussian"
set.seed(79)
x = runif(300, 0, 1)
y = runif(300, 0, 1)
coords=cbind(x,y)

corrmodel = "GenWend"
mean=0; sill=5; nugget=0
scale=0.2;smooth=0;power2=4

param=list(mean=mean,sill=sill,nugget=nugget,scale=scale,smooth=smooth,power2=power2)

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

xx=seq(0,1,0.25)
loc_to_pred=as.matrix(expand.grid(xx,xx))

W=GeoKrigWeights(,coordx=coords,loc=loc_to_pred,corrmodel=corrmodel,
       model=model,param=param)

dim(W$weights)  ### kriging weights
```

GeoNA *Deleting NA values (missing values) from a spatial or spatio-temporal dataset.*

Description

The function deletes NA values from a spatial or spatio-temporal dataset

Usage

```
GeoNA(data, coordx, coordy=NULL,coordz=NULL, coordt=NULL,
coordx_dyn=NULL, grid=FALSE, X=NULL, setting="spatial")
```

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.
- `coordx` A numeric $(d \times 2)$-matrix or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.
- `coordz` A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is `NULL`.
- `coordt` A numeric vector giving 1-dimension of temporal coordinates; 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`
- `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).
- `X` Numeric; Matrix of spatio(temporal) covariates in the linear mean specification.
- `setting` String; are data spatial, spatio-temporal or spatial bivariate (respectively `spatial`, `spacetime`, `bivariate`)

Value

Returns a list containing the following components:

- `coordx` A d-dimensional vector of spatial coordinates;
- `coordy` A d-dimensional vector of spatial coordinates;
- `coordt` A t-dimensional vector of temporal coordinates;
- `data` The data without NAvalues
- `grid` `TRUE` if the spatial data are observed in a regular grid, otherwise `FALSE`;

perc	The percentage of NA values .
setting	Are data of spatial or spatio-temporal or spatial bivariate type
X	Covariates matrix

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

Examples

```
library(GeoModels)

# 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 = "Matern"
mean=0
sill=1
nugget=0
scale=0.3/3
smooth=0.5
param=list(mean=mean,sill=sill,nugget=nugget,scale=scale,smooth=smooth)

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


data[1:100]=NA
# removing NA
a=GeoNA(data,coordx=coords)
a$perc # percentage of NA values
#a$coordx# spatial coordinates without missing values
#a$data # data without missinng values
```

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

Description

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

Usage

```
GeoNeighborhood(data=NULL, coordx, coordy=NULL,coordz=NULL,
coordt=NULL, coordx_dyn=NULL, bivariate=FALSE,
                 distance="Eucl", grid=FALSE,
                 loc, neighb=NULL,maxdist=NULL,
                 maxtime=NULL, radius=1, time=NULL,
                 X=NULL,M=NULL,spobj=NULL,spdata=NULL,
                 parallel=FALSE,ncores=NULL)
```

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 or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.
`coordz`	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is `NULL`.
`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 spatial neighborhood.
`maxdist`	Numeric; a positive value indicating the maximum spatial distance considered in the spatial neighborhood selection.
`maxtime`	Numeric; an optional positive integer indicating the order of temporal neighborhood.
`radius`	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is 1.
`time`	Numeric; a value giving the temporal instant for which a neighborhood is computed.
`X`	Numeric; an optional Matrix of spatio (temporal) covariates.

M	Numeric; an estimated spatio (temporal) mean vector.
spobj	An object of class sp or spacetime
spdata	Character:The name of data in the sp or spacetime object
parallel	Logical; if TRUE then parallelizzation is used
ncores	Numeric; number of cores involved in parallelization.

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.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

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)

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,neighb=3,maxtime=0.5)

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

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

###################################################
#### 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,neighb=5)

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

GeoNeighbSelect *A brute force algorithm for spatial or spatiotemoral optimal neighbo-ord selection for pairwise composite likelihood estimation.*

Description

The procedure performs different pairwise composite likelihood estimation using user's specified spatial or spatiotemporal neighboords in the weight function. The neighbor minimizing the sum of the squared differences between the estimated semivariogram and the empirical semivariogram is selected. The procedure needs an object obtained using the GeoVariogram function.

Usage

```
GeoNeighbSelect(data, coordx, coordy=NULL,coordz=NULL, coordt=NULL, coordx_dyn=NULL,
    copula=NULL,corrmodel=NULL, distance="Eucl",fixed=NULL,anisopars=NULL,
    est.aniso=c(FALSE,FALSE), grid=FALSE, likelihood='Marginal',lower=NULL,
    neighb=c(1,2,3,4,5),p_neighb=1,maxtime=Inf, memdist=TRUE,model='Gaussian',
    n=1, ncores=NULL,optimizer='Nelder-Mead', parallel=FALSE,
    bivariate=FALSE,radius=1, start=NULL,type='Pairwise', upper=NULL,
   weighted=FALSE,X=NULL,nosym=FALSE,spobj=NULL,spdata=NULL,vario=NULL,progress=TRUE)
```

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 or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.

coordz	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is NULL.
coordt	A numeric vector assigning 1-dimension of temporal coordinates. Optional argument, the default is NULL then a spatial random fields 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
copula	String; the type of copula. It can be "Clayton" or "Gaussian"
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.
anisopars	A list of two elements: "angle" and "ratio" i.e. the anisotropy angle and the anisotropy ratio, respectively.
est.aniso	A bivariate logical vector providing which anisotropic parameters must be estimated.
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 Section **Details** in GeoFit).
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 bobyqa or optimize. The names of the list must be the same of the names in the start list.
neighb	Numeric; a vector of positive integers indicating the order of neighborhood in the weight function of composite likelihood (see Section **Details** in GeoFit).
p_neighb	Numeric; a value in $(0, 1]$ that specifies the expected fraction of nearest-neighbor pairs retained through stochastic thinning. If is equal to 1 (default), no thinning is applied and all nearest-neighbor pairs are used. If it is lower than 1, nearest-neighbor pairs are randomly retained using independent Bernoulli sampling.
maxtime	Numeric; an optional positive integer indicating the order of temporal neighborhood in the composite likelihood computation.
memdist	Logical; if TRUE then all the distances useful in the composite likelihood estimation are computed before the optimization. FALSE is deprecated.
model	String; the type of random fields and therefore the densities associated to the likelihood objects. Gaussian is the default, see the Section **Details** in GeoFit.
n	Numeric; number of trials in a binomial random fields; number of successes in a negative binomial random fields
ncores	Numeric; the number of cores involved in the parallelization
optimizer	String; the optimization algorithm (see optim for details). Nelder-Mead is the default. Other possible choices are nlm, BFGS, SANN, L-BFGS-B and nlminb and bobyqa. In these last three cases upper and lower bounds can be passed by the user. In the case of one-dimensional optimization, the function optimize is used.

parallel — Logical; if TRUE the procedure is parallelized using dofuture.

bivariate — Logical; if TRUE the bivariate case is considered.

radius — Numeric; the radius of the sphere in the case of lon-lat coordinates. Default value is 1.

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 Section **Details** in GeoFit).

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

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 bobyqa or nlminb or optimize. The names of the list must be the same of the names in the start list.

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

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

nosym — Logical; if TRUE simmetric weights are not considered. This allows a faster but less efficient CL estimation.

spobj — An object of class sp or spacetime

spdata — Character:The name of data in the sp or spacetime object

vario — An object of the class GeoVariogram obtained using the GeoVariogram function

progress — Logic; If TRUE then a progress bar is shown.

Details

The procedure performs different pairwise composite likelihood estimation using user's specified spatial or spatiotemporal neighboords in the weight function. The neighbor minimizing the sum of the squared differences between the estimated semivariogram and the empirical semivariogram is selected. The procedure needs an object obtained using the GeoVariogram function.

Value

Returns a list containing the estimates for each neighborhood, the optimal neighborhood selected, and, if the selected neighborhood is large, a recommended alternative.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

Examples

```
library(GeoModels)


######### spatial case
set.seed(32)
N=500  # number of location sites
x <- runif(N, 0, 1)
y <- runif(N, 0, 1)
coords <- cbind(x,y)
mean <- 0.2
# Set the covariance model's parameters:
corrmodel <- "Matern"
sill <- 1;nugget <- 0
scale <- 0.2/3;smooth=0.5

model="Gaussian"
param<-list(mean=mean,sill=sill,nugget=nugget,scale=scale,smooth=smooth)
# Simulation
data <- GeoSim(coordx=coords,corrmodel=corrmodel, param=param,model=model)$data
I=Inf
fixed<-list(nugget=nugget)
start<-list(mean=mean,scale=scale,smooth=smooth,sill=sill)
lower<-list(mean=-I,scale=0,sill=0,smooth=0)
upper<-list(mean=I,scale=I,sill=I,smooth=I)

vario = GeoVariogram(coordx=coords,data=data,maxdist=0.3,numbins=15)

neighb=c(1,2,3,4) ## trying  different neighbs
selK <- GeoNeighbSelect(vario=vario,data=data,coordx=coords,corrmodel=corrmodel,
                        model=model,neighb=neighb,
                        likelihood="Conditional",type="Pairwise",parallel=FALSE,
                        optimizer="nlminb",lower=lower,upper=upper,
                        start=start,fixed=fixed)
print(selK$best_neighb) ## selected neighbor
```

GeoNeighIndex   *Spatial or spatiotemporal near neighbour indices.*

Description

The function returns the indices associated with a given spatial (temporal) neighbour and/or distance. Optionally, a stochastic thinning mechanism can be applied to randomly retain only a fraction of the indices.

Usage

```
GeoNeighIndex(coordx, coordy=NULL, coordz=NULL, coordt=NULL,
              coordx_dyn=NULL, distance="Eucl", neighb=4,
              maxdist=NULL, maxtime=1, radius=1,
              bivariate=FALSE, p_neighb=1, thin_method="iid")
```

Arguments

coordx	A numeric $(d \times 2)$-matrix or $(d \times 3)$-matrix. 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; optional argument, default is NULL.
coordz	A numeric vector giving 1-dimension of spatial coordinates; optional argument, default is NULL.
coordt	A numeric vector assigning 1-dimension of temporal coordinates. Optional argument, default is NULL; if 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, default is NULL.
distance	String; the name of the spatial distance. Default is "Eucl" (euclidean distance). See the Section **Details** of GeoFit.
neighb	Numeric; an optional (vector of) positive integer indicating the order of neighborhood. See the Section **Details** for more information.
maxdist	A numeric value denoting the maximum spatial distance; see **Details**.
maxtime	A numeric value denoting the maximum temporal distance; see **Details**.
radius	Numeric; a value indicating the radius of the sphere when using great-circle distances. Default value is 1.
bivariate	Logical; if FALSE (default) the data are interpreted as univariate spatial or spatio-temporal realisations. Otherwise, they are interpreted as a realization from a bivariate field.
p_neighb	Numeric; a value in $(0, 1]$ specifying the expected fraction of nearest-neighbor pairs retained through stochastic thinning. If 1 (default), no thinning is applied and all nearest-neighbor pairs are used. If <1, pairs are randomly retained using independent Bernoulli sampling.
thin_method	String; thinning scheme used when p_neighb<1. Default is "iid"(independent Bernoulli thinning).

Details

The function returns spatial or spatio-temporal indices of the pairs that are neighbours of a given order and/or within a given distance threshold.

Value

Returns a list containing the following components:

colidx	First vector of indices.
rowidx	Second vector of indices.
lags	Vector of spatial distances.
lagt	Vector of temporal distances.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>, https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian Caamaño-Carrillo, <chcaaman@ubiobio.cl>, https://www.researchgate.net/profile/Christian-Caamano

Examples

```
require(GeoModels)
NN <- 400
coords <- cbind(runif(NN), runif(NN))

corrmodel <- "Matern"
scale <- 0.5/3
param <- list(mean=0, sill=1, nugget=0, scale=scale, smooth=0.5)

set.seed(951)
data <- GeoSim(coordx=coords, corrmodel=corrmodel,
              model="Gaussian", param=param)$data

sel <- GeoNeighIndex(coordx=coords, neighb=5)

data1 <- data[sel$colidx]
data2 <- data[sel$rowidx]

## plotting pairs that are neighbours of order 5
plot(data1, data2, xlab="", ylab="",
     main="h-scatterplot, neighb=5")
```

GeoOutlier *Spatio (temporal) outliers detection*

Description

Given a set of spatio (temporal) locations and data, the procedure select the spatial or spatiotemporal ouliers using a specific algorithm.

Usage

```
GeoOutlier(data, coordx, coordy=NULL,coordz=NULL, coordt=NULL, coordx_dyn=NULL,
              distance="Eucl", grid=FALSE,  neighb=10,alpha=0.001,
              method="Z-Median", radius=1, bivariate=FALSE,X=NULL)
```

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 or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.
coordz	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is `NULL`.
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`
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).
neighb	Numeric; an optional positive integer indicating the order of neighborhoodused for Z-Median algorithm.
alpha	Numeric; a numeric value between 0 and 1 used for Z-Median algorithm.
method	String; The name of the algorithm for detecting spatial ouliers. Default is Z-median proposed in Chen et al. (2008)
radius	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is 1.
bivariate	If TRUE then data is considered as spatial bivariate data.
X	Numeric; an optional Matrix of spatio (temporal) covariates.

Value

Return a matrix or a list containing the dected spatial or spatio-temporal outliers

Author(s)

Moreno Bevilacqua, `<moreno.bevilacqua89@gmail.com>`,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, `<victor.morales@uv.cl>`, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, `<chcaaman@ubiobio.cl>`,https://www.researchgate.net/profile/Christian-Caamano

References

Chen D, Lu C, Kou Y, Chen F (2008) On detecting spatial outliers. Geoinformatica 12:455–475

Bevilacqua M., Caamaño C., Arellano-Valle R. B., Camilo Gomez C. (2022) A class of random fields with two-piece marginal distributions for modeling point-referenced data with spatial outliers. *Test* 10.1007/s11749-021-00797-5

Examples

```
library(GeoModels)
set.seed(1428)
NN = 1500
coords = cbind(runif(NN),runif(NN))
###
scale=0.5/3
corrmodel = "Matern";

param = list(mean=0,sill=1,nugget=0,scale=scale,smooth=0.5,skew=0)
data = GeoSim(coordx = coords,corrmodel = corrmodel,
                  model = "TwoPieceGaussian",param = param)$data

K=15         #parameter for outliers detection alghoritm
alpha=0.005  #parameter for outliers detection alghoritm
outlier=GeoOutlier(data=data, coordx = coords,neighb=K,alpha=alpha)
quilt.plot(coords,data)
for (i in 1:nrow(outlier)) plotrix::draw.circle(outlier[i,1], outlier[i,2],radius=0.02,lwd=2)
nrow(outlier) # number of outliers

param = list(mean=0,sill=1,nugget=0.4,scale=scale,smooth=0.5)
data = GeoSim(coordx = coords,corrmodel = corrmodel,
                  model = "Gaussian",param = param)$data

K=15         #parameter for outliers detection alghoritm
alpha=0.005  #parameter for outliers detection alghoritm
outlier=GeoOutlier(data=data, coordx = coords,neighb=K,alpha=alpha)
quilt.plot(coords,data)
for (i in 1:nrow(outlier)) plotrix::draw.circle(outlier[i,1], outlier[i,2],radius=0.02,lwd=2)
nrow(outlier) # number of outliers
```

GeoPit *Probability integral or normal score tranformation*

Description

The procedure for a given GeoFit object applies the probability integral tranformation or the normal score transformation to the data

Usage

```
GeoPit(object,type="Uniform")
```

Arguments

object	A GeoFit object
.	
type	The type of transformation. If "Uniform" then the probability integral tranformation is performed. If "Gaussian" then the normal score transformation is performed.

Value

Returns an (updated) object of class GeoFit

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

Examples

```
library(GeoModels)

model="Beta2"
copula="Clayton"

set.seed(221)
NN=800
x <- runif(NN);y <- runif(NN)
coords=cbind(x,y)


shape=1.5
scale=0.2;power2=4
smooth=0
nugget=0
nu=8

corrmodel="GenWend"

min=-2;max=1
mean=0


param=list(smooth=smooth,power2=power2, min=min,max=max,
             mean=mean, nu=nu,
             scale=scale,nugget=nugget,shape=shape)

optimizer="nlminb"

data <- GeoSimCopula(coordx=coords, corrmodel=corrmodel,
model=model,param=param,copula=copula)$data

I=50
fixed<-list(nugget=nugget,sill=1,scale=scale,smooth=smooth,power2=power2,min=min,max=max,nu=nu)
start<-list(shape=shape,mean=mean)
lower<-list(shape=0,mean=-I)
upper<-list(shape=10,mean=I)

#### maximum independence likelihood
fit1 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
```

```
model=model,likelihood="Marginal",type="Independence",
                      optimizer=optimizer,lower=lower,
                      upper=upper,copula=copula,
                    start=start,fixed=fixed)

## PIT transformation
aa=GeoPit(fit1,type="Uniform")
hist(aa$data,freq=FALSE)
GeoScatterplot(aa$data,coords,neighb=c(1,2))
## Normal score transformation
bb=GeoPit(fit1,type="Gaussian")
hist(bb$data,freq=FALSE)
GeoScatterplot(bb$data,coords,neighb=c(1,2))
```

GeoQQ *Quantile-quantile plot*

Description

Based on a GeoFit object, the procedure plots a quantile-quantile plot or compares the fitted density with the histogram of the data. It is useful as diagnostic tool.

Usage

```
GeoQQ(fit,type="Q",add=FALSE,ylim=c(0,1),xlim=NULL,breaks=10,...)
```

Arguments

`fit`	A GeoFit object possibly obtained from `GeoResiduals`.
`type`	The type of plot. If Q then a qq-plot (default) is performed. If D then a comparison between histrogram and the estimated marginal density is performed
`add`	Logical; if TRUE the the estimated density ia added over an existing one
`ylim`	Numeric; a vector of length 2 used for the ylab parameter of the histogram plot.
`xlim`	Numeric; a vector of length 2 used for the xlab parameter of the histogram plot.
`breaks`	Numeric; an integer number specifyng the number of cells ofthe histogram plot if the option type=D is chosen.
`...`	Optional parameters passed to the plot function.

Value

Produces a plot. No values are returned.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

Examples

```
library(GeoModels)


##################
### Example 1
##################
set.seed(21)
model="Tukeyh";tail=0.1
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,tail=tail,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,tail=tail)
fixed=list(nugget=nugget,sill=sill,power2=power2)
# Maximum composite-likelihood fitting
fit = GeoFit(data,coordx=coords, corrmodel=corrmodel,model=model,X=X,
                    likelihood="Conditional",type='Pairwise',start=start,
                    fixed=fixed,neighb=4)

res=GeoResiduals(fit)
GeoQQ(res,type="Q")
GeoQQ(res,type="D",lwd=2,ylim=c(0,0.5),breaks=20)
```

```
##################
### Example 2
##################
set.seed(21)
model="Weibull";shape=1.5
N=600 # 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 = 0

# correlation parameters:
corrmodel = "Matern"
smooth=0.5
nugget = 0
scale = 0.2/3


param=list(mean=mean, sill=1, nugget=nugget,
             scale=scale,smooth=smooth, shape=shape)
# Simulation of the Gaussian RF:
data = GeoSim(coordx=coords, corrmodel=corrmodel,model=model,param=param)$data

start=list(mean=mean, scale=scale,shape=shape)
I=Inf
lower=list(mean=-I, scale=0,shape=0)
upper=list(mean= I, scale=I,shape=I)
I=Inf
fixed=list(nugget=nugget,sill=1,smooth=smooth)
# Maximum composite-likelihood fitting
fit = GeoFit(data,coordx=coords, corrmodel=corrmodel,model=model,
                    likelihood="Conditional",type='Pairwise',start=start,
                    optimizer="nlminb",lower=lower,upper=upper,
                    fixed=fixed,neighb=3)
GeoQQ(fit,type="Q")
GeoQQ(fit,type="D",lwd=2,ylim=c(0,1),breaks=20)
```

GeoResiduals *Computes fitted covariance and/or variogram*

Description

The procedure return a GeoFit object associated to the estimated residuals. For a random field Y defined on the real line (Gaussian, Skew Gaussian, Tukeyh etcc) they are computed as (Y-m)/sqrt(v) where m and v are the estimated mean and variance respectively. For a random field Y defined on the positive real line (Gamma, Weibull, Log-Gaussian) they are computed as Y/m where m is the estimated mean. In the first case residuals have zero mean and unut variance with a specific distribution defined on the real line. In the second case residuals have unit mean with a specific distribution defined on the positive real line. When the function is coupled with the functions GeoQQ and GeoCovariogram, it is useful as diagnostic tool (See examples).

Usage

```
GeoResiduals(fit)
```

Arguments

fit	A fitted object obtained from the GeoFit.

Value

Returns an (updated) object of class GeoFit

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit.

Examples

```
library(GeoModels)



###########################
###Example 1: Residuals using a Gaussian RF
###########################
set.seed(211)
model="Gaussian";
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,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,sill=sill)
fixed=list(nugget=nugget,power2=power2)
# Maximum composite-likelihood fitting
fit = GeoFit(data,coordx=coords, corrmodel=corrmodel,model=model,X=X,
                    likelihood="Conditional",type='Pairwise',start=start,
                    fixed=fixed,neighb=3)

res=GeoResiduals(fit)
mean(res$data) # should be approx 0
var(res$data) # should be approx 1
# checking goodness of fit marginal model
GeoQQ(res);GeoQQ(res,type="D",col="red",ylim=c(0,0.5),breaks=20);
# Empirical estimation of the variogram for the residuals:
vario = GeoVariogram(res$data,coordx=coords,maxdist=0.5)
# Comparison between empirical amd estimated semivariogram for the residuals
GeoCovariogram(res, show.vario=TRUE, vario=vario,pch=20)




###########################
###Example 2: Residuals using a Weibull RF
###########################
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

I=Inf
start=list(mean=mean,mean1=mean1, scale=scale,shape=shape)
lower=list(mean=-I,mean1=-I, scale=0,shape=0)
upper=list(mean= I,mean1= I, scale=I,shape=I)
fixed=list(nugget=nugget,sill=sill,power2=power2)
# Maximum composite-likelihood fitting
fit = GeoFit(data,coordx=coords, corrmodel=corrmodel,model=model,X=X,
                    likelihood="Conditional",type='Pairwise',start=start,
                   optimizer="nlminb", lower=lower,upper=upper,
                    fixed=fixed,neighb=3)


res=GeoResiduals(fit)
mean(res$data) # should be approx 1
# checking goodness of fit marginal model
GeoQQ(res);GeoQQ(res,type="D",lwd=2,ylim=c(0,1.7),breaks=20);
# Empirical estimation of the variogram for the residuals:
vario = GeoVariogram(res$data,coordx=coords,maxdist=0.5)
# Comparison between empirical amd estimated semivariogram for the residuals
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, coordz=NULL, coordt=NULL, coordx_dyn=NULL,
            distance='Eucl', grid=FALSE, maxdist=NULL,neighb=NULL,
            times=NULL, numbins=4, radius=1, 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 or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.
coordz	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is `NULL`.
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 1.
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.
...	Optional parameters passed to the plot function.

Details

h-scatterplot is the plot of the pair values that are neighborhood of a certain order or with distances belonging to a certain interval. In the first case a (vector of) neighborhood must be specified. In the second case a maximum distance (maxdist) and a number of lag-bins (numbins) must be specified. The method based on neighborhoods is recommended in particular for large datasets.

Value

Produces a plot. No values are returned.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

Examples

```
library(GeoModels)
set.seed(514)

NN = 600
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

# h-scatterplots for given a vector of neighborhoods
GeoScatterplot(data,coords,neighb=c(2,4))
```

GeoScores *Computation of predictive scores*

Description

The function computes some predictive scores for a spatial, spatiotemporal and bivariate Gaussian RFs

Usage

```
GeoScores(data_to_pred, probject=NULL,pred=NULL,mse=NULL,
   score=c("brie","crps","lscore","pit","pe", "intscore", "coverage"))
```

Arguments

data_to_pred	A numeric vector of data to predict about a response
probject	A Geokrig object obtained using the function Geokrig
pred	A numeric vector with predictions for the response.
mse	a numeric vector with prediction variances.
score	A character defining what statistic of the prediction errors should be computed. Possible values are lscore, crps, brie and pe. In the latter case scores based on prediction errors such as rmse, mae, mad are computed. Finally, the character pit allows to compute the probability integral transform for each value

Details

GeoScores computes the items required to evaluate the diagnostic criteria proposed by Gneiting et al. (2007) for assessing the calibration and the sharpness of probabilistic predictions of (cross-) validation data. To this aim, GeoScores uses the assumption that the prediction errors are Gaussian with zero mean and standard deviations equal to the Kriging standard errors. This assumption is an approximation if the errors are not Gaussian.

Value

Returns a list containing the following informations:

lscore	Logarithmic predictive score
crps	Continuous ranked probability predictive score
rmse	Root mean squared error
mae	Mean absolute error
mad	Median absolute error
pit	A vector of probability integral transformation
intscore	Mean interval score
coverage	Coverage of the prediction intervals

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

References

Gneiting T. and Raftery A. *Strictly Proper Scoring Rules, Prediction, and Estimation*. Journal of the American Statistical Association, **102**

Examples

```
library(GeoModels)

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

library(GeoModels)
model="Gaussian"
set.seed(79)
N=1000
x = runif(N, 0, 1)
y = runif(N, 0, 1)
coords=cbind(x,y)

# Set the exponential cov parameters:
corrmodel = "GenWend"
mean=0; sill=5; nugget=0
scale=0.2;smooth=0;power2=4

param=list(mean=mean,sill=sill,nugget=nugget,scale=scale,smooth=smooth,power2=power2)

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

```
               param=param)$data


sel=sample(1:N,N*0.8)
coords_est=coords[sel,]; coords_to_pred=coords[-sel,]
data_est=data[sel]; data_to_pred=data[-sel]

## estimation with pairwise likelihood
fixed=list(nugget=nugget,smooth=0,power2=power2)
start=list(mean=0,scale=scale,sill=1)
I=Inf
lower=list(mean=-I,scale=0,sill=0)
upper=list(mean= I,scale=I,sill=I)
# Maximum pairwise likelihood fitting :
fit = GeoFit(data_est, coordx=coords_est, corrmodel=corrmodel,model=model,
                    likelihood='Marginal', type='Pairwise',neighb=3,
                    optimizer="nlminb", lower=lower,upper=upper,
                    start=start,fixed=fixed)
# locations to predict
xx=seq(0,1,0.03)
loc_to_pred=as.matrix(expand.grid(xx,xx))

pr=GeoKrig(loc=coords_to_pred,coordx=coords_est,corrmodel=corrmodel,
       model=model,param= param, data=data_est,mse=TRUE)

Pr_scores =GeoScores(data_to_pred,pred=pr$pred,mse=pr$mse)
Pr_scores$rmse;Pr_scores$brie
hist(Pr_scores$pit,freq=FALSE)
```

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,coordz=NULL, coordt=NULL, coordx_dyn=NULL, corrmodel,
      distance="Eucl", grid=FALSE, method="cholesky",
      model='Gaussian', n=1, param,anisopars=NULL,radius=1,
      sparse=FALSE,X=NULL,spobj=NULL,nrep=1,progress=TRUE)
```

Arguments

coordx — A numeric $(d \times 2)$-matrix or $(d \times 3)$-matrix. 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; Optional argument, the default is NULL.
coordz	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is NULL.
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.
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).
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**.
anisopars	A list of two elements "angle" and "ratio" i.e. the anisotropy angle and the anisotropy ratio, respectively.
radius	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is 1.
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.
spobj	An object of class sp or spacetime
nrep	Numeric; Numbers of indipendent replicates.
progress	Logic; If TRUE then a progress bar is shown.

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;
coordz	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;
method	The method of simulation
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;
spacetime	TRUE if spatio-temporal and FALSE if spatial RF;
nrep	The number of indipendent replicates;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

Examples

```
library(GeoModels)
library(mapproj)


################################################################
###
### Example 1. Simulation of a spatial Gaussian RF
### with Matern and Generalized Wendland correlations
###############################################################

# Define the spatial-coordinates of the points:
x <- runif(500);y <- runif(500)
coords=cbind(x,y)
set.seed(261)
# Simulation of a spatial Gaussian RF with Matern correlation function
data1 <- GeoSim(coordx=coords, corrmodel="Matern", param=list(smooth=0.5,
             mean=0,sill=1,scale=0.4/3,nugget=0))$data

set.seed(261)
data2 <- GeoSim(coordx=coords,  corrmodel="GenWend", param=list(smooth=0,
              power2=4,mean=0,sill=1,scale=0.4,nugget=0))$data
opar=par(no.readonly = TRUE)
```

```
par(mfrow=c(1,2))
quilt.plot(coords,data1,main="Matern",xlab="",ylab="")
quilt.plot(coords,data2,main="Wendland",xlab="",ylab="")
par(opar)


################################################################
###
### Example 2. Simulation of a spatial geometric RF
### with  underlying Wend0 correlation
###
################################################################

# Define the spatial-coordinates of the points:
x <- runif(800);y <- runif(800)
coords <- cbind(x,y)
set.seed(251)
# Simulation of a spatial Binomial RF:
sim <- GeoSim(coordx=coords, corrmodel="Wend0",
             model="BinomialNeg",n=1,sparse=TRUE,
             param=list(nugget=0,mean=0,scale=.2,power2=4))

quilt.plot(coords,sim$data,nlevel=max(sim$data),col=terrain.colors(max(sim$data+1)))

################################################################
###
### Example 3. Simulation of a spatial Weibull  RF
### with  underlying Matern correlation on a regular grid
###
###############################################################
# 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="Matern",model="Weibull",
         param=list(shape=1.2,mean=0,scale=0.3/3,nugget=0,smooth=0.5))$data
image.plot(x,y,data1,main="Weibull RF",xlab="",ylab="")

################################################################
###
### Example 4. Simulation of a spatial t  RF
### with  with  underlying Generalized Wendland 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(500, 0, 2)
y <- runif(500, 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  bivariate Matern correlation model
###
###############################################################

# Define the spatial-coordinates of the points:
x <- runif(500, 0, 2)
y <- runif(500, 0, 2)
coords <- cbind(x,y)

# Simulation of a bivariate spatial Gaussian RF:
# with a separable Bivariate Matern
param=list(mean_1=4,mean_2=2,smooth_1=0.5,smooth_2=0.5,smooth_12=0.5,
           scale_1=0.12,scale_2=0.1,scale_12=0.15,
           sill_1=1,sill_2=1,nugget_1=0,nugget_2=0,pcol=0.5)
data <- GeoSim(coordx=coords,corrmodel="Bi_matern",
              param=param)$data
```

```
opar=par(no.readonly = TRUE)
par(mfrow=c(1,2))
quilt.plot(coords,data[1,],col=terrain.colors(100),main="1",xlab="",ylab="")
quilt.plot(coords,data[2,],col=terrain.colors(100),main="2",xlab="",ylab="")
par(opar)


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



coordt=1:5

# Define the spatial-coordinates of the points:
x <- runif(50, 0, 2)
y <- runif(50, 0, 2)
coords <- cbind(x,y)

param<-list(nugget=0,mean=0,scale_s=0.2/3,scale_t=2/3,sill=1,smooth_s=0.5,smooth_t=0.5)
data <- GeoSim(coordx=coords, coordt=coordt, corrmodel="Matern_Matern",
                     param=param)$data
dim(data)

################################################################
###
### Example 8. Simulation of a  spatio temporal Gaussian RF.
### observed on  dynamical location sites with double Matern correlation
###
###############################################################

# Define the dynamical spatial-coordinates of the points:

coordt=1:5
coordx_dyn=list()
maxN=30
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,smooth_s=0.5,smooth_t=0.5)
data <- GeoSim(coordx_dyn=coordx_dyn, coordt=coordt, corrmodel="Matern_Matern",
                     param=param)$data
```

```
## spatial realization at first temporal instants
data[[1]]
## spatial realization at third temporal instants
data[[3]]



################################################################
###
### Example 9. Simulation of a Gaussian RF
###  with a Wend0 correlation in the north emisphere of the planet earth
### using geodesic distance
###############################################################
distance="Geod";radius=6371

NN=3000 ## 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=1000,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
#require(globe)
#globe::globeearth(eye=place("newyorkcity"))
#globe::globepoints(loc=coords,pch=20,col = cm.colors(length(data),alpha=0.4)[rank(data)])
```

`GeoSimapprox` *Fast simulation of Gaussian and non Gaussian Random Fields.*

Description

Simulation of Gaussian and some non Gaussian spatial, spatio-temporal and spatial bivariate random fields using two approximate methods of simulation: circulant embeeding and spectral turning band. (see Examples).

Usage

```
GeoSimapprox(coordx, coordy=NULL, coordz=NULL,coordt=NULL,
coordx_dyn=NULL,corrmodel, distance="Eucl",GPU=NULL,
grid=FALSE,max.ext=1,
method="TB", L=1000,model='Gaussian',parallel=FALSE,ncores=NULL,
n=1,param,anisopars=NULL, radius=6371,X=NULL,spobj=NULL,
nrep=1,progress=TRUE)
```

Arguments

coordx	A numeric $(d \times 2)$-matrix or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.
coordz	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is `NULL`.
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**.
parallel	Logical; if `TRUE` then the TB method is parallelized
ncores	Numeric; number of cores involved in parallelization.
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).
max.ext	Numeric; The maximum extension of the simulation window (for the spatial CE method).
method	String; the type of approximation method. Default is `TB` that is the turning band method. The other possible choice is and `CE` (circular embeeding).
L	Numeric; the number of lines in the turning band method.
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**.
anisopars	A list of two elements "angle" and "ratio" i.e. the anisotropy angle and the anisotropy ratio, respectively.
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)
X	Numeric; Matrix of space-time covariates.
spobj	An object of class sp or spacetime
nrep	Numeric; Numbers of indipendent replicates.
progress	Logic; If TRUE then a progress bar is shown.

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;
`method`	The method of simulation
`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;
`spacetime`	`TRUE` if spatio-temporal and `FALSE` if spatial RF;
`nrep`	The number of indipendent replicates;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

References

T. Gneiting, H. Sevcikova, D. B. Percival, M. Schlather and Y. Jiang (2006) Fast and Exact Simulation of Large Gaussian Lattice Systems in R2: Exploring the Limits *Journal of Computational and Graphical Statistics* 15 (3)

M. Bevilacqua, X. Emery, F. Cuevas Pacheco (2025) Fast simulation of Gaussian random fields with flexible correlation models in Euclidean spaces *arxiv*

Examples

```
library(GeoModels)


################################################################
###
### Example 1. Simulation of a large spatial Gaussian RF
###            with  Matern  covariance model
###            using circulant embeeding method
###            It works only for regular grid
###############################################################
set.seed(68)
x = seq(0,1,0.005)
y = seq(0,1,0.005)
param=list(smooth=1.5,mean=0,sill=1,scale=0.2/3,nugget=0)
# Simulation of a spatial Gaussian RF with Matern correlation function
data1 <- GeoSimapprox(coordx=x,coordy=y, grid=TRUE,corrmodel="Matern", model="Gaussian",
                      method="CE",param=param)$data
fields::image.plot( matrix(data1, length(x), length(y), byrow = TRUE) )

################################################################
###
### Example 2. Simulation of a large spatial Tukey-h RF
###            with  GenWend-Matern  covariance model
###            using spectral Turning band method
###            It works for (ir)regular grid
###############################################################
set.seed(68)
x = runif(50000)
y = runif(50000)
coords=cbind(x,y)
param=list(smooth=0.5,mean=0,sill=1,scale=0.04,nugget=0,tail=0.15,power2=1/4)
# Simulation of a spatial Gaussian RF with Matern correlation function
data1 <- GeoSimapprox(coords, corrmodel="GenWend_Matern", model="Tukeyh",
                      method="TB",param=param)$data
quilt.plot(coords,data1)



################################################################
###
### Example 3. Simulation of a large spacetime Gaussian RF
###            with separable matern  covariance model
###            using  Circular embeeding method
###            It works  for (large) regular time grid
###############################################################
set.seed(68)
coordt <- (0:100)
coords <- cbind( runif(100, 0 ,1), runif(100, 0 ,1))
param <- list(mean  = 0, sill = 1, nugget = 0.25,
              scale_s = 0.05, scale_t = 2,
              smooth_s = 0.5, smooth_t = 0.5)
```

```
# Simulation of a spatial Gaussian RF with Matern correlation function
param<-list(nugget=0,mean=0,scale_s=0.2/3,scale_t=2/3,sill=1,smooth_s=0.5,smooth_t=0.5)

data <- GeoSimapprox(coordx=coords, coordt=coordt, corrmodel="Matern_Matern",
                     model="Gaussian",method="CE",param=param)$data
dim(data)

#################################################################
###
### Example 4. Simulation of a large spacetime Gaussian RF
###            with separable GenWend covariance model
###            using  Circular embeeding method in time
################################################################
set.seed(68)
# Simulation of a spatial Gaussian RF with Matern correlation function
param<-list(nugget=0,mean=0,scale_s=0.2,scale_t=3,sill=1,
             smooth_s=0,smooth_t=0, power2_s=4,power2_t=4)

data <- GeoSimapprox(coordx=coords, coordt=coordt, corrmodel="GenWend_GenWend",
                     model="Gaussian",method="CE",param=param)$data
dim(data)


#################################################################
###
### Example 6. Simulation of a large bivariate Gaussian RF
### with  bivariate Matern correlation model
### using spectral Turning band method
################################################################

# Define the spatial-coordinates of the points:
#x <- runif(20000, 0, 2)
#y <- runif(20000, 0, 2)
#coords <- cbind(x,y)

# Simulation of a bivariate spatial Gaussian RF:
# with a  Bivariate Matern
#set.seed(12)
#param=list(mean_1=4,mean_2=2,smooth_1=0.5,smooth_2=0.5,smooth_12=0.5,
#           scale_1=0.12,scale_2=0.1,scale_12=0.15,
#           sill_1=1,sill_2=1,nugget_1=0,nugget_2=0,pcol=0.5)
#data <- GeoSimapprox(coordx=coords,corrmodel="Bi_matern",
#              param=param,method="TB",L=1000)$data
#opar=par(no.readonly = TRUE)
#par(mfrow=c(1,2))
#quilt.plot(coords,data[1,],col=terrain.colors(100),main="1",xlab="",ylab="")
#quilt.plot(coords,data[2,],col=terrain.colors(100),main="2",xlab="",ylab="")
```

GeoSimcond *Conditional (fast) simulation of Gaussian and non Gaussian Random Fields.*

Description

Simulates spatial (spatio-temporal) Gaussian and non-Gaussian random fields conditioned on observed data using specified correlation models. For large conditioning datasets approximation methods like turning band can be used. For large locations to simlate local kriging can be used.

Usage

```
GeoSimcond(estobj = NULL, data, coordx, coordy = NULL, coordz = NULL, coordt = NULL,
           coordx_dyn = NULL, corrmodel, distance = "Eucl", grid = FALSE, loc,
         maxdist = NULL, maxtime = NULL, method = "Cholesky", model = "Gaussian",
           n = 1, nrep = 1, local = FALSE, L = 1000, neighb = NULL,
           param, anisopars = NULL, radius = 1, sparse = FALSE, time = NULL,
           copula = NULL, X = NULL, Xloc = NULL, Mloc = NULL,
           parallel=FALSE, ncores = NULL, progress=FALSE)
```

Arguments

estobj	Object of class Geofit containing model information
data	Numeric vector/matrix/array of observed data
coordx	Numeric matrix of spatial coordinates (d x 2 or d x 3)
coordy	Optional numeric vector of y-coordinates
coordz	Optional numeric vector of z-coordinates
coordt	Optional numeric vector of temporal coordinates
coordx_dyn	Optional list of dynamic spatial coordinates
corrmodel	String specifying correlation model name
distance	String specifying distance metric (default: "Eucl")
grid	Logical for regular grid (default: FALSE)
loc	Numeric matrix of prediction locations (n x 2)
maxdist	Optional maximum distance for local kriging
maxtime	Optional maximum temporal distance
method	String for decomposition method ("Cholesky", "TB", or "CE")
model	String specifying random field type (default: "Gaussian")
n	Number of trials for binomial RFs (default: 1)
nrep	Number of independent replicates (default: 1)
local	Logical for local kriging (default: FALSE)
L	Number of lines for turning bands method (default: 1000)
neighb	Optional neighborhood order for local kriging
param	List of parameter values
anisopars	List with anisotropy angle and ratio
radius	Radius for spherical coordinates (default: Earth's radius)
sparse	Logical for sparse matrix algorithms (default: FALSE)

`time`	Optional vector of temporal instants to predict
`copula`	Optional string specifying copula type
`X`	Optional matrix of spatio-temporal covariates
`Xloc`	Optional matrix of covariates for prediction locations
`Mloc`	Optional vector of estimated means for prediction locations
`parallel`	If TRUE the the computation is parallelized
`ncores`	Numbers of cores involved in the parallelization
`progress`	If TRUE then a progress bar is shown.

Details

For Gaussian RF, performs conditional simulation using three steps:

1. Unconditional simulation at observed and prediction locations
2. Simple kriging estimates at observed locations
3. Combination to produce conditional simulations

For large datasets, approximate methods ("TB" or "CE") are recommended coupled with local kriging (local=TRUE and neighb=k) and using parallelization (parallel=T).

Value

Returns an object of class `GeoSimcond` containing:

- Simulated field realizations
- Model parameters and settings
- Spatial/temporal coordinates
- Covariance information

Author(s)

Moreno Bevilacqua <moreno.bevilacqua89@gmail.com>,\ Víctor Morales Oñate <victor.morales@uv.cl>,\ Christian Caamaño-Carrillo <chcaaman@ubiobio.cl>

References

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

See Also

`GeoSim`, `GeoKrig`

Examples

```
library(GeoModels)

###############################################
## conditional simulation of a Gaussian rf ###
###############################################
model="Gaussian"
set.seed(79)
### conditioning locations
x = runif(250, 0, 1)
y = runif(250, 0, 1)
coords=cbind(x,y)

# Set the exponential cov parameters:
corrmodel = "GenWend"
mean=0; sill=1; nugget=0
scale=0.2;smooth=0;power2=4

param=list(mean=mean,sill=sill,nugget=nugget,scale=scale,smooth=smooth,power2=power2)

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

## estimation with pairwise likelihood
fixed=list(nugget=nugget,smooth=smooth,power2=power2)
start=list(mean=0,scale=scale,sill=1)
I=Inf
lower=list(mean=-I,scale=0,sill=0)
upper=list(mean= I,scale=I,sill=I)
# Maximum pairwise likelihood fitting :
fit = GeoFit(data, coordx=coords, corrmodel=corrmodel,model=model,
             likelihood='Marginal', type='Pairwise',neighb=3,
             optimizer="nlminb", lower=lower,upper=upper,
             start=start,fixed=fixed)

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

# Conditional simulation
sim_result <- GeoSimcond(fit,loc = loc_to_sim,nrep=50)

cond_mean=sim_result$cond_mean  # conditional mean
cond_var =sim_result$cond_var    # conditional var

par(mfrow=c(1,3))
quilt.plot(coords, data)
quilt.plot(loc_to_sim, cond_mean)
quilt.plot(loc_to_sim, cond_var)
par(mfrow=c(1,1))
```

```
##############################################
## conditional simulation of a LogGaussian rf
##############################################
model="LogGaussian"
set.seed(79)
### conditioning locations
x = runif(500, 0, 1)
y = runif(500, 0, 1)
coords=cbind(x,y)

# Set the exponential cov parameters:
corrmodel = "Matern"
mean=0; sill=.1; nugget=0
scale=0.2;smooth=0.5

param=list(mean=mean,sill=sill,nugget=nugget,scale=scale,smooth=smooth)

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

## estimation with pairwise likelihood
fixed=list(nugget=nugget,smooth=smooth)
start=list(mean=0,scale=scale,sill=1)
I=Inf
lower=list(mean=-I,scale=0,sill=0)
upper=list(mean= I,scale=I,sill=I)
# Maximum pairwise likelihood fitting :
fit = GeoFit(data, coordx=coords, corrmodel=corrmodel,model=model,
             likelihood='Marginal', type='Pairwise',neighb=3,
             optimizer="nlminb", lower=lower,upper=upper,
             start=start,fixed=fixed)

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

# Conditional simulation
sim_result <- GeoSimcond(fit,loc = loc_to_sim,nrep=50)

cond_mean=sim_result$cond_mean  # conditional mean
cond_var =sim_result$cond_var    # conditional var

par(mfrow=c(1,3))
quilt.plot(coords, data)
quilt.plot(loc_to_sim,cond_mean)
quilt.plot(loc_to_sim,cond_var)
par(mfrow=c(1,1))
```

GeoSimCopula *Simulation of Gaussian and non Gaussian Random Fields using copula.*

Description

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

Usage

```
GeoSimCopula(coordx, coordy=NULL,coordz=NULL, coordt=NULL,
coordx_dyn=NULL, corrmodel, distance="Eucl", grid=FALSE,
      method="cholesky", model='Gaussian', n=1, param,
      anisopars=NULL,radius=1, sparse=FALSE,
      copula="Gaussian",X=NULL,spobj=NULL,nrep=1,progress=FALSE)
```

Arguments

coordx	A numeric ($d \times 2$)-matrix or ($d \times 3$)-matrix 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; Optional argument, the default is NULL.
coordz	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is NULL.
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.
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).
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**.

anisopars	A list of two elements "angle" and "ratio" i.e. the anisotropy angle and the anisotropy ratio, respectively.
radius	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is 1.
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.
copula	String; the type of copula. It can be "Clayton" or "Gaussian"
X	Numeric; Matrix of space-time covariates.
spobj	An object of class sp or spacetime
nrep	Numeric; Numbers of indipendent replicates.
progress	Logic; If TRUE then a progress bar is shown.

Value

Returns an object of class GeoSimCopula. An object of class GeoSimCopula 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;
method	The method of simulation
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	A list of the parameters
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;
copula	The type of copula

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

References

Bevilacqua M., Alvarado E., Caamano C. (2024) A flexible Clayton-like spatial copula with application to bounded support data. *Journal of Multivariate Analysis* **201**

Examples

```
library(GeoModels)

################################################################
###
### Example:  Simulation of a reparametrized Beta RF
### for beta regression
### with Gaussian and Clayton Copula
### with  underlying Wendland correlation.
###
###############################################################
set.seed(261)
NN=1400
x <- runif(NN);y <- runif(NN)
coords=cbind(x,y)

corrmodel="GenWend"
X=cbind(rep(1,NN),runif(NN))

NuisParam("Beta2",num_betas=2,copula="Gaussian")
CorrParam("GenWend")
#### Gaussian copula
param=list(smooth=0,power2=4, min=0,max=1,
             mean=0.1,mean1=0.1,scale=0.3,nugget=0,shape=5)

data <- GeoSimCopula(coordx=coords, corrmodel=corrmodel, model="Beta2",param=param,
  copula="Gaussian",sparse=TRUE,X=X)$data

quilt.plot(coords,data)


#### Clayton copula
NuisParam("Beta2",num_betas=2,copula="Clayton")
CorrParam("GenWend")
param=list(smooth=0,power2=4, min=0,max=1,
             mean=0.2,mean1=0.1,scale=0.3,nugget=0,shape=6,nu=4)
data1 <- GeoSimCopula(coordx=coords, corrmodel=corrmodel, model="Beta2",param=param,
  copula="Clayton",sparse=TRUE,X=X)$data

hist(data1,freq=FALSE)
quilt.plot(coords,data1)
```

GeoTestIsotropy *Parametric bootstrap test for isotropy*

Description

Performs an isotropy test based on *parametric bootstrap* by comparing an isotropic model (H0) with an anisotropic model (H1). A user-chosen spatial correlation model is fitted to spatial data, and the test statistic is a likelihood ratio between H1 (with anisotropy estimated) and H0 (anisotropy fixed at ratio = 1 and angle = 0). The *p*-value is computed via parametric bootstrap.

Usage

```
GeoTestIsotropy(data, coordx,
                start, fixed,
                optimizer = "bobyqa",
                model = "Gaussian",
                corrmodel = "Matern",
                lower = NULL, upper = NULL,
                B = 1000,
                likelihood = NULL,
                type = NULL,
                copula = NULL,
                neighb = 5,
                parallel = TRUE,
                ncores = NULL,
                progress = TRUE)
```

Arguments

`data`	Numeric vector of observations (length `n`).
`coordx`	`n x d` matrix with the spatial coordinates of the sites (at least 2 columns for anisotropy analysis).
`start`	*Named* list with the parameters **to be estimated**; names must match those expected by `CorrParam(corrmodel)` and `NuisParam(model)`.
`fixed`	*Named* list with the parameters **kept fixed** during estimation; together with `start` it must exactly cover the required set of parameters.
`optimizer`	Optimization algorithm to be used in `GeoFit` (e.g., `"bobyqa"`).
`model`	Marginal model for the data (e.g., `"Gaussian"`).
`corrmodel`	Spatial correlation model (e.g., `"Matern"`).
`lower, upper`	*Named* lists with lower/upper bounds for parameters in `start` only. If `NULL`, no extra bounds beyond defaults.
`B`	Number of parametric bootstrap replications (default `1000`).
`likelihood`	Type of (composite) likelihood to pass to `GeoFit` (e.g., `"Marginal"`).
`type`	Composite-likelihood type (e.g., `"Pairwise"`).
`copula`	Copula object for simulation via `GeoSimCopula`; if `NULL` Gaussian simulation is used.
`neighb`	Number of neighbors for composite-likelihood estimation, where applicable.
`parallel`	If `TRUE`, bootstrap fits are distributed in parallel through **future**.

ncores	Number of cores to use in parallel; if NULL it is chosen automatically up to B or parallel::detectCores()-1.
progress	If TRUE, show progress bars using **progressr**.

Details

GeoTestIsotropy implements a *likelihood ratio test* (LRT) between:

- **H0**: isotropic model with anisotropy parameters fixed at angle = 0, ratio = 1;
- **H1**: anisotropic model with angle and ratio estimated.

The observed statistic is $\mathrm{LRT}_{\mathrm{obs}} = 2(\ell_1 - \ell_0)$, truncated at zero if negative. The null distribution is approximated by simulating B datasets under H0 with parameters equal to the H0 fit, and recomputing the LRT. The p-value is computed as $(1 + \#\{\mathrm{LRT}_b \geq \mathrm{LRT}_{\mathrm{obs}}\})/(1 + B_{\mathrm{valid}})$. The function includes robust checks on parameter names, bounds, simulation method choice (Cholesky vs TB), and parallelization.

Value

An (invisible) list with components:

statistic	Observed LRT statistic.
pvalue	Bootstrap p-value.
ratio_hat	Estimated anisotropy ratio (> 1 indicates stretching along the principal axis).
angle_hat	Estimated anisotropy angle (radians/degrees according to GeoFit convention).
corrmodel	Correlation model used.
model	Marginal model used.
fit_H0	GeoFit object fitted under H0.
fit_H1	GeoFit object fitted under H1.
B_rep	Vector of LRT values obtained on the valid bootstrap replications.

Author(s)

Moreno Bevilacqua <moreno.bevilacqua89@gmail.com> https://sites.google.com/view/moreno-bevilacqua/home
Víctor Morales Oñate <victor.morales@uv.cl> https://sites.google.com/site/moralesonatevictor/
Christian Caamaño-Carrillo <chcaaman@ubiobio.cl> https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit, GeoSim, GeoSimapprox, GeoVarestbootstrap

`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' statistis 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^2_i 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) we refere to the article (see **References**). This versions can be called from the routine assigning respectively the values: `WilksCB`.

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.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

References

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

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. Test on the parameter
### of a regression model using conditional composite likelihood
###
###############################################################
set.seed(342)
model="Gaussian"
# Define the spatial-coordinates of the points:
NN=1500
x = runif(NN, 0, 1)
y = runif(NN, 0, 1)
coords = cbind(x,y)
# Parameters
mean=1; mean1=-1.25;  # regression parameters
nugget=0; sill=1

# matrix covariates
X=cbind(rep(1,nrow(coords)),runif(nrow(coords)))

# model correlation
corrmodel="Wend0"
power2=4;c_supp=0.15

# simulation
param=list(power2=power2,mean=mean,mean1=mean1,
              sill=sill,scale=c_supp,nugget=nugget)
data = GeoSim(coordx=coords, corrmodel=corrmodel,model=model, param=param,X=X)$data

I=Inf
##### H1: (regressian mean)
fixed=list(nugget=nugget,power2=power2)
start=list(mean=mean,mean1=mean1,scale=c_supp,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:
fitH1 = GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
```

```
              likelihood="Conditional",type="Pairwise",sensitivity=TRUE,
                   lower=lower,upper=upper,neighb=3,
                   optimizer="nlminb",X=X,
                    start=start,fixed=fixed)

unlist(fitH1$param)

##### H0: (constant mean i.e beta1=0)
fixed=list(power2=power2,nugget=nugget,mean1=0)
start=list(mean=mean,scale=c_supp,sill=sill)
lower0=list(mean=-I,scale=0,sill=0)
upper0=list(mean=I,scale=I,sill=I)
# Maximum pairwise composite-likelihood fitting of the RF:
fitH0 = GeoFit(data=data,coordx=coords,corrmodel=corrmodel, model=model,
            likelihood="Conditional",type="Pairwise",sensitivity=TRUE,
                      lower=lower0,upper=upper0,neighb=3,
                   optimizer="nlminb",X=X,
                    start=start,fixed=fixed)
unlist(fitH0$param)

# not run
##fitH1=GeoVarestbootstrap(fitH1,K=100,optimizer="nlminb",
##                       lower=lower, upper=upper)
##fitH0=GeoVarestbootstrap(fitH0,K=100,optimizer="nlminb",
##                       lower=lower0, upper=upper0)

#  Composite likelihood Wald and ratio statistic statistic tests
#  rejecting null  hypothesis beta1=0
##GeoTests(fitH1, fitH0 ,statistic='Wald')
##GeoTests(fitH1, fitH0 , statistic='WilksS')
##GeoTests(fitH1, fitH0 , statistic='WilksCB')




#################################################################
###
### Example 2. Parametric test of Gaussianity
### using SinhAsinh random fields using standard likelihood
###
################################################################
set.seed(99)
model="SinhAsinh"
# Define the spatial-coordinates of the points:
NN=200
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 from Gaussian  model (H0)
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: SinhAsinh model
fixed=list(power2=power2,nugget=nugget,mean=mean)
start=list(scale=c_supp,skew=skew,tail=tail,sill=sill)

lower=list(scale=0,skew=-I, tail=0,sill=0)
upper=list(scale=I,skew= I,tail=I,sill=I)
# Maximum pairwise composite-likelihood fitting of the RF:
fitH1 = GeoFit2(data=data,coordx=coords,corrmodel=corrmodel, model=model,
                   likelihood="Full",type="Standard",varest=TRUE,
                   lower=lower,upper=upper,
                   optimizer="nlminb",
                    start=start,fixed=fixed)

unlist(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)

unlist(fitH0$param)

#  Standard likelihood Wald and ratio statistic statistic tests
#  not rejecting null  hypothesis tail=1,skew=0 (Gaussianity)
GeoTests(fitH1, fitH0,statistic='Wald')
GeoTests(fitH1, fitH0,statistic='Wilks')
```

GeoTestsupp_space *Parametric bootstrap test for spatial support (scale)*

Description

Performs a parametric bootstrap likelihood-ratio test on the spatial *scale* parameter using the compactly supported GenWend correlation model. Two hypotheses are compared: **H0** imposes a maximum admissible scale (or independence), while **H1** estimates scale freely. The test statistic is the likelihood ratio and the *p*-value is obtained by parametric bootstrap under **H0**.

Usage

```
GeoTestsupp_space(data, coordx,
                  start, fixed,
                  model = "Gaussian",
                  h0 = NULL,
                  optimizer = "bobyqa",
                  lower = NULL, upper = NULL,
                  neighb = 5,
                  B = 1000,
                  likelihood = NULL,
                  type = NULL,
                  method = "Cholesky",
                  parallel = TRUE,
                  ncores = NULL,
                  progress = TRUE)
```

Arguments

`data`	Numeric vector of observations of length `n`.
`coordx`	`n x d` matrix with spatial coordinates (at least 2 columns).
`start`	*Named* list of parameters **to be estimated**. Names must match `CorrParam(corrmodel)` and `NuisParam(model)`.
`fixed`	*Named* list of parameters **kept fixed**. Together with `start` it must exactly cover the required set.
`model`	Marginal model for the data (default `"Gaussian"`).
`h0`	Positive threshold for the *scale* parameter under **H0**. If `NULL` (default), **H0** represents *spatial independence* with `scale = h0_val` set to the (adjusted) minimum inter-point distance.
`optimizer`	Optimization method passed to `GeoFit` (e.g., `"bobyqa"`).
`lower, upper`	*Named* lists with bounds for parameters in `start` only. `NULL` means no extra bounds beyond defaults. Bounds are internally widened if too narrow.
`neighb`	Number of neighbors for composite likelihood when applicable. Ignored when the full likelihood is used.
`B`	Number of parametric bootstrap replications (default `1000`).
`likelihood`	Type of (composite) likelihood to pass to `GeoFit` (e.g., `"Marginal"`).
`type`	Composite-likelihood type (e.g., `"Pairwise"`).
`method`	Simulation method for bootstrap under **H0**: `"Cholesky"` or `"TB"` (turning bands via `GeoSimapprox`).
`parallel`	If `TRUE`, bootstrap estimations are parallelized using **future**.
`ncores`	Number of cores for parallelization. If `NULL`, it is selected automatically up to `B` or `parallel::detectCores()-1`.
`progress`	If `TRUE`, progress bars are shown via **progressr**.

Details

The function fixes the correlation model to `corrmodel = "GenWend"` (generalized Wendland, compact support). Input validation ensures consistency of parameter names between `start` and `fixed` w.r.t. `CorrParam(corrmodel)` and `NuisParam(model)`.

Automatic likelihood selection. If $n > 10{,}000$, a composite-likelihood is used (`likelihood = "Marginal"`, `type = "Pairwise"`) with `neighb` neighbors; otherwise the full likelihood is used (`likelihood = "Full"`, `type = "Standard"`).

Threshold `h0` and minimum distance. Let d_{min} be the minimum inter-point distance, computed with **nabor**. If `h0 = NULL`, the test defaults to independence with `h0\_val = max(1e-6, d_min - 1e-6)`. Under **H0** one enforces `scale <= h0` (or `scale = h0\_val` for independence), while under **H1** the scale is estimated freely but capped above by a large, data-driven `U\_big`.

Likelihood-ratio statistic. Let $\Lambda_{obs} = \max\{0, 2(\ell_1 - \ell_0)\}$ where ℓ_0 and ℓ_1 are the log-(composite) likelihoods under **H0** and **H1**. The null distribution of Λ is approximated via parametric bootstrap with `B` datasets simulated under **H0**; the *p*-value is computed as

$$(1 + \#\{\Lambda_b \geq \Lambda_{obs}\})/(1 + B_{valid}).$$

Early exit. If the unconstrained fit under **H1** already satisfies `scale <= h0` (or `<= h0\_val`), the function returns `lambda\_obs = 0` and `pvalue = 1` without running the bootstrap.

Robust bounds. The helper automatically widens degenerate or ultra-narrow boxes and enforces positivity for `sill` and `scale`.

Value

An (invisible) list with components:

`d_min`	Minimum inter-point distance d_{min}.
`h0`	Threshold actually used under **H0**.
`lambda_obs`	Observed likelihood-ratio statistic Λ_{obs}.
`pvalue`	Bootstrap *p*-value.
`B_rep`	Vector of bootstrap statistics Λ_b for valid replications.
`fit_H0`	`GeoFit` object fitted under **H0**.
`fit_H1`	`GeoFit` object fitted under **H1**.

Note

The function uses `corrmodel = "GenWend"` internally. For very large `n`, `method = "TB"` may be preferable for speed.

Author(s)

Moreno Bevilacqua <moreno.bevilacqua89@gmail.com> https://sites.google.com/view/moreno-bevilacqua/home
Víctor Morales Oñate <victor.morales@uv.cl> https://sites.google.com/site/moralesonatevictor/
Christian Caamaño-Carrillo <chcaaman@ubiobio.cl> https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit, GeoSim, GeoSimapprox, GeoVarestbootstrap

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

Description

The procedure update a GeoFit object computing stderr estimation, confidence intervals and p-values using parametric bootstrap.

Usage

```
GeoVarestbootstrap(fit,K=100,sparse=FALSE,
  optimizer=NULL, lower=NULL, upper=NULL,
  method="cholesky",alpha=0.95, L=1000,parallel=TRUE,ncores=NULL,progress=TRUE)
```

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).
optimizer	The type of optimization algorithm (see GeoFit for details). If NULL then the optimization algorithm of the object fit is chosen.
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.
method	String; The method of simulation. Default is cholesky. For large data set three options are TB or CE (see the GeoSimapprox) function.
alpha	Numeric; The level of the confidence interval.
L	Numeric; the number of lines in the turning band method.
parallel	Logical; if TRUE then the estimation step is parallelized
ncores	Numeric; number of cores involved in parallelization.
progress	Logic; If TRUE then a progress bar is shown.

Details

The function update a GeoFit object estimating stderr estimation and confidence interval using parametric bootstrap.

Value

Returns an (updated) object of class GeoFit.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit.

Examples

```
library(GeoModels)

################################################################
###
### Example 1. Test on the parameter
### of a regression model using conditional composite likelihood
###
###############################################################
set.seed(342)
model="Gaussian"
# Define the spatial-coordinates of the points:
NN=3500
x = runif(NN, 0, 1)
y = runif(NN, 0, 1)
coords = cbind(x,y)
# Parameters
mean=1; mean1=-1.25;  # regression parameters
 sill=1 # variance

# matrix covariates
X=cbind(rep(1,nrow(coords)),runif(nrow(coords)))

# model correlation
corrmodel="Matern"
smooth=0.5;scale=0.1; nugget=0;

# simulation
param=list(smooth=smooth,mean=mean,mean1=mean1,
              sill=sill,scale=scale,nugget=nugget)
data = GeoSim(coordx=coords, corrmodel=corrmodel,
                model=model, param=param,X=X)$data

I=Inf

fixed=list(nugget=nugget,smooth=smooth)
```

```
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,
              likelihood="Conditional",type="Pairwise",sensitivity=TRUE,
                   lower=lower,upper=upper,neighb=3,
                   optimizer="nlminb",X=X,
                    start=start,fixed=fixed)

unlist(fit$param)


#fit_update=GeoVarestbootstrap(fit,K=100,parallel=TRUE)
#fit_update$stderr
#fit_update$conf.int
#fit_update$pvalues
```

GeoVariogram *Empirical semivariogram estimation*

Description

Computes an empirical estimate of the semivariogram for spatial, spatio-temporal, and bivariate random fields.

Usage

```
GeoVariogram(data, coordx, coordy=NULL, coordz=NULL, coordt=NULL,
              coordx_dyn=NULL, cloud=FALSE, distance="Eucl",
              grid=FALSE, maxdist=NULL, neighb=NULL,
              maxtime=NULL, numbins=NULL,
              radius=1, type='variogram', bivariate=FALSE,
              subsample=1, subsample_t=1)
```

Arguments

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

coordx	Spatial coordinates. Either a numeric vector giving the first coordinate, or a $d \times 2$ (or $d \times 3$) matrix of coordinates. If distance refers to great-circle distances, coordinates must be provided in lon/lat format (decimal degrees) and the sphere radius is set by radius.
coordy	A numeric vector giving the second spatial coordinate. Optional, default is NULL.
coordz	A numeric vector giving the third spatial coordinate (if needed). Optional, default is NULL.
coordt	A numeric vector of temporal coordinates. If NULL (default), a purely spatial random field is assumed.
coordx_dyn	A list of m numeric matrices $d_t \times 2$ providing time-varying spatial coordinates (dynamic locations). Optional, default is NULL.
cloud	Logical; if TRUE the semivariogram cloud is computed. If FALSE (default), a binned empirical semivariogram is returned.
distance	String specifying the spatial distance. Default is "Eucl" (Euclidean distance). See the **Details** section of GeoFit.
grid	Logical; if FALSE (default) data are interpreted as observations on irregularly spaced locations. If TRUE, data are interpreted as observations on a regular grid.
maxdist	Numeric; maximum spatial distance to be considered in semivariogram estimation. See **Details**.
neighb	Numeric; an optional positive integer indicating the order of neighborhood (useful for large datasets). See **Details**.
maxtime	Numeric; maximum temporal lag to be considered for spatio-temporal semivariograms. See **Details**.
numbins	Numeric; number of distance bins used to compute the binned semivariogram. See **Details**.
radius	Numeric; radius of the sphere when using great-circle distances. Default is 1.
type	String; type of semivariogram. Currently available: "variogram".
bivariate	Logical; if FALSE (default) data are interpreted as univariate spatial/spatio-temporal realisations. If TRUE, data is interpreted as a realisation from a bivariate field and (cross-)semivariograms are computed.
subsample	Numeric in $(0, 1]$. Proportion of spatial locations to be used to compute the semivariogram (useful for large datasets). Default is 1 (use all locations).
subsample_t	Numeric in $(0, 1]$. Proportion of time points to be used in spatio-temporal settings (when coordt is provided). Default is 1 (use all time points).

Details

We report the definition of the semivariogram in the spatial case; extensions to spatio-temporal and bivariate settings are based on the same principles.

For a spatial random field $Z(\cdot)$, the (classical) binned semivariogram estimator is defined as

$$\hat{\gamma}(h) = \frac{1}{2|N(h)|} \sum_{(x_i, x_j) \in N(h)} \{Z(x_i) - Z(x_j)\}^2,$$

where $N(h)$ is the set of all sample pairs whose spatial distance falls within a tolerance region around lag h (equally spaced intervals are used when cloud=FALSE).

The numbins argument sets the number of spatial lag bins used when cloud=FALSE.

The maxdist argument sets the maximum spatial distance considered in the estimation.

The maxdist option can be combined with neighb to reduce the number of pairs when handling large datasets, by restricting computations to local neighborhoods.

The maxtime argument sets the maximum temporal lag considered for spatio-temporal semivariograms.

The subsample and subsample_t arguments provide additional control for large datasets by using only a proportion of spatial locations and/or time points.

Value

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

bins	Spatial distance bins if cloud=FALSE. If cloud=TRUE, all spatial pairwise distances.
bint	Temporal distance bins if cloud=FALSE. If cloud=TRUE, all temporal pairwise distances.
cloud	Logical; TRUE if the variogram cloud is returned, FALSE otherwise.
centers	Centers of the spatial bins.
distance	Type of spatial distance.
lenbins	Number of pairs in each spatial bin.
lenbinst	Number of pairs in each spatio-temporal bin.
lenbint	Number of pairs in each temporal bin.
maxdist	Maximum spatial distance used in the estimation; NULL if not specified.
maxtime	Maximum temporal lag used in the estimation; NULL if not specified.
spacetime_dyn	Logical; TRUE if dynamic coordinates (coordx_dyn) are used.
variograms	Empirical spatial semivariogram.
variogramst	Empirical spatio-temporal semivariogram.
variogramt	Empirical temporal semivariogram.
type	Type of estimated semivariogram.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>, https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian Caamaño-Carrillo, <chcaaman@ubiobio.cl>, https://www.researchgate.net/profile/Christian-Caamano

References

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

Gaetan, C. and Guyon, X. (2010) *Spatial Statistics and Modeling*. Springer-Verlag, New York.

See Also

GeoFit

Examples

```
library(GeoModels)

################################################################
### Example 1. Empirical semivariogram from a spatial Gaussian
### random field with Matérn correlation.
################################################################
set.seed(514)
x = runif(200, 0, 1)
y = runif(200, 0, 1)
coords = cbind(x,y)

corrmodel = "Matern"
mean = 0
sill = 1
nugget = 0
scale = 0.3/3
smooth = 0.5

data = GeoSim(coordx=coords, corrmodel=corrmodel,
              param=list(mean=mean, smooth=smooth, sill=sill,
                         nugget=nugget, scale=scale))$data

vario = GeoVariogram(coordx=coords, data=data, maxdist=0.6)
plot(vario, pch=20, ylim=c(0,1), ylab="Semivariogram", xlab="Distance")

################################################################
### Example 2. Empirical semivariogram for a spatio-temporal
### Gaussian random field with Gneiting correlation.
################################################################
set.seed(331)
x = runif(200, 0, 1)
y = runif(200, 0, 1)
coords = cbind(x,y)
times = seq(1,10,1)

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

vario_st = GeoVariogram(data=data, coordx=coords, coordt=times,
                        maxtime=7, maxdist=0.5)
plot(vario_st, pch=20)

################################################################
### Example 3. Empirical (cross-)semivariograms for a bivariate
### Gaussian random field with Bi-Matérn covariance.
################################################################
```

```
set.seed(293)
x = runif(400, 0, 1)
y = runif(400, 0, 1)
coords = cbind(x,y)

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
biv_vario = GeoVariogram(data, coordx=coords, bivariate=TRUE, maxdist=0.5)
plot(biv_vario, pch=20)
```

GeoVariogramDir *Empirical directional semivariogram*

Description

Computes empirical semivariograms in multiple directions (e.g., 0, 45, 90, 135 degrees) to assess spatial anisotropy.

Usage

```
GeoVariogramDir(data, coordx, coordy = NULL, coordz = NULL,
                directions = c(0, 45, 90, 135), tolerance = 22.5, numbins = 13,
                maxdist = NULL, neighb = NULL, distance = "Eucl",
                subsample = 1)
```

Arguments

data	A numeric vector containing the observed values at each location.
coordx	Spatial coordinates. Either a numeric vector giving the first coordinate, or a matrix with 2 (or 3) columns. If a matrix is provided, `coordy` and `coordz` are ignored.
coordy	A numeric vector of the second coordinate. Optional; defaults to `NULL`.
coordz	A numeric vector of the third coordinate (if needed). Optional; defaults to `NULL`.
directions	Numeric vector giving the principal directions (in degrees) for which the semivariogram is computed (default: `c(0, 45, 90, 135)`).
tolerance	Angular tolerance (in degrees) around each direction (default: 22.5).
numbins	Number of distance bins for the empirical semivariogram (default: 13).
maxdist	Maximum spatial distance to consider between pairs. If `NULL`, an internal default is used.

neighb	Number of nearest neighbors to use for each location. If NULL, an internal default is used.
distance	Type of distance metric to use (default: "Eucl"). See GeoFit for options.
subsample	Numeric in $(0, 1]$. Proportion of spatial locations used to compute the directional semivariograms (useful for large datasets). Default is 1 (use all locations).

Details

The function computes empirical semivariograms for several directions by:

- Selecting pairs of points within maxdist and among the neighb nearest neighbors using GeoNeighIndex.
- Computing squared differences for each selected pair.
- Assigning each pair to a directional class if the vector connecting the pair falls within the specified angular tolerance around a given direction.
- Binning pairs by distance and computing the average squared difference (semivariogram) within each bin.

The direction is defined in the *xy-plane* even in 3D. For 2D data, set coordz = NULL.

The subsample argument can be used to reduce computational cost by randomly selecting a subset of spatial locations prior to pair construction. This is particularly useful for large datasets.

Value

A list of class "GeoVariogramDir" with one element for each direction. Each element is a list with components:

centers	Centers of the distance bins.
gamma	Empirical semivariogram values for each bin.
npairs	Number of point pairs in each bin.

See Also

GeoVariogram, GeoNeighIndex

Examples

```
require(GeoModels)
set.seed(960)
NN <- 2500
coords <- cbind(runif(NN), runif(NN))
scale <- 0.5/3
param <- list(mean = 0, sill = 1, nugget = 0, scale = scale, smooth = 0.5)
corrmodel <- "Matern"

set.seed(951)
data <- GeoSim(coordx = coords, corrmodel = corrmodel,
               model = "Gaussian", param = param)$data
```

```
vario_dir <- GeoVariogramDir(data = data, coordx = coords, maxdist = 0.4,
                             subsample = 0.5)

plot(vario_dir, ylim = c(0,1))
```

GeoWLS *WLS of Random Fields*

Description

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

Usage

```
GeoWLS(data, coordx, coordy=NULL,coordz=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=1, start=NULL, weighted=FALSE,optimization=TRUE)
```

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 or $(d \times 3)$-matrix 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; Optional argument, the default is NULL.
coordz	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is NULL.
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 1.
`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.
`optimization`	Logical; if `TRUE` then the weighted least square minimization is performed. Otherwise the weighted least square function is evaluated at the starting value.

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.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

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)
```

`getInvC` *Utility function for kriging computation*

Description

Utility function that computes ($\Sigma^{-1}\mathbf{CC}$) where Σ is a covariance matrix and $\mathbf{CC}$ is the matrix of of covariances between the observed location and the locations to predict. Optionally, the quadratic form $\mathbf{cc}^\top\Sigma^{-1}\mathbf{cc}$ can be computed. Both dense and sparse matrix representations are supported.

Usage

```
getInvC(covmatrix, CC, mse = TRUE)
```

Arguments

`covmatrix` A covariance matrix:

- `covmatrix` A covariance matrix (dense or sparse).
- `sparse` Logical scalar indicating whether the matrix is stored in sparse format.

`CC` Numeric matrix to be multiplied by the inverse of the covariance matrix (length must match the dimension of `covmatrix`).

`mse` Logical. If `TRUE` (default) the quadratic form $\mathbf{cc}^\top\Sigma^{-1}\mathbf{cc}$ is also returned.

Details

For dense matrices the function uses the Cholesky decomposition provided by `FastGP::rcppeigen_get_chol`. For sparse matrices (class `spam`) the factorisation is performed with `spam::chol.spam`. If the covariance matrix is not positive definite an error is thrown.

Value

A list with components

- `a` Numeric vector: the product $\Sigma^{-1}\mathbf{cc}$.
- `b` Numeric scalar: the quadratic form $\mathbf{cc}^\top\Sigma^{-1}\mathbf{cc}$ (only if `mse = TRUE`).

Author(s)

Moreno Bevilacqua, `<moreno.bevilacqua@uai.cl>`

`Lik` *Optimizes the Log Likelihood*

Description

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

Usage

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

Arguments

`copula`	String; the type of copula. It can be "Beta" or "Gaussian"
`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 or $(d \times 3)$-matrix 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; Optional argument, the default is `NULL`.
`coordz`	A numeric vector giving 1-dimension of spatial coordinates; Optional argument, the default is `NULL`.

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 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.
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.
neighb	Numeric;parameter for vecchia approximation using GPvecchia package
MM	Numeric;a non constant fixed mean
aniso	Logical; should anisotropy be considered?
score	Logical; should score function be computed?

Value

Return a list from an optim call.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit

---

madagascartemp   *July Average Temperature of Madagascar*

Description

A 2500×3 matrix containing UTM coordinates and July average temperatures at 2500 location sites in Madagascar, averaged over the period 1970–2000.\ Data obtained using the **Geodata** package with the function worldclim_country.

Usage

```
data(madagascartemp)
```

Format

A numerical matrix of dimension 2500×3.

Source

Fick, S.E. and Hijmans, R.J. (2017).\ WorldClim 2: new 1 km spatial resolution climate surfaces for global land areas.\ *International Journal of Climatology*, **37**(12), 4302–4315.

MargParam *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

```
MargParam(model, bivariate=FALSE,num_betas=c(1,1),copula=NULL)
```

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)
copula	The type of copula.

Details

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

Value

Return a vector string of nuisance parameters.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit

Examples

```
library(GeoModels)

MargParam("Gaussian")

MargParam("Binomial")

MargParam("Weibull",num_betas=2)

MargParam("SkewGaussian", num_betas=3)

MargParam("SinhAsinh")

MargParam("Beta2",copula="Clayton")

MargParam("StudentT")
## note that in the bivariate case sill_1 e sill_2 are considered as correlation parameteres
MargParam("Gaussian", bivariate=TRUE)
```

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.

Value

Return a matrix decomposition

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

---

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(mtx)
MatLogDet(mat.decomp,method)
```

Arguments

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

Value

The function returna a square root or inverse or log determinant of a (semi)positive definite matrix using the function in the FastGP package.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

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

NuisParam *Lists the Nuisance Parameters of a Random Field*

Description

Internal function handling Nuisance Parameters of a Random Field.

Usage

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

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)
copula	The type of copula.

Details

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

Value

Return a vector string of nuisance parameters.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit

NuisParam2 *Internal function handling Nuisance Parameters of a Random Field*

Description

Internal function handling Nuisance Parameters of a Random Field.

Usage

```
NuisParam2(model, bivariate=FALSE,num_betas=c(1,1),copula=NULL)
```

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)
copula	The type of copula.

Details

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

Value

Return a vector string of nuisance parameters.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit

plot.GeoCorrFct — *Plot Spatial and Spatio-temporal correlation or covariance of (non) Gaussian random fields*

Description

Plot Spatial and Spatio-temporal correlation or covariance of (non) Gaussian random fields for a given set of spatial or spatiotemporal distances `GeoCorrFct`.

Usage

```
## S3 method for class 'GeoCorrFct'
plot(x,type="p", ...)
```

Arguments

`x`	an object of the class `"GeoCorrFct"`
`type`	The type of graphic. The possible options are "p" and "l". If "p" then a point type graphic is displayed. Otherwise a lines type graphic displayed.
`...`	Other graphical options arguments. `plot`

Details

Plot Spatial and Spatio-temporal correlation or covariance of (non) Gaussian random fields

Value

Produces a plot. No values are returned.

See Also

`GeoCorrFct` for examples.

plot.GeoVariogram — *Plot empirical spatial, spatio-temporal and spatial bivariate semi-Variogram*

Description

Plot empirical spatial, spatio-temporal and spatial bivariate semi-Variogram using on object `GeoVariogram`.

Usage

```
## S3 method for class 'GeoVariogram'
plot(x, ...)
```

Arguments

`x`	an object of the class "GeoVariogram"
`...`	other arguments to be passed to the function `plot`

Details

This function plots empirical semi variogram in the spatial, spatio-temporal and spatial bivariate case

Value

Produces a plot. No values are returned.

See Also

`GeoVariogram` for variogram computation and examples.

`plot.GeoVariogramDir` *Plot empirical directional semi-variogram*

Description

Plots empirical directional semi-variograms for objects of class "GeoVariogramDir" as produced by `GeoVariogramDir`. All directions are displayed in a single plot, each with a different color and a legend indicating the direction (e.g., "0°", "45°", etc.).

Usage

```
## S3 method for class 'GeoVariogramDir'
plot(x, ..., main = "Directional Empirical Semivariograms",
  pch = 20, lwd = 1, col = 1:8, ylab = "Semivariogram", xlab = "Distance")
```

Arguments

`x`	An object of class "GeoVariogramDir" as produced by `GeoVariogramDir`.
`main`	A main title for the plot.
`pch`	Plotting character (point type) for the points (default: 20).
`lwd`	Line width for the lines connecting points (default: 1).
`col`	A vector of colors, one for each direction (default: 1:8).
`ylab`	Label for the y-axis (default: "Semivariogram").
`xlab`	Label for the x-axis (default: "Lag").
`...`	Additional graphical parameters passed to `plot`.

Details

This function plots all empirical directional semi-variograms in a single graph, using different colors and a legend in the top left corner that indicates the direction (e.g., "0°", "45°", etc.). Each direction is represented by points connected by lines.

Value

Produces a plot. No values are returned.

See Also

`GeoVariogramDir` for directional variogram computation and examples.

`rainNLD` *April Precipitation over the Netherlands with Distance-to-Coast Covariate*

Description

A numeric matrix containing gridded April total precipitation (in mm) over the Netherlands, together with projected coordinates (UTM, km) and a distance-to-coast covariate (km). The precipitation field is derived from the WorldClim v2.1 monthly climatology for April (i.e., long-term monthly averages rather than a specific year).

Usage

```
data(rainNLD)
```

Format

A numeric matrix with 71,401 rows and 4 columns:

Column 1 Easting coordinate (UTM zone 31N), in kilometers.

Column 2 Northing coordinate (UTM zone 31N), in kilometers.

Column 3 Distance to the coastline, in kilometers (computed from Natural Earth coastline geometry, transformed to UTM).

Column 4 April total precipitation, in millimeters.

Details

Coordinates were projected from longitude/latitude (WGS84) to UTM zone 31N and rescaled to kilometers. The distance-to-coast covariate was computed as the Euclidean distance (in UTM) from each grid cell to the nearest coastline.

Source

Fick, S.E., Hijmans, R.J. (2017) WorldClim 2: new 1km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*, **37**, 4302–4315.

`SimCE` *Circulant embeeding simulation*

Description

Subroutine called by GeoSimapprox. The procedure return a simulation on a regular grid from a standard spatial Gaussian random field with a specified correlation model

Usage

```
SimCE(M,N,x,y,z,corrmodel,param,mean.val, max.ext)
```

Arguments

`M`	Numeric; the dimension of x
`N`	Numeric; the dimension of y
`x`	A numeric M-dimensional vector giving 1-dimension of spatial coordinates.
`y`	A numeric N-dimensional vector giving 1-dimension of spatial coordinates.
`z`	A numeric N-dimensional vector giving 1-dimension of spatial coordinates.
`corrmodel`	String; the name of a correlation model.
`param`	A list of parameter values required in the simulation procedure.
`mean.val`	The mean of the random field.
`max.ext`	The maximum extension of the simulation window.

Value

Return a list from an `optim` call.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

See Also

`GeoSimapprox`

sp2Geo *Extracting information from an sp or spacetime object*

Description

Extracting information from an sp or spacetime object

Usage

```
sp2Geo(spobj,spdata = NULL)
```

Arguments

spobj	An object of class sp or spacetime
spdata	Character: The name of data in the sp or spacetime object

Details

The function accepts as input a sp or spacetime object and the name of the data of interest in the object and it returns a list with some useful informations for Geomodels functions.

Value

A list with spatio-temporal informations

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>,https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian", Caamaño-Carrillo, <chcaaman@ubiobio.cl>,https://www.researchgate.net/profile/Christian-Caamano

Examples

```
# Define the spatial-coordinates of the points:
set.seed(3)
N <- 30  # 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 and regression parameters
X <- cbind(rep(1,N),runif(N))
# Define spatial matrix dependent variable
Y <- rnorm(nrow(X))

obj1 <- sp::SpatialPoints(coords)
obj2 <- sp::SpatialPointsDataFrame(coords,data = data.frame(X,Y))
```

```
# sp2Geo info extraction
b <- sp2Geo(obj2,spdata  = "Y")
class(b)
b
```

---

spanish_wind *August monthly average wind speed in Spain between 1970-2000*

Description

A $(6000x3)$-matrix containing lon/lat and august monthly average wind speed (2 m above the ground, meter/second) registered at 6000 location sites in the Iberian peninsula. Data obtained from WorldClim version 2.1

Usage

```
data(spanish_wind)
```

Format

A numerical matrix of dimension $6000x3$.

Source

Fick, S.E., Hijmans, R.J. (2017) WorldClim 2: new 1km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*, **37**, 4302–4315.

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, coordz, coordt, coordx_dyn, corrmodel,
           data, distance, fcall, fixed, grid, likelihood,
           maxdist, neighb, maxtime, model, n, param,
           parscale, paramrange, radius, start, taper, tapsep,
           type, typereal, weighted, copula, X, memdist, nosym,
           p_neighb, thin_method)
```

Arguments

coordx	A numeric $(d \times 2)$-matrix or $(d \times 3)$-matrix. 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; optional argument, default is NULL.
coordz	A numeric vector giving 1-dimension of spatial coordinates; optional argument, default is NULL.
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, default is NULL.
corrmodel	String; the name of a correlation model.
data	A numeric vector or a $(n \times d)$-matrix or $(d \times d \times n)$-array of observations.
distance	String; the name of the spatial distance. Default is "Eucl" (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 (default) the data are interpreted as a vector or a $(n \times d)$-matrix; if TRUE then a $(d \times d \times n)$-array 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.
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 factors to improve the maximizing procedure; see optim.
paramrange	A numeric vector of parameter ranges; see optim.
radius	Numeric; the radius of the sphere in the case of lon-lat coordinates. The default is 6371, the radius of the earth.
start	A named list with the initial values of the parameters that are used by the numerical routines in the maximization procedure.
taper	String; the name of the type of covariance matrix. It can be "Standard" (default) or "Tapering" for tapered covariance matrices.

tapsep	Numeric; an optional value indicating the separability parameter in the space-time adaptive taper (see **Details**).
type	String; the type of likelihood objects. Temporary value set to "WLeastSquare" (weighted least-square) in order to compute starting values.
typereal	String; the real type of likelihood objects. See GeoFit.
weighted	Logical; if TRUE the likelihood objects are weighted; see GeoFit.
copula	String; the type of copula.
X	Numeric; matrix of space-time covariates.
memdist	Logical; if TRUE then the distances in the composite likelihood are computed before the optimization.
nosym	Logical; if TRUE symmetric weights are not considered.
p_neighb	Numeric; a value in $(0,1]$ specifying the expected fraction of nearest-neighbor pairs retained through stochastic thinning. If 1 (default), no thinning is applied and all nearest-neighbor pairs are used. If <1, pairs are randomly retained using independent Bernoulli sampling.
thin_method	String; thinning scheme used when p_neighb<1. Default is "iid" (independent Bernoulli thinning).

Details

Internal function called by WlsStart.

Value

A list with a set of useful information in the estimation procedure.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>, https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian Caamaño-Carrillo, <chcaaman@ubiobio.cl>, https://www.researchgate.net/profile/Christian-Caamano

---

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(winds)
```

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(winds.coords)
```

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 appropriate starting values for the composite-likelihood fitting procedure based on weighted least squares.

Usage

```
WlsStart(coordx, coordy, coordz, coordt, coordx_dyn, corrmodel, data,
         distance, fcall, fixed, grid, likelihood, maxdist, neighb,
         maxtime, model, n, param, parscale, paramrange, radius, start,
         taper, tapsep, type, varest, weighted, copula, X, memdist,
         nosym, p_neighb, thin_method)
```

Arguments

coordx	A numeric $(d \times 2)$-matrix or $(d \times 3)$-matrix. 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; optional argument, default is NULL.
coordz	A numeric vector giving 1-dimension of spatial coordinates; optional argument, default is NULL.
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, default is NULL.
corrmodel	String; the name of a correlation model.
data	A numeric vector or a $(n \times d)$-matrix or $(d \times d \times n)$-array of observations.
distance	String; the name of the spatial distance. Default is "Eucl" (euclidean distance).
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 (default) the data are interpreted as a vector or a $(n \times d)$-matrix; if TRUE then a $(d \times d \times n)$-array 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.
maxtime	Numeric; an optional positive value indicating the maximum temporal separation considered in the composite-likelihood computation.
model	String; the name of the model.
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 great circle distance. Default value is the radius of the earth in km (i.e., 6371).
start	A numeric vector (or list) with starting values.
taper	String; the name of the type of covariance matrix. It can be "Standard" (default) or "Tapering" for tapered covariance matrices.
tapsep	Numeric; an optional value indicating the separability 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. Default is FALSE.
weighted	Logical; if TRUE the likelihood objects are weighted; see GeoFit.
copula	String; the type of copula. It can be "Clayton" or "Gaussian".
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 optimization.
nosym	Logical; if TRUE symmetric weights are not considered.
p_neighb	Numeric; a value in $(0, 1]$ specifying the expected fraction of nearest-neighbor pairs retained through stochastic thinning. If 1 (default), no thinning is applied and all nearest-neighbor pairs are used. If <1, pairs are randomly retained using independent Bernoulli sampling.
thin_method	String; thinning scheme used when p_neighb<1. Default is "iid" (independent Bernoulli thinning).

Details

Internal function called by GeoFit.

Value

A list with a set of useful information in the estimation procedure.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua89@gmail.com>, https://sites.google.com/view/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/, Christian Caamaño-Carrillo, <chcaaman@ubiobio.cl>, https://www.researchgate.net/profile/Christian-Caamano

See Also

GeoFit.

Index