Package ‘GeoModels’

August 21, 2026

Version 2.2.8

Date 2026-08-15

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

Suggests numDeriv, memuse, mapproj, scatterplot3d, sp

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, dotCall64, FastGP, pbivnorm, sn, nabor, hypergeo, VGAM, future, minqa

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 2026-08-21 05:46:27 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 one dimension of temporal coordinates. The values need not be equally spaced; input validation does not impose a regular temporal grid.

`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 Details for the accepted options.
`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 non-negative maximum temporal-distance threshold, expressed in the same units as `coordt`.
`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 separability 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 likelihood object. Supported values are `Pairwise`, `Standard`, `Tapering`, `Tapering1`, `Tapering2`, `GeoWLS`, `CV`, and `Independence`.

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 one dimension of temporal coordinates. Optional argument; the default is `NULL`, in which case a spatial random field is expected. The values need not be equally spaced.
`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 spatio-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). Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
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 GeoFit for 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 for fitting covariance models with composite likelihood, CompLik2 and CompIndLik2 for related composite-likelihood routines, GeoVarest for variance estimation.

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 one dimension of temporal coordinates. Optional argument; the default is `NULL`, in which case a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
`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 Details for the accepted options.
`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 spatio-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 returns 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 returns 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 random field. 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. For univariate curves, omitted `mean`, `sill`, and `nugget` use the defaults 0, 1, and 0, respectively.
`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`	Positive integer size parameter. For Binomial it is the common number of trials used by this lag-only curve. For Negative Binomial it is the single common number r of successes; Geometric corresponds to $r = 1$. Site-specific Binomial sizes are handled by `GeoCovmatrix()` rather than by a single lag-only curve. Default is 1.
`covariance`	Logic; if TRUE then the covariance is returned. Default is FALSE
`variogram`	Logical; if FALSE the correlation/covariance is returned. If TRUE, `covariance=FALSE` returns the normalized semivariogram $1-\rho(h)$, while `covariance=TRUE` returns the semivariogram on the response scale.

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` *Correlation, Covariance, and Variogram Functions for Spatial Copula Models*

Description

Computes the observed-scale correlation, covariance, or semivariogram of a univariate spatial or spatio-temporal random field constructed with a Gaussian, Clayton-like, or skew-Gaussian copula. The function uses the same numerical second-order covariance engines as `GeoCovmatrix()`, `GeoCovariogram()`, and `GeoKrig()`.

Usage

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

Arguments

`x`	Numeric vector of non-negative spatial distances.
`t`	Optional numeric vector of non-negative temporal distances for a spatio-temporal correlation model.
`corrmodel`	String giving the latent Gaussian correlation model. See `GeoCovmatrix` for implemented correlation models.
`model`	String giving the marginal distribution. Copula covariance is implemented for `Gaussian`, `StudentT`, `LogGaussian`, `Gamma`, `Weibull`, `Beta`, `Beta2`, `Kumaraswamy`, `Kumaraswamy2`, `Logistic`, `SkewLaplace`, `Tukeyh`, `Tukeyh2`, and `SinhAsinh`.
`copula`	String giving the copula. Supported values are `Gaussian`, `Clayton`, and `SkewGaussian`. For the Clayton-like copula, `param$nu` must be a positive integer. For the skew-Gaussian copula, `param$nu` is the asymmetry parameter and must belong to $(-1, 1)$.

distance	String giving the spatial distance. The default is Eucl. The supplied x values are already distances; this argument is retained for consistency with the other GeoModels functions.
param	List of correlation, marginal, nugget, and copula parameters. Since this is a lag-only function and has no design matrix, location-dependent margins use an intercept-only marginal predictor mean.
radius	Numeric radius of the sphere when great-circle distances are used. The default is 6371 km.
n	Numeric number of trials, retained for interface compatibility.
covariance	Logical. If TRUE and variogram=FALSE, return observed-scale covariance. If FALSE, return observed-scale correlation.
variogram	Logical. If TRUE, return a semivariogram. With covariance=TRUE, this is the observed-scale semivariogram $V_Y - C_Y(h)$. With covariance=FALSE, it is the normalized semivariogram $1 - \rho_Y(h)$.

Details

The spatial dependence supplied by corrmodel is interpreted as the correlation $\rho(h)$ of the latent Gaussian random field underlying the copula. The marginal covariance is then obtained on the observed scale using the selected copula and marginal quantile transformation.

For Gaussian copulas, the implementation uses the Gaussian-copula covariance engine. The Tukeyh, Tukeyh2, and SinhAsinh margins are monotone transforms of one Gaussian field, so their Gaussian-copula covariance is evaluated with the corresponding exact transformed-Gaussian covariance rather than a truncated Hermite expansion. Clayton-like covariances use the cached deterministic quadrature/interpolation engine, avoiding adaptive two-dimensional integration at every lag. Skew-Gaussian covariances use the cached bivariate Hermite representation. Consequently, repeated calls with the same copula and marginal parameters can reuse cached numerical objects.

The nugget is applied to the latent correlation for non-zero lags. At the exact zero spatial and temporal lag, the function represents the same random variable and therefore returns correlation one, the marginal variance for covariance, and zero for the semivariogram.

Value

An object of class GeoCorrFct. Its corr component contains the requested correlation, covariance, or semivariogram values. The object also contains the spatial distances, temporal distances, marginal model, copula, parameters, and flags describing the requested scale.

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)

################################################################
### Correlation of a mean-reparametrized Beta random field
### with a Matern latent correlation model.
################################################################

x <- seq(0, 0.4, 0.02)

param <- list(smooth = 0.5, scale = 0.2 / 3, nugget = 0,
              mean = 0, min = 0, max = 1, shape = 2)

corr_gauss <- GeoCorrFct_Cop(
    x = x, corrmodel = "Matern", param = param,
    copula = "Gaussian", model = "Beta2"
)
plot(corr_gauss, ylab = "Correlation", lwd = 2)

## Clayton-like and skew-Gaussian copulas are selected through param$nu.
## Their fast covariance engines are also used by GeoCovmatrix and GeoKrig.
```

`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=TRUE,
 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. When omitted, the distance stored in `fitted` is used when available. The empirical lag values are taken from `vario`; the fitted and empirical objects should therefore use the same spatial metric.
`answer.cov`	Logical; if `TRUE` the estimated covariance function is returned. For a bivariate Gaussian fit, the returned components are `covariance11`, `covariance12`, and `covariance22`.

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 practical range is returned; if FALSE (the default) the practical range is not returned.
fix.lags	Integer; a positive index in the empirical spatial-lag grid (including its zero-lag entry) used for the temporal profile. The theoretical surface is matched to the corresponding lag value rather than reusing this index on a different grid. For dynamic spatial supports, if the zero-lag entry is selected but no collocated pairs exist across distinct times, the plot automatically uses the first positive-distance space-time bin containing finite empirical pairs; the fitted curve is evaluated at that bin's center and the panel title reports the bin interval.
fix.lagt	Integer; a positive index in the empirical temporal-lag grid (including its zero-lag entry) used for the spatial profile. The theoretical surface is matched to the corresponding time-lag value.
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 practical range is added on the plot; if FALSE (the default) the practical 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 scalar in [0,100]; the percentage of the marginal variance reached by the semivariogram when computing the practical range.
vario	A Variogram object obtained from the GeoVariogram procedure.
invisible	Logical; if TRUE, compute the sum of squared differences between the empirical and fitted semivariograms on the empirical lag grid. This diagnostic is computed independently of whether the variogram is plotted.
...	other optional parameters which are passed to plot functions.

Details

When a fitted univariate model has a nonconstant mean, the returned lag-only theoretical curve requires one representative marginal location. The function uses the average fitted linear predictor $X\widehat{\beta}$; for a site-specific fixed mean it uses the average of that vector. This does not alter the fitted mean used by likelihood, residual, simulation, or kriging calculations. For a univariate GeoFit object fitted with copula="Gaussian", copula="Clayton", or copula="SkewGaussian", the theoretical covariance and semivariogram are computed on the observed marginal scale with the same covariance engine used by GeoCovmatrix and GeoKrig. The supported continuous margins are "Gaussian", "StudentT", "LogGaussian", "Gamma", "Weibull", "Beta", "Beta2", "Kumaraswamy", "Kumaraswamy2", "Logistic", "SkewLaplace", "Tukeyh", "Tukeyh2", and "SinhAsinh". For the Gaussian copula, the monotone transformed-Gaussian Tukey and sinh–arcsinh margins use their exact model covariance. Clayton-like covariance uses cached deterministic copula quadrature and skew-Gaussian covariance uses the cached Hermite representation also used for linear prediction.

For "Beta2" and "Kumaraswamy2" with covariates, and more generally whenever the marginal location varies over the observation sites, a single lag-only covariance curve is not uniquely defined by distance alone. The displayed curve therefore uses the representative marginal predictor described above (the average fitted linear predictor). It is a representative theoretical semivariogram for diagnostic comparison with the empirical one.

For non-Gaussian univariate models, the practical range is computed directly from the fitted observed-scale covariance and semivariogram. This avoids using a Gaussian-correlation range as a surrogate for a transformed marginal model. If the requested percentage is not reached on the evaluated lag grid, the returned practical range is NA. For a Gaussian model, the correlation root calculation is retained. A bivariate practical range is not defined by this function and is returned as NA.

For space-time profile plots, fix.lags and fix.lagt select an empirical lag value; the closest point on the fitted 150-point surface grid is then used. This keeps empirical and theoretical profiles aligned even though their grids have different resolutions. For dynamic spatial supports, finite empirical space-time bins are displayed as points rather than vertical stems. The empirical temporal margin $\gamma(0, u)$ is used when actual collocated locations are available at distinct times. If fix.lags=1 requests that zero-lag profile but no collocated dynamic pairs exist, the plot instead uses the first positive-distance space-time bin with finite empirical values. Thus the panel represents a near-zero spatial band (for example $0 < h < h_1$) rather than displaying an empty empirical profile; the fitted curve is evaluated at the center of the selected spatial bin. The title reports the actual bin interval. No zero-valued empirical temporal observations are imputed, and the underlying GeoVariogram temporal margin remains the true collocated $\gamma(0, u)$ margin.

For a non-copula LogGaussian model, covariance and semivariogram are reported on the observed scale, including the representative factor $\exp(2\eta)$ in the marginal variance, consistently with GeoCovmatrix.

For a non-copula SinhAsinh model, the nugget attenuation is applied to the latent Gaussian correlation before the nonlinear sinh–arcsinh correlation map is evaluated. This is the same ordering used by GeoCovmatrix, GeoCorrFct, and GeoKrig.

Objects returned by GeoResiduals() with residual_type = "Pearson" are intentionally rejected. The empirical variogram of Pearson residuals is useful by itself, but it is not on the same scale as the response covariance model stored in the original discrete GeoFit object.

The function computes the fitted variogram model

Value

By default the function produces the requested plot invisibly. When answer.cov=TRUE and/or answer.vario=TRUE, it invisibly returns a list containing the lag grid and the fitted covariance and/or semivariogram. When answer.range=TRUE, the practical range is included as range. With invisible=TRUE and no requested answer, the sum-of-squares diagnostic used by the function is returned invisibly. For a bivariate Gaussian fit with answer.cov=TRUE, the list contains covariance11, covariance12, and covariance22.

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. *Springer-Verlag, New York*.

See Also

GeoFit.

Examples

```
library(GeoModels)

################################################################
###
### 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 returned by `GeoCovmatrix`.
`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 Gaussian or supported non-Gaussian random field with given underlying covariance model and a set of spatial location sites (and temporal instants).

Usage

```
GeoCovmatrix(estobj = NULL, coordx = NULL, 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,
 check.duplicates = FALSE)
```

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. The default is `NULL`; this is useful when coordinates are supplied through `coordx_dyn`, `estobj`, or `spobj`.
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 one dimension of temporal coordinates. Optional argument; the default is NULL, in which case a spatial random field is expected. At the moment implemented only for the Gaussian case. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
coordx_dyn	For dynamic locations, a list of length T, where T is length(coordt). Element coordx_dyn[[t]] contains the two- or three-dimensional locations at coordt[t]; covariance-matrix rows and columns concatenate these blocks by time.
corrmodel	String; the name of a correlation model; see GeoCovmatrix for the list of implemented correlation models.
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 spatio-temporal realisations on a set of non-equispaced spatial sites (irregular grid). See GeoFit.
n	Positive integer size parameter. For a direct Binomial random field it may be a scalar or one value per observation. For a direct Negative-Binomial random field it is the common number r of successes and must be scalar; the Geometric model is the special case $r = 1$. Default is 1.
model	String; the type of random field. 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; optional copula specification. Observed-scale covariance-matrix construction is implemented for "Gaussian", "Clayton", and "SkewGaussian" for the supported continuous copula margins.
X	Numeric design matrix used to evaluate the mean when required by the marginal covariance. For fixed sites, rows follow time-major order; for dynamic sites, supply a stacked matrix in temporal-block order or a list aligned with coordx_dyn.
spobj	An object of class sp or spacetime. For space-time objects, the current sp2Geo() conversion uses sequential temporal indices and does not preserve irregular original time spacing; to retain irregular temporal distances, use the explicit-coordinate interface with numeric coordt.
check.duplicates	Logical. If TRUE, perform a fast exact scan for duplicated observation locations at this user entry point. The default is FALSE, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

Details

For plain (non-copula) bivariate correlation models, observed-scale covariance is currently validated only for Gaussian margins; unsupported non-Gaussian bivariate combinations are rejected explicitly. For direct Negative-Binomial fields, the public argument `n` denotes the single common integer r in the waiting-time construction. For `copula="Gaussian"`, `copula="Clayton"`, and `copula="SkewGaussian"`, continuous-margin covariance matrices are constructed directly on the observed scale. The implementation supports `"Gaussian"`, `"StudentT"`, `"LogGaussian"`, `"Gamma"`, `"Weibull"`, `"Beta"`, `"Beta2"`, `"Kumaraswamy"`, `"Kumaraswamy2"`, `"Logistic"`, `"SkewLaplace"`, `"Tukeyh"`, `"Tukeyh2"`, and `"SinhAsinh"`. Under the Gaussian copula, covariance for the last three is evaluated with the exact covariance of the corresponding monotone transformed-Gaussian field. For location-dependent copula margins, pairwise covariances use the two site-specific marginal predictors supplied through `X` or an external mean vector. Clayton-like covariance uses cached deterministic copula quadrature; skew-Gaussian covariance uses a cached bivariate Hermite expansion of the latent representation, avoiding pair-by-pair adaptive two-dimensional integration. The count margins `"Poisson"`, `"Binomial"`, and `"BinomialNeg"` are also available for Gaussian, Clayton-like, and skew-Gaussian copula covariance in the global prediction path; Binomial currently requires a common trial count `n`. The same observed-scale covariance engines are used by `GeoCovariogram` and `GeoKrig`, so covariance matrices, fitted semivariogram curves, and linear prediction are numerically consistent.

For univariate models whose covariance depends on the marginal location, the mean is evaluated as $X\beta$, where coefficients `mean`, `mean1`, and so on correspond in order to the columns of `X`. With `X=NULL`, an intercept-only design is used. A vector `param$mean` may instead supply one known mean per observation and is mutually exclusive with `X`.

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, rows and columns are in variable-block order. For common locations the covariance matrix is associated with

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

For heterotopic locations supplied by `coordx_dyn=list(coords1,coords2)`, the first block contains all observations of variable 1 at `coords1` and the second contains all observations of variable 2 at `coords2`.

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:

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

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

12. GenWend_GenWend_nosep defined as:

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

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:

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

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:

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

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: the value supplied by the user, indicating whether the original spatial coordinates described a regular grid.
`nozero`	In the case of tapered matrix the percentage of non-zero values in the covariance matrix; otherwise `NULL`.
`n`	The Binomial number of trials, or the common Negative-Binomial number r of successes.
`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 scalar covariance and marginal parameters.
`mean_external`	For a univariate model, the optional known mean vector supplied through `param$mean`. It is stored separately from `param`, in observation order, so that covariance and prediction routines do not interpret its entries as regression coefficients.
`spacetime`	`TRUE` if spatio-temporal and `FALSE` if spatial covariance model.
`sparse`	Logical: is the returned object of class `spam`?
`call`	The matched call used to create the object.

Spatio-temporal ordering

For fixed locations, covariance rows and columns are ordered

```
(s[1], t[1]), ..., (s[N], t[1]),
(s[1], t[2]), ..., (s[N], t[2]), ...
```

For dynamic locations, all rows of `coordx_dyn[[1]]` come first, then all rows of `coordx_dyn[[2]]`, and so on. The rows of `X` and an external `param$mean` vector use this same order. See `GeoModels-spacetime-ordering`.

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

GeoFit for parameter estimation, GeoSim for simulation, GeoKrig for prediction, CorrParam for correlation-parameter names.

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 *Repeated holdout kriging cross-validation*

Description

The procedure uses the GeoKrig or GeoKrigloc function to compute repeated holdout kriging cross-validation using information from a GeoFit object. The function returns 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 = FALSE, ncores = 6, progress = TRUE,
  seed = NULL
)
```

Arguments

`fit`	An object of class GeoFit.
`K`	Positive integer greater than or equal to 2 giving the number of cross-validation iterations. Non-integer values are rejected rather than rounded.
`estimation`	Logical; if `TRUE`, the model is re-estimated on the training observations at each iteration and the fold-specific estimates are used for prediction. If `FALSE`, the parameter estimates in `fit` are reused; this is therefore a conditional predictive assessment given parameters estimated from the complete dataset, not a fully refitted out-of-sample cross-validation.
`optimizer`	The type of optimization algorithm if `estimation` is `TRUE`. See GeoFit for details. If `NULL`, then the optimization algorithm stored in `fit` is used.
`lower`	An optional named list giving the values for the lower bounds of the parameters when bounded optimization is used and `estimation` is `TRUE`.
`upper`	An optional named list giving the values for the upper bounds of the parameters when bounded optimization is used and `estimation` is `TRUE`.
`n.fold`	Numeric; the fraction of observations randomly deleted and predicted in each cross-validation iteration. At least one training and one prediction observation are retained. In the space-time case, sampling also retains at least one training observation at every observed time.
`local`	Logical; if `TRUE`, 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 spatial neighborhood distance if local kriging is performed.
`maxtime`	Numeric; an optional non-negative temporal-distance threshold, expressed in the same units as the fitted `coordt`, when local kriging is performed.
`sparse`	Logical; if `TRUE`, kriging and simulation are computed with sparse matrix algorithms using the **spam** package. The default is `FALSE`. It should be used with compactly supported covariance models.

type_krig
: String; the type of kriging. If "Simple", the default, the fitted mean coefficients are treated as plug-in values. If "Universal", the prediction MSE is additionally corrected using the covariance matrix of the estimated mean coefficients. With estimation = TRUE, Universal cross-validation is supported for full-likelihood refits, for which varest = TRUE is requested automatically in each fold. It is deliberately rejected for composite-likelihood refits because each fold would require its own Godambe covariance matrix. With estimation = FALSE, a stored fit$varcov may be used; for composite likelihood it can be obtained with GeoVarest.

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

parallel
: Logical; default FALSE. Set TRUE to evaluate the cross-validation iterations in parallel.

ncores
: Positive integer or NULL; default 6. With parallel=TRUE, an explicit integer requests that many workers, capped by detected cores and the number of cross-validation jobs. Set ncores=NULL for automatic selection, capped at six workers and normally leaving one detected core free. Non-integer values are rejected.

progress
: Logical; if TRUE, a progress bar is shown.

seed
: Optional finite integer seed used to make the random selection of folds and any stochastic refitting step reproducible. Non-integer values are rejected. If NULL, the current random number generator state is left unchanged. When supplied, the previous .Random.seed is restored on exit.

Details

For a spatio-temporal GeoFit object, the stored temporal coordinates are reused without imposing equal spacing. Local temporal neighborhoods interpret maxtime as a distance threshold in the same units as those coordinates.

The function randomly removes a fraction n.fold of the observations at each iteration, predicts the removed observations using kriging, and computes a set of predictive scores.

If estimation = TRUE, the model is re-estimated at each cross-validation iteration before prediction. Refits preserve the fitted likelihood settings, including pair weighting, pair thinning, distance-memory handling and anisotropy. For misspecified estimators, the response model stored in fit$model remains distinct from the working likelihood stored in fit$estimation_model; the latter is used for each refit. Estimated anisotropy parameters are passed once through anisopars, avoiding duplicate angle/ratio entries in the starting or fixed parameter lists.

For Universal kriging with fold-specific re-estimation, full-likelihood fits request varest = TRUE within each fold. Composite-likelihood refits are not assigned the Godambe covariance matrix from the full dataset: GeoCV instead rejects estimation = TRUE, type_krig = "Universal" for composite likelihood, because a statistically coherent analysis would require a fold-specific GeoVarest calculation.

If estimation = FALSE, the parameter estimates from the original fit are reused after removing the validation observations from the conditioning data. Consequently, the reported errors assess prediction conditional on parameters that were estimated using the complete dataset and may be more optimistic than a fully refitted out-of-sample cross-validation.

For Poisson, Binomial, and BinomialNeg with copula="SkewGaussian", global cross-validation (local=FALSE) supports both fold refitting and fixed-parameter prediction. Discrete Clayton-like copula models still require estimation=FALSE until their pairwise likelihood is implemented.

Regular-grid fits are converted to explicit point coordinates after observations are removed, because each training sample is no longer a complete Cartesian grid. Space-time covariates may be stored either as one matrix in observation order or as a list containing one matrix per time.

For Gaussian fitted models all documented scores are computed from the Gaussian predictive mean and MSE. For non-Gaussian fitted models, only RMSE, MAE and MAD are returned; brie, crps, lscore, pit, intscore and coverage are set to NA, because an exact non-Gaussian predictive distribution is not available from only a mean and MSE.

When seed is supplied, the cross-validation samples are reproducible. The function preserves and restores the user's random number generator state.

Value

Returns a list containing the following components:

rmse	The vector of root mean squared errors.
mae	The vector of mean absolute errors.
mad	The vector of median absolute errors.
brie	The vector of Brier scores, or NA for non-Gaussian fits.
crps	The vector of continuous ranked probability scores, or NA for non-Gaussian fits.
lscore	The vector of log-scores, or NA for non-Gaussian fits.
pit_mean	The vector of mean probability integral transform values within each holdout fold, or NA for non-Gaussian fits. This is a descriptive PIT summary, not a calibration score: a mean near 0.5 does not by itself imply a uniform PIT distribution.
pit	Backward-compatible alias of pit_mean.
intscore	The vector of interval scores, or NA for non-Gaussian fits.
coverage	The vector of empirical coverage values, or NA for non-Gaussian fits.
seed	The seed used for reproducibility, or 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

See Also

GeoKrig, GeoKrigloc, GeoFit.

Examples

```
library(GeoModels)

################################################################
### Example of spatial kriging cross-validation
################################################################

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

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

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)

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, seed = 123)
#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, MAD, logarithmic score and CRPS from exact drop-one linear-prediction identities based on a `GeoCovmatrix` object.

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 spatio-temporal realisation). or a a$(2 \times d)$-matrix (a single bivariate realisation).
`method`	String; matrix decomposition used for dense covariance matrices. Possible values are `"cholesky"` (default) and `"svd"`. Sparse covariance matrices use the sparse Cholesky path and therefore require `"cholesky"`.
`matrix`	An object returned by `GeoCovmatrix`.

Details

Let $Q = \Sigma^{-1}$ and let r denote the data after subtraction of the model mean. The drop-one residual and conditional variance are computed without repeatedly refitting the model:

$$e_i^{(-i)} = (Qr)_i/Q_{ii}, \qquad v_i^{(-i)} = 1/Q_{ii}.$$

The standardized residual is $z_i = (Qr)_i/\sqrt{Q_{ii}}$. The logarithmic score uses $\frac{1}{2}\{\log(2\pi v_i^{(-i)}) + z_i^2\}$, and the Gaussian CRPS uses its standard closed-form expression with the normal density ϕ and distribution function Φ. The SVD path solves the linear system through the SVD itself rather than passing an SVD object to triangular solvers.

Value

Returns a list containing the following information:

`RMSE`	Root-mean-square error predictive score
`MAE`	Mean absolute drop-one prediction error.
`MAD`	Median absolute drop-one prediction error.
`LSCORE`	Mean Gaussian negative log predictive density for the drop-one predictions.
`CRPS`	Mean Gaussian continuous ranked probability score for the drop-one predictions.

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

Description

Maximum weighted (stochastic) 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="bernoulli", type="Pairwise", upper=NULL,
 varest=FALSE, weighted=FALSE,
 X=NULL, spobj=NULL, spdata=NULL, check.duplicates=FALSE)
```

Arguments

`data` A d-dimensional vector (a single spatial realisation) or a $(d \times d)$-matrix (a single spatial realisation on regular grid) or 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). See Details for the accepted data layouts.

`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 one dimension of the observation-time coordinates. Optional argument, default is `NULL`; if `NULL`, a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.

`coordx_dyn` For dynamic locations, a list with one numeric coordinate matrix per temporal instant. If T is `length(coordt)`, the list must have length T; `coordx_dyn[[t]]` must have N_t rows and two or three columns. Its row order must match `data[[t]]` and, when supplied as a list, `X[[t]]`. See `GeoModels-spacetime-ordering`.

`copula` String; the copula used by pairwise copula likelihoods. Supported values are `"Gaussian"`, `"Clayton"` (the constructive Clayton-like spatial copula), and `"SkewGaussian"`.

`corrmodel` String; the name of a correlation model; see `GeoCovmatrix` for the list of implemented correlation models.

`distance` String; the name of the spatial distance. Default is `"Eucl"` (Euclidean distance). See Details for the accepted options.

`fixed` An optional named list giving the values of the parameters that will be considered as known values. Parameter names must match exactly the nuisance/marginal or correlation parameters supported by the selected model and correlation model; invalid names are reported explicitly. 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. Geometric anisotropy is available only with distance="Eucl".

`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. If TRUE, coordx, coordy (and optionally coordz) are interpreted as grid axes and the data dimensions must match those axes; grid inputs are internally converted to the canonical explicit-coordinate layout before fitting.

`likelihood` String; the configuration of the composite likelihood. "Marginal" is the default; see Details for the accepted options.

`lower` An optional named list giving lower bounds for parameters when the optimizer is L-BFGS-B, nlminb, bobyqa or optimize. Names must match parameters that are being estimated.

`maxdist` Numeric; an optional positive value indicating the maximum spatial distance considered in the composite computation. See Details for more information.

`neighb` Numeric; an optional positive integer indicating the order of neighborhood in the composite likelihood computation. See Details for more information.

`p_neighb` Numeric scalar in $(0,1]$. If 1 (default), no thinning is applied. For thin_method="bernoulli", it is the candidate-pair inclusion probability in the constant-weight design and controls the retained size in expectation. For thin_method="FixedBudget", it defines the exact global retained-pair budget $K = \mathrm{round}(p_{neighb} d)$, where d is the candidate-pair count.

`maxtime` Numeric; an optional non-negative maximum temporal-distance threshold, expressed in the same units as coordt, used to select pairwise composite-likelihood contributions.

`memdist` Deprecated logical argument retained for backward compatibility. The selected pair structure is always precomputed and reused during composite-likelihood optimization. Supplying FALSE produces a warning and is treated as TRUE.

`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 Details for the accepted options.

`n` Positive integer size parameter. For direct Binomial models it may be scalar or contain one value per observation. For direct Negative-Binomial models it is the common number r of successes and must be scalar; Geometric corresponds to $r = 1$. Non-integer or non-positive values are rejected.

`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. Parameter names must match exactly the nuisance/marginal or correlation parameters supported by the selected model and correlation model; invalid names are reported explicitly. Parameters omitted from start receive conservative internal initial values; explicitly supplied values are preserved (see Details).
thin_method	String; thinning scheme in stochastic weighted pairwise likelihood (used when p_neighb < 1). Default is "bernoulli" (independent Bernoulli thinning). An alternative thinning scheme is "FixedBudget". The legacy name "TargetBalanced" is accepted as an alias for "FixedBudget".
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 parameters that are being estimated.
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 GeoVarest. Default is FALSE.
weighted	Logical; if TRUE the likelihood objects are weighted; see Details for the accepted options. Default is FALSE.
X	Numeric design matrix for the linear mean $X\beta$. Its columns correspond in order to mean, mean1, ... For fixed-location space-time data, rows follow c(t(data)): all sites at the first time, then all sites at the second time, and so on. For dynamic locations, use either a stacked matrix in time-wise order or a list with X[[t]] aligned with coordx_dyn[[t]]. See GeoModels-spacetime-ordering.
spobj	An object of class sp or spacetime. For space-time objects, the current sp2Geo() conversion uses sequential temporal indices and does not preserve irregular original time spacing; to retain irregular temporal distances, use the explicit-coordinate interface with numeric coordt.
spdata	Character; the name of the data component in the sp or spacetime object.
check.duplicates	Logical. If TRUE, perform a fast exact scan for duplicated observation locations at this user entry point. The default is FALSE, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

Details

For a univariate model the mean is specified as

$$\mu = X\beta.$$

The entries of β are represented by the parameters mean, mean1, and so on. The columns of X correspond to these coefficients in exactly this order. When X=NULL, an intercept-only model is used, which is equivalent to a one-column matrix of ones and the single coefficient mean. Missing starting values for regression coefficients and other estimated parameters are initialized internally. Explicit entries in start always define starting values for parameters that remain estimated, including likelihood="Full", type="Standard"; they are never converted to fixed mean coefficients. A coefficient is fixed only when it is supplied in fixed. Supplied coefficient names must be compatible with ncol(X). For difficult non-Gaussian models, user-supplied starting values can still improve numerical optimization.

A site-specific known mean can instead be supplied as a vector in fixed$mean, with one value per observation. This external mean is mutually exclusive with X and with estimated mean coefficients in start. A scalar fixed$mean is not an external vector: it is the fixed intercept coefficient.

GeoFit provides weighted and stochastic weighted composite-likelihood estimation based on pairs for Gaussian and non-Gaussian random fields, including nearest-neighbor and stochastic nearest-neighbor pairwise likelihoods; see Caamaño-Carrillo et al. (2024) and Bevilacqua et al. (2026). It also provides independence composite-likelihood estimation. The accepted likelihood/type combinations are checked explicitly: likelihood="Full" uses the full-likelihood types (including "Standard"). Full-likelihood objectives are currently implemented for Gaussian, SinhAsinh, LogGaussian, Tukeyh, Tukeyh2, and the internally supported misspecified-Gaussian full likelihoods. Other margins, including Gamma and Weibull, are rejected before optimization with an explicit capability message; likelihood="Marginal" is used with "Pairwise" or "Independence"; and likelihood="Conditional" is used with "Pairwise". The historical Difference composite likelihood is no longer supported. For space-time pairwise fitting, GeoFit checks that a registered space-time native kernel exists for the requested marginal model and stops before optimization when it does not. Bivariate pairwise fitting is currently implemented only for the Gaussian model with marginal pairwise likelihood.

For pairwise copula fitting, the current implementation is univariate and purely spatial. Continuous margins supported with the Gaussian, Clayton-like, and skew-Gaussian copulas are "Gaussian", "StudentT", "LogGaussian", "Gamma", "Weibull", "Beta", "Beta2", "Kumaraswamy", "Kumaraswamy2", "Logistic", and "SkewLaplace". The Gaussian and skew-Gaussian copulas additionally support the discrete margins "Poisson", "Binomial", and "BinomialNeg". For the skew-Gaussian copula these probabilities are evaluated as copula-rectangle probabilities; when nu=0 the calculation reduces exactly to the Gaussian copula likelihood. For copula="Clayton", nu is the positive-integer parameter of the constructive Clayton-like random field and must be supplied in fixed; it is not a continuously estimable copula parameter. For copula="SkewGaussian", nu is the bounded asymmetry parameter $\eta \in (-1, 1)$ and invalid optimizer proposals are rejected by the objective function.

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.

Observation coordinates are checked for exact duplicates before fitting. For space-time data, the full space-time point must be unique; the same spatial site observed at different times is allowed. The check is hash-based and is performed once at the user-facing fit, not during bootstrap/refit iterations that reuse an already validated design. If stochastic thinning retains no pairwise contributions, fitting stops with an explicit error rather than optimizing an empty criterion.

For `model="PoissonGamma"` and `model="PoissonGammaZIP"`, pairwise fitting accepts every finite `shape>0`. This is the continuous Kibble–Gamma pairwise extension used by the marginal moments, correlation function, and bivariate probabilities. The constructive random-field simulation based on a finite sum of squared Gaussian fields is more restrictive and requires $2\,shape$ to be a positive integer; `GeoSim` enforces that simulation constraint explicitly.

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

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 after a `GeoVarest` call; 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 maximum temporal-distance threshold used in the composite likelihood, expressed in the same units as `coordt`.

`message` Extra message passed from the numerical routines.

`model` The density associated to the likelihood objects.

`estimation_model` The inferential model/working likelihood supplied to `GeoFit`. For misspecified fits this differs from `model`, which records the data-generating response model used by simulation, diagnostics, and prediction routines.

`missp` `TRUE` if a misspecified Gaussian model is used in the composite likelihood.

`n` The Binomial number of trials, or the common Negative-Binomial number r of successes.

`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.

Spatio-temporal ordering

For fixed spatial locations, `data` is a $T \times N$ matrix: row t corresponds to `coordt[t]` and column i to row i of `coordx`. Observation-level quantities are ordered as `c(t(data))`, hence `X` and a vector `fixed$mean` use the order time then site.

For dynamic locations, `coordx_dyn`, `data`, and optionally a list-valued `X` have one aligned element per time. Element t contains the coordinates, responses, and covariate rows observed at `coordt[t]`; internal concatenation is by increasing list index. See `GeoModels-spacetime-ordering` for the complete convention.

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:

Alegría, A., Caro, S., Bevilacqua, M., Porcu, E. and Clarke, J. (2017). Estimating covariance functions of multivariate skew-Gaussian random fields on the sphere. *Spatial Statistics*, **22**, 388–402. doi:10.1016/j.spasta.2017.05.003

Alegría, A., Bevilacqua, M. and 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., Caamaño-Carrillo, C. and Gaetan, C. (2020). On modelling positive continuous data with spatio-temporal dependence. *Environmetrics*, **31**(7), e2628. doi:10.1002/env.2628

Bevilacqua, M., Caamaño-Carrillo, C., Arellano-Valle, R. B. and Morales-Oñate, V. (2021). Non-Gaussian geostatistical modeling using (skew) t processes. *Scandinavian Journal of Statistics*, **48**(1), 212–245. doi:10.1111/sjos.12447

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

Bevilacqua, M., Caamaño-Carrillo, C., Arellano-Valle, R. B. and Gómez, C. (2022). A class of random fields with two-piece marginal distributions for modeling point-referenced data with spatial outliers. *TEST*, **31**(3), 644–674. doi:10.1007/s11749021007975

Morales-Navarrete, D., Bevilacqua, M., Caamaño-Carrillo, C. and Castro, L. M. (2024). Modelling point referenced spatial count data: A Poisson process approach. *Journal of the American Statistical Association*, **119**(545), 664–677. doi:10.1080/01621459.2022.2140053

Bevilacqua, M., Alvarado, E. and Caamaño-Carrillo, C. (2024). A flexible Clayton-like spatial copula with application to bounded support data. *Journal of Multivariate Analysis*, **201**, 105277. doi:10.1016/j.jmva.2023.105277

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 and Methods*, **107**, 268–280. doi:10.1080/01621459.2011.646928

Bevilacqua, M. and Gaetan, C. (2015). Comparing composite likelihood methods based on pairs for spatial Gaussian random fields. *Statistics and Computing*, **25**(5), 877–892. doi:10.1007/s11222-01494713

Caamaño-Carrillo, C., Bevilacqua, M., López, C. and Morales-Oñate, V. (2024). 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*, **191**, 107887. doi:10.1016/j.csda.2023.107887

Bevilacqua, M., Cuevas-Pacheco, F. and Caamaño-Carrillo, C. (2026). Fast stochastic nearest neighbor pairwise composite likelihood for massive spatial datasets. *arXiv preprint*, arXiv:2607.06142. doi:10.48550/arXiv.2607.06142

See Also

GeoCovmatrix for covariance matrix construction, GeoSim for simulation, GeoKrig for prediction, GeoVariogram and GeoWLS for variogram-based tools, GeoVarest for bootstrap variance estimation.

Examples

```
library(GeoModels)

################################################################
############ Examples of spatial Gaussian random fields ################
################################################################


# 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 field:
data <- GeoSim(coordx=coords,corrmodel=corrmodel, param=param,X=X)$data



################################################################
###
### Example 0. Maximum independence composite likelihood fitting of
### a Gaussian random field (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 field 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. Maximum stochastic marginal pairwise likelihood
### fitting of a Gaussian random field using Nelder-Mead
###
###############################################################
```

```
#N=100000 # number of location sites
#x <- runif(N, 0, 1)
#y <- runif(N, 0, 1)
#coords <- cbind(x,y)
#X=cbind(rep(1,N),runif(N))
#data <- GeoSimapprox(coordx=coords,method="TB",L=20000,parallel=TRUE,
#   corrmodel=corrmodel, param=param,X=X)$data
#fixed<-list(nugget=nugget)
#start<-list(mean=mean,mean1=mean1,scale=scale,sill=sill,smooth=smooth)

#fit2 <- GeoFit(data=data,coordx=coords,corrmodel=corrmodel,
# neighb=3,likelihood="Marginal",optimizer="Nelder-Mead",
# p_neighb=0.2, thin_method="bernoulli",
# type="Pairwise", start=start,fixed=fixed,X=X)
#print(fit2)

###############################################################
###
### Example 3. Standard Maximum likelihood fitting of
### a Gaussian random field 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 fields #############
###############################################################
```

```
################################################################
###
### Example 4. Maximum pairwise likelihood fitting of a Weibull random field
### 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 field:
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 %*% unlist(fit$param)[1:2])
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 5. 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 6. Maximum pairwise likelihood fitting of
### a Bernoulli random field 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)) # matrix 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 field:
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 %*% unlist(fit$param))
fixed <- list(nugget=0,power2=4,mean=Xb)
start <- list(scale=0.2)
lower <- list(scale=0)
upper <- list(scale=2)
# Maximum Marginal pairwise likelihood:
fit1 <- GeoFit(data=data, coordx=coords, corrmodel=corrmodel, n=1,
 likelihood="Marginal", type="Pairwise", neighb=3,
 model="Binomial", start=start, fixed=fixed,
 lower=list(scale=0.001), upper=list(scale=1))

print(fit1)


## estimating jointly marginal and dependence 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="Marginal", type="Pairwise", neighb=3,
 model="Binomial", start=start, fixed=fixed)

print(fit2)
```

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

################################################################
###
### Example 7. 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 field ###########
###############################################################

################################################################
### Example 8. 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 field:
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 *Fitting Gaussian and Non-Gaussian Random Fields with Automatic Marginal Initialization*

Description

A univariate starting-value wrapper around GeoFit. The function optionally performs a preliminary fit under spatial independence to estimate available marginal parameters, uses those estimates to update the starting values, and then calls GeoFit for the requested final fit. If the preliminary independence fit is unavailable or fails, the original starting values are retained. Bivariate correlation models are not supported.

Usage

```
GeoFit2(data, coordx = NULL, 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 = "bernoulli", type = "Pairwise",
upper = NULL, varest = FALSE, weighted = FALSE, X = NULL,
spobj = NULL, spdata = NULL, independence_start = TRUE,
independence_optimizer = "Nelder-Mead",
warn_independence_failure = TRUE, check.duplicates = FALSE)
```

Arguments

`data`	A d-dimensional vector (a single spatial realisation) or a $(d \times d)$-matrix (a single spatial realisation on regular grid) or a $(t \times d)$-matrix (a single spatio-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatio-temporal realisation on regular grid). See Details for the accepted data layouts.
`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. It may be omitted when coordinates are supplied through `coordx_dyn` or `spobj`.
`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 one dimension of the observation-time coordinates. Optional argument, default is `NULL`; if `NULL`, a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
`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; see `GeoCovmatrix` for the list of implemented correlation models.
`distance`	String; the name of the spatial distance. Default is `"Eucl"` (Euclidean distance). See Details for the accepted options.
`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. Geometric anisotropy requires `distance="Eucl"`.
`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 spatio-temporal realisations on a set of non-equispaced spatial sites. If `TRUE`, the grid axes and data dimensions are handled by `GeoFit` and converted internally to the canonical explicit-coordinate layout before fitting.

likelihood	String; the configuration of the composite likelihood. "Marginal" is the default; see Details for the accepted options.
lower	An optional named list giving lower bounds for parameters when the optimizer is L-BFGS-B, nlminb, bobyqa or optimize. Names must match parameters that are being estimated.
maxdist	Numeric; an optional positive value indicating the maximum spatial distance considered in the composite likelihood computation. See Details for more information.
neighb	Numeric; an optional positive integer indicating the order of neighborhood in the composite likelihood computation. See Details for more information.
p_neighb	Numeric scalar in $(0, 1]$. If 1 (default), no thinning is applied. For thin_method="bernoulli", it is the candidate-pair inclusion probability in the constant-weight design and controls the retained size in expectation. For thin_method="FixedBudget", it defines the exact global retained-pair budget $K = \mathrm{round}(p_{neighb} d)$, where d is the candidate-pair count.
maxtime	Numeric; an optional non-negative maximum temporal-distance threshold, expressed in the same units as coordt, used to select pairwise composite-likelihood contributions.
memdist	Deprecated logical argument retained for backward compatibility. The selected pair structure is always precomputed and reused during composite-likelihood optimization. Supplying FALSE produces a warning and is treated as TRUE.
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 Details for the accepted options.
n	Positive integer scalar, or one positive integer per observation, for models using a Binomial/Negative-Binomial trial or success count.
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; omitted parameters receive internal starting values (see Details).

thin_method
: String; thinning scheme used when p_neighb < 1. Default is "bernoulli" (independent Bernoulli thinning). An alternative thinning scheme is "FixedBudget". The legacy name "TargetBalanced" is accepted as an alias for "FixedBudget".

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 parameters that are being estimated.

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 GeoVarest. Default is FALSE.

weighted
: Logical; if TRUE the likelihood objects are weighted; see Details for the accepted options. 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.

independence_start
: Logical; if TRUE (default), GeoFit2 attempts a preliminary marginal fit under spatial independence before the final fit. The preliminary fit is skipped when the requested fit already has type="Independence". Bivariate correlation models are not supported by GeoFit2; use GeoFit directly for those models.

independence_optimizer
: String; optimizer used only for the preliminary independence fit. Default is "Nelder-Mead". The optimizer used for the final fit is still specified by optimizer.

warn_independence_failure
: Logical; if TRUE (default), a warning is issued when the preliminary independence fit is unavailable or fails. In that case the supplied start values are used unchanged.

check.duplicates
: Logical. If TRUE, perform a fast exact scan for duplicated observation locations at this user entry point. The default is FALSE, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

Details

GeoFit2 does not implement a second fitting engine. It is a wrapper around the canonical GeoFit function.

When independence_start=TRUE, the function first constructs a preliminary call to GeoFit with likelihood="Marginal" and type="Independence". The preliminary fit estimates the marginal parameters available for the selected model. When start is supplied, matching marginal entries and omitted mean coefficients are updated as before. When start=NULL, all eligible marginal estimates from the independence fit are used, while dependence parameters use the conservative automatic initial values prepared by GeoFit. Parameters supplied in fixed always take precedence.

The final model is then fitted by calling GeoFit with the original requested likelihood, likelihood-object type, optimizer, bounds, pair-selection settings, and the updated starting values. Consequently, the objective function, parameter constraints, and returned standard GeoFit components are the same as in a direct call to GeoFit; only the starting values may differ.

For Gaussian models, sill is the total marginal variance and nugget attenuates off-diagonal correlation. Therefore an independence estimate may initialize sill, while a user-supplied nugget is preserved unchanged and is never added to or subtracted from sill.

Bivariate correlation models are not supported by GeoFit2; use GeoFit directly. The preliminary initialization is skipped when type="Independence". If an independence likelihood is not implemented for the selected model, or if the preliminary optimization fails, the final fit still proceeds using the supplied starting values or, when start=NULL, the internal starting values prepared by GeoFit. This behavior can be controlled with warn_independence_failure.

Stochastic thinning of nearest-neighbor pairs in the final fit can be enabled through p_neighb < 1; thin_method specifies the thinning scheme.

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 after a GeoVarest call; 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 maximum temporal-distance threshold used in the composite likelihood, expressed in the same units as coordt.
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 random field.
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 random field.
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.
start_original	The named list of starting values supplied by the user.
start_used	The named list of starting values passed to the final GeoFit call after any successful marginal initialization.
independence_fit	The preliminary independence GeoFit object, or NULL when the preliminary fit was skipped or failed.
independence_start_message	A character string describing why the preliminary initialization was unavailable or failed, or NULL when no such message was generated.

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

The methodological references for maximum weighted composite-likelihood fitting of Gaussian and non-Gaussian random fields are reported in GeoFit.

See Also

GeoCovmatrix for covariance matrix construction, GeoSim for simulation, GeoKrig for prediction, GeoVariogram and GeoWLS for variogram-based tools, GeoVarest for bootstrap variance estimation.

Examples

```
library(GeoModels)


###############################################################
############ Examples of spatial Gaussian random fields ################
###############################################################

################################################################
###
### 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 random fields with exponential correlation.
###
###############################################################
fit1 <- GeoFit2(data=data,coordx=coords,corrmodel=corrmodel,
 neighb=3,likelihood="Conditional",
 type="Pairwise", start=start,fixed=fixed,X=X)
```

```
print(fit1)



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


################################################################
###
### 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 random fields 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=NULL, 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,
progress=FALSE,check.duplicates=FALSE)
```

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 spatio-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatio-temporal realisation on regular grid) giving the data used for prediction.

`coordx` A numeric $d \times 2$ or $d \times 3$ coordinate matrix. It may be omitted when `coordx_dyn` or `spobj` is supplied. 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 the temporal coordinates of the observations. The default is `NULL`, in which case a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.

`coordx_dyn` For dynamic observation locations, a list with one two- or three-column coordinate matrix per element of `coordt`. The rows of `coordx_dyn[[t]]` correspond to `data[[t]]` and to the same temporal block of `X`. Dynamic observation locations do not alter the ordering of prediction tasks. See `GeoModels-spacetime-ordering`.

`corrmodel` String; the name of a correlation model, for the see `GeoCovmatrix` for the list of implemented correlation models.

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

`grid` Logical; if `FALSE` (the default) the data used for prediction are interpreted as spatial or spatio-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; matrix decomposition used to solve the kriging system. The choices are `cholesky` (default) and `svd`. Only `cholesky` is available when `sparse=TRUE`.

`n` Positive integer size parameter. For Binomial it may be scalar or contain one value per observation. For direct Negative Binomial it is the single common number r of successes; Geometric corresponds to $r = 1$.

`nloc` Positive integer size at prediction tasks. For Binomial, if observation-side `n` is location-specific then `nloc` is required and is never inferred by averaging `n`; otherwise a scalar `n` is reused when `nloc=NULL`. For direct Negative Binomial, `nloc`, if supplied, must equal the common r; for Geometric it must equal 1.

`mse` Logical; if `TRUE`, compute the MSE of the kriging predictor. The default is `FALSE`; in that case the returned `mse` component is `NULL`, including for fixed- and dynamic-support space-time prediction.

`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 Details for the accepted options.

`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 vector giving the temporal instants to be predicted. Values need not be equally spaced and are interpreted on the same numeric time scale as `coordt`. The default is `NULL`, in which case only spatial prediction is performed.

`type_krig` String; `Simple` (default) treats the supplied or fitted mean coefficients as known. `Universal` uses the same plug-in coefficients and, when `mse=TRUE`, adds their estimation uncertainty from `varcov` to the prediction MSE.

`weigthed` Logical legacy argument retained for backward compatibility. It has no effect in standard kriging.

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

`copula` String; optional copula specification. Linear prediction is implemented for `"Gaussian"`, the constructive Clayton-like copula `"Clayton"`, and `"SkewGaussian"`.

`X` Numeric design matrix at the observations. For fixed-location space-time data, rows follow `c(t(data))`, i.e. time then site. For dynamic observations, use a stacked matrix ordered by temporal blocks, or a list aligned with `coordx_dyn`.

`Xloc` Numeric design matrix at prediction tasks. For space-time prediction its rows are location-major: all requested times for `loc[1, ]`, then all requested times for `loc[2, ]`, and so on.

`Mloc` Numeric vector giving the known marginal location predictor at prediction tasks, in the same location-major order as `Xloc`. For Gaussian additive margins this is the mean itself; for models with link-scale parametrizations (for example Gamma/Weibull/LogGaussian or Beta2) it is on the same scale as `param$mean`. Use either `Mloc` or `Xloc`, not both.

`spobj` An object of class `sp` or `spacetime`. For space-time objects, the current `sp2Geo()` conversion uses sequential temporal indices and does not preserve irregular original time spacing; to retain irregular temporal distances, use the explicit-coordinate interface with numeric `coordt`.

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

`varcov` Covariance matrix of the estimated parameters. It is required for `type_krig="Universal"` when `mse=TRUE`. For composite-likelihood fits this is the sandwich covariance, i.e. the inverse Godambe information, usually taken automatically from a `GeoFit` object. Mean coefficients absent from `varcov` are treated as fixed.

`progress` Logical; if `TRUE`, show progress while the optimized blocked prediction path processes multiple prediction blocks. The default is `FALSE`. No progress bar is shown when the calculation uses a single block or a legacy one-shot prediction path.

`check.duplicates`
Logical. If `TRUE`, perform a fast exact scan for duplicated observation locations at this user entry point. The default is `FALSE`, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

Details

For univariate kriging, the mean at the observations is either $X\beta$, where the entries of β are represented by `mean`, `mean1`, and so on, or a site-specific vector supplied as `param$mean`. With `X=NULL`, the model is intercept-only and uses the scalar coefficient `mean`.

The prediction mean is specified independently: use `Xloc` to obtain $X_{loc}\beta$, or use `Mloc` to supply the prediction mean directly. `Mloc` and `Xloc` are mutually exclusive. If the observation mean is site-specific, `Mloc` is required. When `X` is used at the observations, either `Xloc` or `Mloc` must be supplied; the latter allows a known prediction mean that is not generated by the fitted design.

For `type_krig="Universal"`, prediction uses the same fitted mean coefficients as `type_krig="Simple"`. Thus coefficients estimated by maximum likelihood, REML, or composite likelihood are not recomputed inside `GeoKrig`. When `mse=TRUE`, the MSE is augmented by $(X_{loc} - \lambda^T X)\, Var(\widehat{\beta})\, (X_{loc} - \lambda^T X)^T$, using the mean-parameter block of `varcov`. For composite likelihood, `varcov` should be the sandwich covariance (the inverse Godambe information). Coefficients absent from that matrix are regarded as fixed and contribute zero uncertainty. External mean vectors supplied through `param$mean` or `Mloc` are already known, so a requested universal prediction is treated as simple kriging with a warning.

For prediction, names beginning with `Gaussian_misp_` are interpreted as estimation specifications rather than new data-generating models. They are automatically mapped to the corresponding response model before covariance and prediction calculations (for example `Gaussian_misp_Poisson` to `Poisson`). Without a copula, prediction is available only for models with a validated model-specific covariance/predictor implementation. In particular, plain `Logistic`, `Beta2`, `Kumaraswamy`, and `Kumaraswamy2` are rejected rather than falling through an incomplete branch; these margins are available in the validated continuous-copula prediction path.

For discrete models with nonlinear marginal means, the Universal-kriging MSE uses the derivative of the marginal mean with respect to the regression predictor when propagating the mean-parameter block of `varcov`. The MSE is evaluated from its diagonal directly and does not construct a prediction-by-prediction matrix.

For `copula="Gaussian"`, `copula="Clayton"`, and `copula="SkewGaussian"`, prediction is performed on the observed marginal scale using marginal means, marginal variances, and the copula-induced cross-covariances. The supported continuous margins are `"Gaussian"`, `"StudentT"`, `"LogGaussian"`, "Gamma", "Weibull", "Beta", "Beta2", "Kumaraswamy", "Kumaraswamy2", "Logistic", "SkewLaplace", "Tukeyh", "Tukeyh2", and "SinhAsinh". The count margins "Poisson", "Binomial", and "BinomialNeg" are also supported by global GeoKrig for Gaussian, Clayton-like, and skew-Gaussian copulas; the Binomial copula path currently requires a common trial count n. For a Gaussian copula, the last three margins use their exact transformed-Gaussian covariance rather than a truncated Hermite approximation. For location-dependent margins such as "Beta2", "Kumaraswamy2", and the positive-scale models, the covariance between two sites uses both supplied location predictors. Thus X/Xloc or external means are propagated into the covariance consistently. The resulting predictor is the optimal linear predictor on the observed scale; except when the joint field is Gaussian it is generally different from the full conditional mean. For type_krig="Universal", the MSE correction uses the derivative of the marginal mean with respect to the regression predictor (delta-method correction for nonlinear marginal means). For the Clayton-like and skew-Gaussian copulas, the expensive covariance ingredients are precomputed and cached. Clayton-like prediction uses deterministic copula quadrature tables, while skew-Gaussian prediction uses a cached bivariate Hermite expansion of the latent Gaussian representation; neither path performs adaptive two-dimensional integration separately for each covariance entry. For copula="SkewGaussian", param$nu is the paper's $\eta \in (-1, 1)$; when nu=0 the covariance calculation is routed exactly to the Gaussian-copula implementation.

For the Clayton-like copula, direct adaptive two-dimensional integration for every covariance entry would be prohibitively expensive. GeoModels therefore uses a deterministic cached covariance engine: the Clayton-like copula density is evaluated once on a Gauss–Legendre quadrature grid over a compact grid of the underlying squared correlations, the discrete joint distributions are numerically balanced to preserve uniform margins, and covariance values are obtained by cached matrix products and interpolation. The expensive copula kernel is reused across prediction locations and across calls in the same R session. Positive-scale margins exploit their exact multiplicative scaling, while "Beta2" and "Kumaraswamy2" use an additional compact cached grid on the location-predictor scale, built only over the range required by the current kriging problem. This approximation is designed for kriging covariance construction and avoids one numerical double integral per pair.

For global prediction with a supported continuous copula, GeoKrig() constructs observation–prediction cross-covariances in bounded blocks of prediction locations. The observation covariance matrix is factorized once and the same Cholesky (or SVD, when requested) factorization is reused for every block. This changes only the evaluation order: predictions, MSE values, and kriging weights use the same covariance equations as the one-shot calculation. The historical full weights matrix is retained in the returned object, so blocking reduces the peak memory associated with the cross-covariance matrix without changing the returned results. The ordinary Gaussian model without a copula uses the same blocked path. When progress=TRUE and more than one block is required, progress is updated once per completed prediction block; the numerical calculation and block sizes are unchanged.

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 or universal 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 Binomial number of trials, or the common Negative-Binomial number r of successes.
- `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 response model used for prediction after canonicalizing any misspecified-Gaussian fitting name.
- `fit_model` — The model name supplied by the caller or stored in the input `GeoFit` object before prediction canonicalization.
- `param` — The original parameter list supplied to prediction, before any internal standardization used to construct covariance matrices;
- `pred` — For spatio-temporal prediction, a $m \times n$ matrix with rows corresponding to `time` and columns corresponding to rows of `loc`; for spatial prediction, a numeric vector.
- `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` — When `mse=TRUE`, the prediction MSE (a time-by-location matrix for space-time prediction and a vector for spatial prediction); otherwise `NULL`.

Spatio-temporal ordering

Observed fixed-location data use time-major order: a $T \times N$ matrix is vectorized as `c(t(data))`. Dynamic observations are supplied as aligned lists `data[[t]]` and `coordx_dyn[[t]]`, concatenated by time. The rows of `X` and any known observation mean use the same observation order.

Prediction tasks use the different, location-major order

```
loc[1, ] at time[1], ..., loc[1, ] at time[Tloc],
loc[2, ] at time[1], ..., loc[2, ] at time[Tloc], ...
```

Rows of `Xloc` and elements of `Mloc` must follow this order. The returned `pred` and `mse` objects are $Tloc \times Nloc$ matrices with prediction times in rows and prediction locations in columns. See `GeoModels-spacetime-ordering`.

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*. Springer-Verlag, New York.

See Also

`GeoCovmatrix` for covariance matrix construction, `GeoKrigloc` and `GeoKriglocWeights` for local kriging, `GeoFit` for parameter estimation.

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))
if (requireNamespace("fields", quietly = TRUE)) {
 fields::quilt.plot(coords, data, col = colour)
}
# simple kriging map prediction
if (requireNamespace("fields", quietly = TRUE)) {
 fields::image.plot(
  xx, xx, matrix(pr$pred, ncol = length(xx)), col = colour,
  xlab = "", ylab = "", main = " Kriging "
 )
}

# simple kriging MSE map prediction variance
if (requireNamespace("fields", quietly = TRUE)) {
 fields::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))
if (requireNamespace("fields", quietly = TRUE)) {
 fields::quilt.plot(coords, data, col = colour)
```

```
}
# simple kriging map prediction
if (requireNamespace("fields", quietly = TRUE)) {
 fields::image.plot(
  xx, xx, matrix(pr$pred, ncol = length(xx)), col = colour,
  xlab = "", ylab = "", main = " Kriging "
 )
}

# simple kriging MSE map prediction variance
if (requireNamespace("fields", quietly = TRUE)) {
 fields::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))
if (requireNamespace("fields", quietly = TRUE)) {
 fields::quilt.plot(coords, data, main = "Data")
}
map=matrix(pr$pred,ncol=length(xx))
mapmse=matrix(pr$mse,ncol=length(xx))
if (requireNamespace("fields", quietly = TRUE)) {
 fields::image.plot(xx, xx, map, xlab = "", ylab = "", main = "Kriging ")
}

if (requireNamespace("fields", quietly = TRUE)) {
 fields::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)


if (requireNamespace("fields", quietly = TRUE)) {
 fields::quilt.plot(
  coords, data[2, ], col = colour, main = " data at Time 2"
 )
}
if (requireNamespace("fields", quietly = TRUE)) {
 fields::image.plot(
  xx, xx, matrix(pr$pred, ncol = length(xx)), col = colour,
  main = " Kriging at Time 2", ylab = ""
 )
}
if (requireNamespace("fields", quietly = TRUE)) {
 fields::image.plot(
  xx, xx, matrix(pr$mse, ncol = length(xx)), col = colour,
  main = "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)
#fields::quilt.plot(coords,data[1,] ,col=colour,main = paste(" Fist component"))
```

```
#fields::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 random fields.*

Description

For a given set of spatial location sites (and temporal instants), the function computes optimal 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=NULL, 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=6,progress=TRUE,
 check.duplicates=FALSE)
```

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 spatio-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatio-temporal realisation on regular grid) giving the data used for prediction.
`coordx`	A numeric $d \times 2$ or $d \times 3$ coordinate matrix. It may be omitted when `coordx_dyn` or `spobj` is supplied. 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 the temporal coordinates of the observations. The default is NULL, in which case a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
coordx_dyn	For dynamic observation locations, a list with one two- or three-column coordinate matrix per element of coordt. The rows of coordx_dyn[[t]] correspond to data[[t]] and to the same temporal block of X. Dynamic observation locations do not alter the ordering of prediction tasks. See GeoModels-spacetime-ordering.
corrmodel	String; the name of a correlation model, for the see GeoCovmatrix for the list of implemented correlation models.
distance	String; the name of the spatial distance. The default is Eucl, the Euclidean distance. See GeoFit for details.
grid	Logical; if FALSE (the default) the data used for prediction are interpreted as spatial or spatio-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 non-negative maximum temporal-distance threshold, expressed in the same units as coordt, used for the local temporal neighborhood.
method	String; matrix decomposition used to solve each local kriging system. The choices are cholesky (default) and svd. Only cholesky is available when sparse=TRUE.
n	Positive integer size parameter. For Binomial it may be scalar or contain one value per observation. For direct Negative Binomial it is the single common number r of successes; Geometric corresponds to $r = 1$. Default is 1.
nloc	Positive integer size at prediction tasks. For Binomial, site-specific observation sizes require explicit prediction sizes. For direct Negative Binomial, nloc, if supplied, must equal the common r; for Geometric it must equal 1. No prediction size is inferred by averaging observation-side values.
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 Details for the accepted options.
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 vector giving the temporal instants to be predicted. Values need not be equally spaced and are interpreted on the same numeric time scale as `coordt`. The default is `NULL`, in which case only spatial prediction is performed.

`type` String; currently only `Standard` is computed. Other legacy values produce a warning and are treated as `Standard`.

`type_krig` String; `Simple` (default) treats fitted mean coefficients as known. `Universal` uses the same coefficients and adds their uncertainty from `varcov` to each local prediction MSE.

`weigthed` Logical legacy argument retained for backward compatibility. It has no effect for `type="Standard"`.

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

`copula` String; optional copula specification. Local linear prediction is implemented for `"Gaussian"`, `"Clayton"`, and `"SkewGaussian"`.

`X` Numeric design matrix at the observations. For fixed-location space-time data, rows follow `c(t(data))`, i.e. time then site. For dynamic observations, use a stacked matrix ordered by temporal blocks, or a list aligned with `coordx_dyn`.

`Xloc` Numeric design matrix at prediction tasks. For space-time prediction its rows are location-major: all requested times for `loc[1, ]`, then all requested times for `loc[2, ]`, and so on.

`Mloc` Numeric vector giving the known marginal location predictor at prediction tasks, in the same location-major order as `Xloc`. For Gaussian additive margins this is the mean itself; for models with link-scale parametrizations (for example Gamma/Weibull/LogGaussian or Beta2) it is on the same scale as `param$mean`. Use either `Mloc` or `Xloc`, not both.

`varcov` Covariance matrix of the estimated parameters, required for `type_krig="Universal"` when `mse=TRUE`. With a composite fit it should be the sandwich covariance (inverse Godambe information). It is normally inherited from `estobj`.

`spobj` An object of class `sp` or `spacetime`. For space-time objects, the current `sp2Geo()` conversion uses sequential temporal indices and does not preserve irregular original time spacing; to retain irregular temporal distances, use the explicit-coordinate interface with numeric `coordt`.

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

`parallel` Logical; default `FALSE`. If `TRUE`, the independent local `GeoKrig` calls are evaluated with **future.apply**.

`ncores` Numeric or `NULL`; default `6`. With `parallel=TRUE`, a numeric value requests that many worker processes, capped only by the number of prediction tasks and detected cores. Thus the default parallel execution uses up to six workers. Set `ncores=NULL` to request the automatic/safe mode, in which the package-wide automatic cap of six workers and the internal RAM safety check may further reduce the worker count.

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

`check.duplicates`
Logical. If `TRUE`, perform a fast exact scan for duplicated observation locations at this user entry point. The default is `FALSE`, so no duplicate-location scan is

imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

Details

The univariate mean convention is the same as in GeoKrig: $\mu = X\beta$ with coefficients named mean, mean1, and so on, or an external observation mean vector in param$mean. The intercept-only case is equivalent to a one-column matrix of ones. At prediction tasks, use either Xloc for $X_{loc}\beta$ or Mloc for a directly supplied local mean. The rows of Xloc and the entries of Mloc follow the local-task order; for fixed-location space-time kriging this is location first and prediction time second.

For type_krig="Universal", each local prediction uses the mean coefficients already supplied or estimated by GeoFit; they are not re-estimated by GLS inside each neighborhood. If mse=TRUE, the local MSE includes the uncertainty of those coefficients using the corresponding block of varcov. For composite likelihood this is the sandwich covariance (inverse Godambe information). Mean coefficients missing from varcov are treated as fixed. External mean vectors are already known, so a requested universal prediction is treated as simple kriging with a warning.

For copula="Gaussian", copula="Clayton", and copula="SkewGaussian", each local prediction uses the same observed-scale copula covariance, marginal-mean calculations, solver, and MSE implementation as GeoKrig. The local predictor is an optimal linear predictor, not in general the full conditional mean.

GeoKrigloc uses a single computational path. First, GeoNeighborhood constructs the requested neighborhood for each prediction task. Then GeoKrig is called on each local data set. This keeps covariance, copula, mean, numerical-solver, and MSE logic centralized in GeoKrig rather than duplicating those calculations inside GeoKrigloc.

With parallel=TRUE, independent local GeoKrig calls are submitted through **future.apply**. The default ncores=6 requests up to six workers, subject only to the number of prediction tasks and detected cores. Any other numeric ncores value is treated in the same way and is not reduced by the RAM safety heuristic. Set ncores=NULL explicitly to select the automatic/safe mode, where the package-wide worker resolver and the internal RAM safety check may reduce the worker count. On platforms where **future** reports that forked workers are safe, the automatic backend uses multicore; otherwise it uses multisession. Parallel and serial execution use the same local neighborhoods and the same GeoKrig calculations.

When geometric anisotropy is supplied, observation coordinates, prediction locations, and neighborhood selection are transformed by the same metric. Thus the selected local neighbors are the nearest points under the covariance metric, not under the original isotropic distance.

This function uses GeoKrig with a spatial or spatio-temporal neighborhood computed using GeoNeighborhood. Each local prediction task is evaluated through the same GeoKrig implementation used by the non-local prediction interface. The neighborhood is specified with neighb, maxdist, and maxtime. Regular-grid data are internally vectorized in the same order as expand.grid; fixed-location space-time observations remain in time-major order.

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 Binomial number of trials, or the common Negative-Binomial number r of successes.
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 response model used for prediction after canonicalizing any misspecified-Gaussian fitting name.
fit_model	The model name supplied by the caller or stored in the input `GeoFit` object before prediction canonicalization.
param	The parameter list used for prediction;
pred	For spatio-temporal prediction, a $m \times n$ matrix with rows corresponding to `time` and columns corresponding to rows of `loc`; for spatial prediction, a numeric vector.
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	When `mse=TRUE`, the prediction MSE; otherwise `NULL`.

Spatio-temporal ordering

Observed fixed-location data use time-major order: a $T \times N$ matrix is vectorized as `c(t(data))`. Dynamic observations are supplied as aligned lists `data[[t]]` and `coordx_dyn[[t]]`, concatenated by time. The rows of `X` and any known observation mean use the same observation order.

Prediction tasks use the different, location-major order

```
loc[1, ] at time[1], ..., loc[1, ] at time[Tloc],
loc[2, ] at time[1], ..., loc[2, ] at time[Tloc], ...
```

Rows of `Xloc` and elements of `Mloc` must follow this order. The returned `pred` and `mse` objects are $Tloc \times Nloc$ matrices with prediction times in rows and prediction locations in columns. See `GeoModels-spacetime-ordering`.

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*. Springer-Verlag, 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)
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)
#fields::quilt.plot(coords,data[1,] ,col=colour,main = paste(" Fist component"))
#fields::quilt.plot(loc_to_pred,pr1$pred,col=colour,
# main = paste(" Kriging first component"),ylab="")
#par(opar)
```

GeoKriglocWeights *Compute validated local Gaussian kriging weights*

Description

Given a set of spatial locations (and possibly temporal instants), the function computes local Gaussian kriging weights for each requested prediction task using a neighborhood selected from the observations. Non-Gaussian local prediction is implemented by GeoKrigloc, not by this low-level weights helper.

Usage

```
GeoKriglocWeights(coordx=NULL, 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=FALSE,
 ncores=6, compact=FALSE)
```

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 containing the temporal coordinates of the observations. If missing, a purely spatial random field is assumed. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
coordx_dyn	List of one two- or three-column observation-coordinate matrix per temporal instant. Blocks are concatenated by time; rows within each block retain their matrix order. See GeoModels-spacetime-ordering.
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 non-negative maximum temporal-distance threshold, expressed in the same units as coordt, used for the local temporal neighborhood.
method	Character string indicating the matrix factorisation used to solve the kriging system: "cholesky" (default) or "svd".
model	Character string. The validated public implementation currently requires "Gaussian" (the alias "Gauss" is accepted).
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 giving the temporal instants for which weights are required. Values need not be equally spaced and are interpreted on the same numeric time scale as coordt. Ignored if coordt is missing.
which	Integer (1 or 2) selecting the variable whose weights are returned in the bivariate case.
copula	Must be NULL.
X	Numeric design matrix at observation locations. Fixed-location observations are ordered time then site; dynamic observations are ordered by the temporal blocks of coordx_dyn.

Xloc	Numeric design matrix at prediction tasks, ordered location then time: all requested times for the first row of loc, then all requested times for the second row, and so on.
Mloc	Numeric vector of known prediction means in the same location-major order as Xloc. Use either Mloc or Xloc, not both.
parallel	Logical; default FALSE. Set TRUE to compute independent local kriging systems in parallel where supported.
ncores	Positive integer or NULL; default 6. With parallel=TRUE, an explicit integer requests that many workers, capped by detected cores and the number of available jobs. Set ncores=NULL for automatic selection, capped at six workers and normally leaving one detected core free.
compact	Logical. If FALSE (default), each local element contains the full GeoKrigWeights result for backward compatibility. If TRUE, each non-empty element keeps only the numeric weights and neighbor_indices; this memory-saving representation is used internally by GeoSimcond(local=TRUE).

Details

Mean inputs follow GeoKrigloc. Coefficients mean, mean1, and so on correspond in order to the columns of X; an intercept-only model uses only mean. A vector param$mean is a known observation mean and is mutually exclusive with X. At local prediction tasks, use either Xloc or Mloc, not both.

For every prediction task the function builds and solves the local kriging system

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

, where Σ is the Gaussian covariance matrix among the selected observations and $\boldsymbol{\sigma}_0$ contains their covariances with the prediction location. No actual prediction is carried out; for model-specific local prediction use GeoKrigloc. Local results are returned directly rather than through temporary-file references, so their components remain available after the parallel workers have terminated.

Value

An object of class GeoKriglocWeights. Its component weights is a list ordered by prediction task. With compact=FALSE, each non-NULL element is the local result returned by GeoKrigWeights, augmented with neighbor_indices. With compact=TRUE, each non-NULL element contains only weights and neighbor_indices. An element is NULL when the requested neighborhood contains no usable observations. For purely spatial and bivariate models there is one list element per row of loc; for space-time models the order is location then requested time. The remaining components record the coordinates, model, parameters, neighborhood settings, and function call used to construct the local systems.

Spatio-temporal ordering

Observation-level rows follow time-major order. For fixed sites this is the order of c(t(data)); for dynamic sites it is the row-binding of coordx_dyn by list element. Prediction tasks follow location-major order, so rows of Xloc and elements of Mloc enumerate all requested times for the first prediction location before moving to the next location. See GeoModels-spacetime-ordering.

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

`GeoKrigloc` for local kriging prediction, `GeoKrigWeights` for global kriging weights, `GeoCovmatrix` for covariance matrix construction.

`GeoKrigWeights` *Compute validated Gaussian kriging weights for spatial and spatio-temporal random fields*

Description

Given a set of spatial locations (and possibly temporal instants), the function returns Gaussian kriging weights and the covariance quantities used to build the kriging system. Model-specific non-Gaussian prediction is provided by `GeoKrig` and `GeoKrigloc` rather than by this low-level weights helper.

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 containing the temporal coordinates of the observations. If missing, a purely spatial random field is assumed. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.

coordx_dyn	List of one two- or three-column observation-coordinate matrix per temporal instant. Blocks are concatenated by time; rows within each block retain their matrix order. See GeoModels-spacetime-ordering.
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. The validated public implementation currently requires "Gaussian" (the alias "Gauss" is accepted).
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 giving the temporal instants for which weights are required. Values need not be equally spaced and are interpreted on the same numeric time scale as coordt. Ignored if coordt is missing.
which	Integer (1 or 2) selecting the variable whose weights are returned in the bivariate case.
copula	Must be NULL. Copula conditional-simulation code requests these weights on the latent Gaussian scale.
X	Numeric design matrix at observation locations. Fixed-location observations are ordered time then site; dynamic observations are ordered by the temporal blocks of coordx_dyn.
Xloc	Numeric design matrix at prediction tasks, ordered location then time: all requested times for the first row of loc, then all requested times for the second row, and so on.
Mloc	Numeric vector of known prediction means in the same location-major order as Xloc. Use either Mloc or Xloc, not both.

Details

Mean inputs follow GeoKrig. Coefficients mean, mean1, and so on correspond in order to the columns of X; an intercept-only model uses only mean. A vector param$mean is a known observation mean and is mutually exclusive with X. At prediction locations, use either Xloc or Mloc, not both.

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}. The returned weights solve the Gaussian covariance system only. Mean inputs are validated for ordering compatibility, but no non-Gaussian covariance transform is performed here. No actual prediction is carried out; for full Gaussian or non-Gaussian prediction use `GeoKrig`.

Value

A list containing:

- `weights` — Numeric matrix of Gaussian kriging weights. The validated orientation is observations by prediction tasks, as recorded in `weights_orientation`.
- `weights_orientation` — Character string describing the weight-matrix orientation.
- `covmatrix` — Observed Gaussian covariance matrix.
- `CC` — Observation–prediction Gaussian cross-covariance matrix.
- `model` — The canonical model name (currently `"Gaussian"`).
- `corrmodel` — Input correlation model.
- `param` — Parameters used to construct the Gaussian covariance matrix.
- `bivariate` — Logical indicating a bivariate Gaussian correlation model.
- `spacetime` — Logical indicating a space-time correlation model.
- `loc` — Prediction locations.
- `tloc` — Number of requested prediction times.
- `Xloc`, `Mloc`, `mean_obs` — Normalized mean inputs retained for alignment and conditional-simulation workflows.

Spatio-temporal ordering

Observation-level rows follow time-major order. For fixed sites this is the order of `c(t(data))`; for dynamic sites it is the row-binding of `coordx_dyn` by list element. Prediction tasks follow location-major order, so rows of `Xloc` and elements of `Mloc` enumerate all requested times for the first prediction location before moving to the next location. See `GeoModels-spacetime-ordering`.

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
```

GeoModels-3D *Three-dimensional spatial coordinates in GeoModels*

Description

Conventions and current limitations for purely spatial models whose locations have three Cartesian coordinates.

Details

For an irregular three-dimensional spatial data set, supply either an $N \times 3$ numeric matrix in `coordx`, or three equal-length vectors through `coordx`, `coordy`, and `coordz`. Use `grid = FALSE` and `distance = "Eucl"`.

Euclidean lags use all three coordinates. The principal fitting, covariance, variogram, Cholesky simulation, neighborhood, and global/local kriging paths support this representation.

Regular three-dimensional grids specified by three coordinate axes and `grid = TRUE` are not implemented. Supply their Cartesian product as an explicit $N \times 3$ matrix and use `grid = FALSE`.

For purely spatial univariate fields, spectral turning-bands simulation accepts explicit irregular three-dimensional Euclidean coordinates through `GeoSimapprox(..., method = "TB", grid = FALSE)`. The same TB simulator can be used by `GeoVarest()` and `GeoVarestbootstrap()` for such fits. Three-dimensional TB does not support bivariate or spatio-temporal models, and does not accept `anisopars`; transform coordinates explicitly when needed. CE remains restricted to regular two-dimensional grids. Use `method = "cholesky"` for unsupported three-dimensional cases.

Distances `"Geod"` and `"Chor"` require exactly two columns (longitude and latitude) and cannot be combined with a third coordinate.

For `GeoVariogramDir()`, three-dimensional lag lengths use all three coordinates, while direction classes are horizontal azimuths computed from the x- and y-coordinate differences. This is not a full three-dimensional directional analysis with azimuth and elevation.

See Also

`GeoFit`, `GeoCovmatrix`, `GeoVariogram`, `GeoVariogramDir`, `GeoSim`, `GeoSimapprox`, `GeoKrig`, `GeoKrigloc`

`GeoModels-spacetime-ordering`
Ordering Conventions for Spatio-temporal Data, Covariates and Predictions

---

Description

Defines the ordering used by **GeoModels** for fixed-location and dynamic-location spatio-temporal data, observation-level means and covariates, and kriging prediction tasks.

Details

The conventions below apply consistently to data, coordinates, design matrices, known mean vectors, and prediction outputs.

Irregular temporal coordinates

For the explicit-coordinate space-time interfaces, coordt is a numeric coordinate vector and does not need to be equally spaced. Covariance, likelihood, pair-selection, variogram, neighborhood and kriging calculations use temporal lags derived from the supplied values, typically $|t_i - t_j|$. Consequently, maxtime is a temporal-distance threshold expressed in the same units as coordt, not an index lag or an order of neighboring time points.

Important exceptions are explicit. Space-time GeoSimapprox(..., method = "CE") requires finite, strictly increasing, equally spaced temporal coordinates, and space-time turning-bands simulation is not implemented. GeoSimcond and GeoOutlier currently reject space-time input.

The spobj route has a separate legacy limitation: the current sp2Geo() conversion of STFDF/STIDF objects replaces their time positions by sequential indices 1, 2, ..., T. Therefore an irregular original time spacing is not preserved through spobj. When irregular temporal distances matter, pass the data and coordinates through the explicit interface and supply the numeric coordt values directly.

Fixed spatial locations

Let T be length(coordt) and let N be the number of rows of coordx. The data must be a $T \times N$ matrix. Row t corresponds to coordt[t], and column i corresponds to row i of coordx.

The internal observation order is time-major and is equivalent to c(t(data)):

```
time 1: site 1, site 2, ..., site N
time 2: site 1, site 2, ..., site N
...
time T: site 1, site 2, ..., site N
```

Any observation-level vector, including a known mean vector, must use this order. A design matrix X must have NT rows in the same order. For example, if X_time has one row per time, X_time[rep(seq_len(T), each = N), , drop = FALSE] creates the corresponding observation-level design.

Dynamic spatial locations

For dynamic locations, coordx_dyn must be a list of length T. The element coordx_dyn[[t]] is an $N_t \times 2$ or $N_t \times 3$ matrix containing the locations observed at coordt[t].

The data must be a list of length T; data[[t]] must contain N_t values, and its i-th value corresponds to row i of coordx_dyn[[t]]. The internal order is the time-wise concatenation unlist(data, use.names = FALSE), matched by do.call(rbind, coordx_dyn):

```
all observations at time 1,
then all observations at time 2,
...,
then all observations at time T.
```

A design matrix X can be supplied either as a $(\sum_t N_t) \times p$ matrix in this concatenated order, or as a list of length T with X[[t]] having N_t rows and the same row order as coordx_dyn[[t]]. A known observation-level mean uses the same time-wise concatenation.

Mean parameters

For a univariate linear mean,

$$\mu = X\beta.$$

The regression vector β is represented by the parameters `mean`, `mean1`, `mean2`, and so on. The columns of `X` correspond to these coefficients in exactly this order. An intercept-only model is equivalent to a one-column matrix of ones and the single coefficient `mean`. A supplied observation-level mean vector is a known mean and is not interpreted as regression coefficients.

Kriging prediction order

Let N_{loc} be the number of rows of `loc` and let T_{loc} be `length(time)`. Kriging prediction tasks are ordered location-major:

```
location 1: time 1, time 2, ..., time Tloc
location 2: time 1, time 2, ..., time Tloc
...
location Nloc: time 1, time 2, ..., time Tloc
```

Therefore the rows of `Xloc` and the elements of `Mloc` must follow this order. If `X_time_pred` has one row per prediction time, use `X_time_pred[rep(seq_len(Tloc), times = Nloc), , drop = FALSE]`. Equivalently, if `mu_time` contains one prediction mean per time, use `rep(mu_time, times = Nloc)` for `Mloc`.

Dynamic coordinates describe the observed sites only and do not change the prediction-task order. Spatio-temporal kriging output is a $T_{loc} \times N_{loc}$ matrix: rows correspond to `time` and columns correspond to rows of `loc`.

See Also

`GeoFit`, `GeoSim`, `GeoVariogram`, `GeoKrig`, and `GeoKrigloc`.

Examples

```
T <- 3
N <- 2
data <- matrix(seq_len(T * N), nrow = T, ncol = N)
c(t(data))

X_time <- cbind(1, seq_len(T))
X <- X_time[rep(seq_len(T), each = N), , drop = FALSE]

Nloc <- 2
Xloc <- X_time[rep(seq_len(T), times = Nloc), , drop = FALSE]
Mloc <- rep(seq_len(T), times = Nloc)

## Irregular observation times are allowed in the explicit interface.
coordt_irregular <- c(0, 0.25, 1.1, 2.8)
## Temporal lags are based on differences between these values; for example,
abs(outer(coordt_irregular, coordt_irregular, "-"))
```

`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 spatio-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatio-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 one dimension of temporal coordinates; the default is `NULL`, in which case a spatial random field is expected. Temporal coordinates may be irregularly spaced and their supplied numeric values are preserved.
`coordx_dyn`	For dynamic locations, a list with one two- or three-column coordinate matrix per time. It must be aligned with the list `data`; missing responses remove the corresponding rows from both objects.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as spatial or spatio-temporal realisations on a set of non-equispaced spatial sites (irregular grid).
`X`	Covariates aligned with `data`. For fixed-coordinate space-time data, use an $TN \times p$ matrix ordered as `c(t(data))`. For dynamic data, `X` must be a list with `X[[t]]` having one row per element of `data[[t]]` and per row of `coordx_dyn[[t]]`.
`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	The covariate matrix after deleting the rows corresponding to removed observations. For fixed-coordinate spatio-temporal data, deleting a site removes its row from every temporal block.

Spatio-temporal ordering

For fixed sites, `data` has times in rows and sites in columns; rows of `X` follow time-major order. Removing a site removes the corresponding column of `data`, coordinate row, and its row from every temporal block of `X`. For dynamic sites, `data`, `coordx_dyn`, and `X` are aligned lists and missing values are removed independently within each temporal block. See GeoModels-spacetime-ordering.

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=6,check.duplicates=FALSE)
```

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 spatio-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatio-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 one dimension of the observation-time coordinates. Optional argument; the default is `NULL`, in which case a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
`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 `GeoFit` for details.

grid	Logical; if FALSE (the default) the data are interpreted as spatial or spatio-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 non-negative maximum temporal-distance threshold, expressed in the same units as coordt.
radius	Numeric; a value indicating the radius of the sphere when using the great circle distance. Default value is 1.
time	Numeric; the temporal coordinate at which the neighborhood is requested, interpreted on the same numeric time scale as coordt.
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. For space-time objects, the current sp2Geo() conversion uses sequential temporal indices and does not preserve irregular original time spacing; to retain irregular temporal distances, use the explicit-coordinate interface with numeric coordt.
spdata	Character:The name of data in the sp or spacetime object
parallel	Logical; default FALSE. Set TRUE to enable parallel execution where supported.
ncores	Positive integer or NULL; default 6. With parallel=TRUE, an explicit integer requests that many workers, capped by detected cores and the number of available jobs. Set ncores=NULL for automatic selection, capped at six workers and normally leaving one detected core free.
check.duplicates	Logical. If TRUE, perform a fast exact scan for duplicated observation locations at this user entry point. The default is FALSE, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

Details

Observed locations are checked for exact duplicates before neighborhood construction. In space-time data, the same spatial site may occur at different times, but duplicate space-time observations are not allowed.

Value

Returns a list containing the following information:

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 *Neighborhood Selection for Pairwise Composite Likelihood Estimation*

Description

The procedure fits pairwise composite likelihood models over user-specified spatial or space-time neighborhoods and selects the candidate minimizing the sum of squared differences between the fitted and empirical semivariograms. A GeoVariogram object must be supplied.

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=6,optimizer='Nelder-Mead', parallel=FALSE,
 bivariate=FALSE,radius=1, start=NULL,type='Pairwise', upper=NULL,
 weighted=FALSE,X=NULL,spobj=NULL,spdata=NULL,vario=NULL,progress=TRUE,
 check.duplicates=FALSE)
```

Arguments

data	A d-dimensional vector (a single spatial realisation) or a $(d \times d)$-matrix (a single spatial realisation on regular grid) or a $(t \times d)$-matrix (a single spatio-temporal realisation) or an $(d \times d \times t \times n)$-array (a single spatio-temporal realisation on regular grid). See Details for the accepted data layouts.
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 one dimension of temporal coordinates. Optional argument; the default is `NULL`, in which case a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
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 see `GeoCovmatrix` for the list of implemented correlation models.
distance	String; the name of the spatial distance. The default is `Eucl`, the Euclidean distance. See Details for the accepted options.

`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 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 `GeoFit` for details).
`lower`	An optional named list giving the values for the lower bound of the space parameter when the optimizer is `L-BFGS-B` or `nlminb` or `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 `GeoFit` for details).
`p_neighb`	Numeric; a value in $(0, 1]$ specifying the expected fraction of nearest-neighbor pairs retained through stochastic thinning. If equal to 1 (the default), no thinning is applied. Values below 1 use independent Bernoulli thinning, so the selected neighborhood depends on the thinning realization. Use a fixed random seed, or `p_neighb=1`, when a stable comparison is required.
`maxtime`	Numeric vector of non-negative maximum temporal-distance thresholds, expressed in the same units as `coordt`, considered when selecting the neighborhood configuration.
`memdist`	Deprecated logical argument retained for backward compatibility. The selected pair structure is always precomputed and reused. Supplying `FALSE` produces a warning and is treated as `TRUE`.
`model`	String; the type of random fields and therefore the densities associated to the likelihood objects. `Gaussian` is the default, see `GeoFit` for details.
`n`	Numeric; number of trials in a binomial random fields; number of successes in a negative binomial random fields
`ncores`	Positive integer or `NULL`; default `6`. With `parallel=TRUE`, an explicit integer requests that many workers, capped by detected cores and the number of candidate fits. Set `ncores=NULL` for automatic selection, capped at six workers and normally leaving one detected core free.
`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; default `FALSE`. Set `TRUE` to evaluate candidate fits in parallel using a `future` plan.
`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 `GeoFit` for details).
type	String; the type of the likelihood objects. If `Pairwise` (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods (see `GeoFit` for details).
upper	An optional named list giving the values for the upper bound of the space parameter when the optimizer is or `L-BFGS-B` or `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 `GeoFit` for details). If `FALSE` (the default) the composite likelihood is not weighted.
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
vario	An object of class `GeoVariogram` obtained using `GeoVariogram`. Its spatial, bivariate, or space-time structure must match the selected correlation model.
progress	Logic; If TRUE then a progress bar is shown.
check.duplicates	Logical. If `TRUE`, perform a fast exact scan for duplicated observation locations at this user entry point. The default is `FALSE`, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

Details

For a spatial univariate model, the criterion compares the fitted and empirical spatial semivariograms at the empirical bin centers. For a bivariate model, the two marginal semivariograms and the cross-semivariogram are compared in the order $(11, 12, 22)$. For a space-time model, the spatial and temporal margins and the complete rectangular space-time semivariogram surface are compared jointly. Only finite empirical cells contribute to the criterion; a candidate is rejected when its corresponding fitted values are non-finite or dimensionally incompatible.

The absolute minimum defines the best candidate. The function also reports a more parsimonious neighborhood using the existing plateau rule: the smallest neighborhood whose criterion is within 15 percent of the minimum. If every candidate fit fails or produces a non-finite criterion, the function stops with an error instead of selecting an arbitrary candidate.

Value

A list with components:

- `best_neighb`: neighborhood associated with the smallest criterion;
- `best_maxtime`: selected temporal neighborhood for a space-time model, otherwise `NULL`;
- `best_estimates`: named parameter estimates for the best candidate;

- `res`: variogram sum-of-squares criterion for every candidate;
- `estimates`: matrix of parameter estimates, with columns aligned to the names in `start`;
- `sugg_neighb`: parsimonious neighborhood selected by the 15 percent plateau rule;
- `sugg_time`: corresponding temporal neighborhood for a space-time model, otherwise `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

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, spatio-temporal, or bivariate nearest-neighbour indices*

Description

Builds directed nearest-neighbour candidate pairs for spatial, spatio-temporal, or bivariate data. Candidate pairs can optionally be thinned using independent Bernoulli sampling or an exact fixed-budget, stratum-wise design.

Usage

```
GeoNeighIndex(coordx=NULL, 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="bernoulli", check.duplicates=FALSE)
```

Arguments

coordx	A numeric $N \times 2$ or $N \times 3$ coordinate matrix. Longitude/latitude coordinates can be supplied for spherical distances. It may be omitted when `coordx_dyn` is supplied.
coordy	Optional numeric vector giving a second spatial coordinate when coordinates are supplied separately.
coordz	Optional numeric vector giving a third spatial coordinate when coordinates are supplied separately.
coordt	A numeric vector giving the temporal coordinates. Optional argument, default is `NULL`; if `NULL`, a purely spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
coordx_dyn	A list of numeric coordinate matrices. For spatio-temporal data, the list must contain one coordinate matrix per element of `coordt`. For bivariate data with different spatial supports, it must contain the two variable-specific coordinate matrices. Optional argument, default is `NULL`.
distance	Character string specifying the spatial distance. Default is `"Eucl"`. See `GeoFit`.
neighb	A positive integer giving the nearest-neighbour candidate size. In the bivariate case, a length-three positive integer vector can be supplied for within-variable 1, cross-variable, and within-variable 2 pairs.
maxdist	Optional non-negative maximum spatial distance. In the bivariate case, a length-three vector can be supplied for the three pair types.
maxtime	A non-negative numeric value giving the maximum temporal distance in the same units as `coordt`. Spatio-temporal candidate pairs satisfy the corresponding temporal-distance restriction; see Details.

radius
: Numeric radius used for spherical distances. Default is 1.

bivariate
: Logical; if FALSE (default), construct univariate spatial or spatio-temporal pairs. If TRUE, construct bivariate spatial pairs.

p_neighb
: Numeric scalar in $(0, 1]$. If 1, the candidate set is returned without thinning. For thin_method="bernoulli", it is the marginal inclusion probability in the constant-probability design and hence controls the retained size in expectation. For thin_method="FixedBudget", it defines the exact global retained-pair budget $K = \mathrm{round}(p_{neighb} d)$, where d is the candidate pair count.

thin_method
: Character string selecting the thinning design. Accepted values are "bernoulli" and "FixedBudget", case-insensitively. The legacy name "TargetBalanced" is accepted as a backward-compatible alias for "FixedBudget".

check.duplicates
: Logical. If TRUE, perform a fast exact scan for duplicated observation locations at this user entry point. The default is FALSE, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

Details

The unthinned output is a directed candidate graph: rowidx identifies candidate targets and colidx their neighbours. For a purely spatial configuration with N locations and candidate size m, the regular case contains approximately Nm directed candidate pairs.

For chordal and geodesic distances, nearest-neighbour ordering is built in three-dimensional unit-sphere coordinates. Because chordal distance is monotone in great-circle distance, this gives the correct global nearest-neighbour order, including near the dateline and poles. Reported lags are then evaluated in the requested spherical metric and scaled by radius.

For spatio-temporal data, temporal coordinates are treated as numeric coordinates, not as equally spaced indices. Candidate temporal separations are computed as

$$|t_i - t_j|,$$

and maxtime is interpreted as a temporal-distance threshold in the same units as coordt. Thus irregularly spaced temporal coordinates are supported directly.

With thin_method="bernoulli" and p_neighb < 1, candidate pairs are retained independently. In the current constant-weight implementation the inclusion probability is p_neighb, so the retained count is random with expectation approximately $p_{neighb} d$.

With thin_method="FixedBudget" and p_neighb < 1, exactly $K = \mathrm{round}(p_{neighb} d)$ candidate pairs are retained, subject only to truncation to the available candidate count. The budget is allocated across natural candidate strata using proportional quotas with randomized residual allocation, followed by sampling without replacement within each stratum. For purely spatial data, strata are the target-specific nearest-neighbour lists. For spatio-temporal data, target and temporal-lag descriptors define the strata; for bivariate data, the variable-pair descriptors are also included. The resulting global retained size is exact.

For thin_method="FixedBudget", the exact target is $round(pd)$. Hence very small p_neighb values may legitimately return zero retained pairs. This is a valid graph-selection result; GeoFit rejects such a graph because a pairwise likelihood cannot be estimated without pair contributions. Exact duplicate observation locations are rejected before nearest-neighbor construction.

When p_neighb = 1, no thinning is applied, independently of thin_method.

Value

A list containing the pair indices and distances required by downstream GeoModels procedures. Depending on the setting, components include:

`rowidx`	Directed target indices.
`colidx`	Directed neighbour indices.
`lags`	Spatial distances for the retained pairs.
`lagt`	Temporal distances for spatio-temporal pairs.
`first`	First variable indicator for bivariate pairs.
`second`	Second variable indicator for bivariate pairs.
`maxdist`	Spatial-distance threshold, when stored for the selected case.
`neighb`	Nearest-neighbour candidate size, when stored for the selected case.

The function returns only the retained pair structure; it does not append summary fields such as candidate or retained counts. These can be obtained from the lengths of the returned index vectors.

Author(s)

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

Examples

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

## Full directed 5-NN candidate graph.
sel <- GeoNeighIndex(coordx=coords, neighb=5)
length(sel$rowidx)

## Bernoulli thinning: retained size is random with expectation 20 percent
## of the candidate graph.
set.seed(1)
sel_ber <- GeoNeighIndex(coordx=coords, neighb=5,
 p_neighb=0.2, thin_method="bernoulli")
length(sel_ber$rowidx)

## Fixed-budget thinning: the global retained size is exact.
set.seed(1)
sel_fix <- GeoNeighIndex(coordx=coords, neighb=5,
 p_neighb=0.2, thin_method="FixedBudget")
length(sel_fix$rowidx)

## Irregular temporal coordinates: maxtime is a distance threshold.
times <- c(0, 0.25, 1.1, 2.8)
sel_st <- GeoNeighIndex(coordx=coords[1:30, ], coordt=times,
```

```
 neighb=3, maxtime=1)
head(sel_st$lagt)
```

GeoOutlier *Spatial outlier detection*

Description

Detects outliers in a univariate purely spatial data set using the selected local procedure. Space-time and bivariate outlier detection are not currently implemented.

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	A numeric vector containing a single spatial realization, or a matrix representing a single spatial realization on a regular grid. Space-time data are not currently supported.
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	Retained for API compatibility. GeoOutlier is currently implemented only for purely spatial data and rejects space-time input.
coordx_dyn	Retained for API compatibility. Dynamic space-time coordinates are not currently implemented.
distance	String; the name of the spatial distance. The default is Eucl, the Euclidean distance. See GeoFit for details.
grid	Logical; if FALSE (the default), the data are interpreted as a spatial realization on explicit, possibly non-equispaced, spatial sites.
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	Retained for API compatibility. Bivariate outlier detection is not currently implemented and TRUE is rejected.
X	Numeric; an optional matrix of spatial covariates with one row per observation.

Value

Returns a matrix containing the detected spatial outliers

Three-dimensional coordinates

Purely spatial irregular coordinates may be supplied as an $N \times 3$ matrix or as separate `coordx`, `coordy`, and `coordz` vectors. Three-dimensional coordinates require `distance = "Eucl"`. Space-time and bivariate outlier detection are not implemented.

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

`GeoPit` *Probability Integral Transform for Fitted GeoModels Objects*

Description

Transforms observations or prediction targets using the fitted marginal cumulative distribution function. The output can be represented on the uniform probability-integral-transform scale or on the standard Gaussian-score scale.

Usage

```
GeoPit(object, type = c("Uniform", "Gaussian"), data_to_pred = NULL)
```

Arguments

`object`	An object of class `GeoFit`, `GeoKrig`, or `GeoKrigloc`. The spatial bivariate case is not currently supported.
`type`	Character string. `"Uniform"` returns fitted PIT values $F_i(Y_i)$. `"Gaussian"` returns $\Phi^{-1}\{F_i(Y_i)\}$. Partial matching is performed by `match.arg`.
`data_to_pred`	Optional observed values at the prediction locations when `object` is a `GeoKrig` or `GeoKrigloc` object. This is required unless the object already contains a `data_to_pred` component.

Details

For a `GeoFit` object, marginal transformations use the complete fitted mean: a site-specific fixed vector when present, otherwise $X\beta$ with coefficients `mean`, `mean1`, and so on matched to the columns of `X`. The intercept-only case uses scalar `mean`.

For a `GeoFit` object, site-specific fitted means are reconstructed from the design matrix and fitted regression coefficients. An externally supplied fixed mean vector takes precedence. The fitted marginal CDF is then evaluated at each observation. Copula parameters do not enter the PIT because the transformation is marginal. The Student-t and skew Student-t transformations use the fitted degrees of freedom $1/df$ without integer rounding. The continuous PIT implementation includes the bounded Kumaraswamy families, Logistic, and SkewLaplace marginals used by the copula diagnostics.

For `type = "Gaussian"`, PIT values are clipped only for numerical protection before applying the standard-normal quantile function. For discrete marginal models, the current transformation is the ordinary non-randomized PIT and therefore is not exactly uniform. For `PoissonGamma`, the fitted marginal CDF is the negative-binomial CDF with mean `exp(mean)` (or the site-specific fitted mean) and size parameter `shape`. Pearson-residual objects returned by `GeoResiduals()` are rejected; PIT diagnostics for discrete models must use the original fitted object.

For `GeoKrig` and `GeoKrigloc` objects, predictive PIT is currently available only for Gaussian models and requires predictive MSE values. Linear kriging for the non-Gaussian models does not by itself define the full conditional predictive distribution needed for a valid PIT.

Value

The input object with its `data` component replaced by the transformed values.

Examples

```
## Not run:
fit_u <- GeoPit(fit, type = "Uniform")
fit_z <- GeoPit(fit, type = "Gaussian")

## End(Not run)
```

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`	An object of class GeoFit, For fitted models with covariates, the GeoFit object must first be transformed by 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.

Details

This diagnostic requires an intercept-only marginal representation. For continuous models, fits with a nonconstant design matrix or a site-specific fixed mean can first be converted with GeoResiduals. For discrete models with a nonconstant mean, use GeoPit instead, because there is no single common marginal distribution for a conventional QQ plot. Likewise, Binomial and negative-binomial QQ/density plots require a constant `n`; site-specific `n` should be assessed with `GeoPit()`.

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` *Compute Residuals from a Fitted GeoModels Model*

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. For the discrete Bernoulli/Binary, Binomial, Geometric/negative-binomial, Poisson, and Poisson–Gamma models, the function returns Pearson residuals computed from the fitted marginal mean and variance. The paper specifies these marginal moments but does not prescribe a residual definition; Pearson residuals are an explicit package convention. For continuous models, the returned object can be coupled with `GeoQQ` and `GeoCovariogram`. For discrete Pearson residuals, use the returned object for residual maps and variograms, and apply `GeoQQ()` or `GeoPit()` to the original fitted object for marginal assessment.

Usage

```
GeoResiduals(fit)
```

Arguments

`fit`	A fitted object obtained from the `GeoFit`.

Details

The fitted location is evaluated from the full mean specification: a site-specific fixed mean vector when present, otherwise $X\beta$ with coefficients `mean`, `mean1`, and so on matched in order to the columns of X. The returned residual object is converted to an intercept-only, standardized representation for subsequent diagnostics. All other fields of the original GeoFit object, including neighbourhood, thinning, anisotropy, convergence, and pair-selection metadata, are preserved.

For Beta2 and Kumaraswamy2, the residual transformation uses the fitted support min/max and the corresponding fitted marginal CDF, so the standardized residual scale is the unit interval.

For the discrete models described in the package paper, Pearson residuals are used: $(Y-E(Y))/\sqrt{Var(Y)}$. Thus $p = \Phi(\mu)$ for Bernoulli, Binomial and negative-binomial fields, while $\lambda = \exp(\mu)$ for Poisson fields. For Poisson–Gamma fields, with fitted mean λ and shape κ, the fitted variance is $\lambda(1+\lambda/\kappa)$. These deterministic residuals are intended for residual maps and variograms; they are not randomized quantile residuals, and GeoQQ() intentionally rejects the Pearson-residual object. Bivariate residuals are currently supported only for Gaussian models.

Value

Returns an (updated) object of class GeoFit. The component residual_type records whether the data are Pearson or model-standardized residuals.

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 — *Scatterplots of Spatial or Spatio-temporal Pairs and Fitted Bivariate Contours*

Description

Produces scatterplots of observations associated with spatial or spatio-temporal pairs selected by distance classes, nearest-neighbour order, and temporal lag. When the first argument is a fitted `GeoFit` object, the function can optionally superimpose contour lines of the fitted bivariate density on the original response scale, the uniform probability-integral-transform scale, or the Gaussian-score scale.

Usage

```
GeoScatterplot(data, coordx = NULL, coordy = NULL, coordz = NULL,
  coordt = NULL, coordx_dyn = NULL, distance = "Eucl", grid = FALSE,
  maxdist = NULL, neighb = NULL, times = NULL, time.lag = NULL,
  numbins = 4, radius = 1, bivariate = FALSE,
  contour = inherits(data, "GeoFit"),
```

```
residuals = FALSE, scale = c("Original", "Gaussian", "Uniform"),
gaussian.range = 3, ngrid = 80, nlevels = 6, levels = NULL,
contour.col = "red", contour.lwd = 1.5,
contour.labels = FALSE,
point.col = grDevices::adjustcolor("#481567FF", 0.45),
lag.method = c("median", "mean"), ...)
```

Arguments

- `data` Either the data to be plotted or an object of class `GeoFit`. For the original interface, this can be a numeric vector, matrix, or array containing a spatial, spatio-temporal, or bivariate realization. See `GeoFit` for the supported data layouts.
- `coordx` A numeric coordinate vector, a two-column coordinate matrix, or a three-column coordinate matrix. Coordinates on a sphere are supplied in longitude/latitude format, in decimal degrees. This argument is not required when `data` is a `GeoFit` object.
- `coordy` An optional numeric vector containing the second spatial coordinate when `coordx` is supplied as a vector. The default is `NULL`.
- `coordz` An optional numeric vector containing the third spatial coordinate. The default is `NULL`.
- `coordt` An optional numeric vector containing temporal coordinates. If `NULL`, a purely spatial realization is assumed. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
- `coordx_dyn` For dynamic locations, a list with one two- or three-column coordinate matrix per element of `coordt`. In the raw-data interface, `data[[t]]` must align row-by-row with `coordx_dyn[[t]]`. In the `GeoFit` interface this layout is taken from the fitted object.
- `distance` Character string naming the spatial distance. The default is `"Eucl"`. See `GeoFit` for the available options. When `data` is a `GeoFit` object, the distance stored in the fitted object is used.
- `grid` Logical. If `FALSE`, the default, the observations are interpreted as values at non-equispaced sites. If `TRUE`, regular-grid coordinates are generated from the supplied coordinate vectors.
- `maxdist` A positive numeric value defining the maximum spatial distance. When `neighb = NULL`, the interval from zero to `maxdist` is divided into `numbins` distance classes. For a bivariate realization, this can have length one or two. Purely spatial distance-class pairs are generated in memory-bounded blocks rather than by allocating all $n(n-1)/2$ possible pairs at once.
- `neighb` A positive integer or vector of positive integers defining the nearest-neighbour candidate sets used to construct the scatterplots. For a spatio-temporal `GeoFit` object, neighbours are determined for the selected temporal lags. At a positive temporal lag and with fixed spatial locations, the first nearest-neighbour candidate may be the observation at the same spatial location, corresponding to spatial lag zero. More generally, `neighb = m` displays the candidate set formed by the first m nearest neighbours at that temporal lag.

times
: Optional numeric vector selecting temporal instants in the original data-and-coordinate interface. Entries can be values contained in `coordt` or valid one-based indices into `coordt`. At least two instants must be selected. The selected instants are put back in their original temporal order. If `NULL`, all temporal instants are used.

time.lag
: Optional numeric vector of non-negative temporal lags used for spatio-temporal data. The requested values must be obtainable from the selected temporal coordinates. For a fitted `GeoFit` object, if `NULL`, all available temporal lags not exceeding the fitted `maxtime` are used. For the original data-and-coordinate interface, if `NULL`, all temporal lags generated by the selected instants are used. Lag zero is included. Regular temporal grids are handled without constructing a quadratic outer-difference matrix; irregular-grid lags are generated incrementally and numerically equivalent floating-point lags are merged.

numbins
: A positive integer giving the number of distance classes when `neighb = NULL`. The default is 4.

radius
: Numeric value giving the radius of the sphere for great-circle or chordal distances. The default is 1. For a `GeoFit` object, the radius stored in the fitted object is used unless this argument is supplied.

bivariate
: Logical. If `TRUE`, the observations are interpreted as a realization of a bivariate random field. The default is `FALSE`.

contour
: Logical. If `TRUE`, contour lines of the fitted bivariate density are added to each scatterplot. This option requires `data` to be a univariate spatial or spatio-temporal `GeoFit` object. The default is `TRUE` when `data` is a `GeoFit` object and `FALSE` otherwise. Set `contour = FALSE` to obtain only the scatterplots.

residuals
: Logical. Relevant only when `data` is a `GeoFit` object and `scale = "Original"`. If `TRUE`, the scatterplot and fitted density are represented on a residual scale. For real-valued models, the fitted location is subtracted. For `"LogGaussian"`, `"Gamma"`, and `"Weibull"`, the observations are divided by the fitted multiplicative mean. For the direct count models `"Poisson"`, `"Binomial"`, `"BinomialNeg"`/`"Geometric"`, and `"PoissonGamma"`, residual-scale scatterplots use Pearson residuals based on the fitted marginal mean and variance, consistently with `GeoResiduals`. The default is `FALSE`.

scale
: Character string specifying the scale used when `data` is a `GeoFit` object. `"Original"`, the default, uses the response scale. `"Uniform"` applies the fitted marginal distribution function to each observation. `"Gaussian"` additionally applies the standard-normal quantile function. The aliases `"Response"`, `"PIT"`, and `"Normal"` are also accepted. The uniform and Gaussian scales require a fitted continuous marginal model. Pearson-residual objects can be displayed only on the original residual scale.

gaussian.range
: Positive finite number defining the symmetric plotting interval `c(-gaussian.range, gaussian.range)` on the Gaussian-score scale when the user does not supply `xlim` and `ylim`. The default is 3, matching the Gaussian-scale copula comparison plots commonly used to display reflection asymmetry.

ngrid
: Positive integer giving the number of grid points in each coordinate direction used to evaluate the fitted bivariate density. It must be at least 20. The default is 80.

nlevels	Positive integer controlling the number of automatically selected contour levels. Ignored when levels is supplied. The default is 6.
levels	Optional numeric vector of density levels at which contour lines are drawn. The default is NULL, in which case the levels are selected automatically.
contour.col	Colour of the fitted contour lines. The default is "red".
contour.lwd	Line width of the fitted contour lines. The default is 1.5.
contour.labels	Logical indicating whether contour labels are drawn. The default is FALSE.
point.col	Colour used for the scatterplot points in both the raw-data and GeoFit interfaces. The default is a semi-transparent purple. A graphical col supplied through ... takes precedence.
lag.method	Character string specifying the representative spatial lag used for a nearest-neighbour panel. "median", the default, uses the median distance of the displayed pairs; "mean" uses their mean distance. For distance classes, the midpoint of the class is used. In a spatio-temporal panel, the fitted density is evaluated at this representative spatial lag and at the temporal lag displayed in the panel title.
...	Additional graphical arguments passed to plot.

Details

An h-scatterplot displays paired observations associated with spatial locations separated by a given range of distances or by a specified nearest-neighbour order. The distance-class version requires maxdist and numbins; the nearest-neighbour version requires neighb. Nearest-neighbour scatterplots are generally preferable for large datasets. Purely spatial distance-class pairs are generated in blocks with bounded working memory. This avoids the former allocation of three vectors of length $n(n-1)/2$; the total number of retained pairs can still be quadratic when maxdist includes most location pairs. Distance classes use left-closed, right-open intervals, with the final class also including maxdist.

The original interface, in which data and coordinates are supplied directly, continues to produce scatterplots without requiring a fitted model. Supplying a GeoFit object is an additional interface that extracts the data, coordinates, distance, fitted parameters, and, when present, the fitted copula from the object. If neither neighb nor maxdist is supplied, the function attempts to use the corresponding pair-selection setting stored in the fitted object. Nearest-neighbour selection uses the same distance and radius convention as the fitted model (or as supplied in the raw interface), and the lags returned by GeoNeighIndex are reused directly.

For fixed spatio-temporal locations in the original interface, data must be a length(coordt) by nrow(coordx) matrix, following the same time-by-site convention used by GeoFit, or a vector in the corresponding time-block order. For dynamic locations, coordx_dyn must contain one coordinate matrix per temporal instant and data can be a matching list or a vector obtained by concatenating the temporal blocks. Pairs are constructed with GeoNeighIndex; no pair-graph object is required. The times argument can restrict the temporal instants before pairs are generated, while time.lag selects the displayed temporal lags.

With contour = TRUE, the bivariate density implied by the fitted model is evaluated on a rectangular grid and superimposed on the points. For a nearest-neighbour panel, the density is evaluated at the representative spatial lag selected by lag.method; for a distance-class panel, it is evaluated at the midpoint of the class.

For a fit obtained with `likelihood = "Marginal"` and `type = "Independence"`, the theoretical contour is based only on the fitted univariate marginal distribution. Any correlation model or copula parameters stored in the `GeoFit` object are ignored because they do not enter the independence likelihood. On the original scale the bivariate density is therefore $f_1(y_1)f_2(y_2)$. On the uniform scale the independence copula density is identically one, so there are no non-trivial contour levels. On the Gaussian-score scale the density is $\phi(z_1)\phi(z_2)$, giving the circular contours of two independent standard normal variables.

For a spatio-temporal `GeoFit` object, pairs are classified by both the spatial lag h and the temporal lag u. The fitted contour in each panel therefore uses the fitted correlation $\rho(h,u)$. At a positive temporal lag and with fixed spatial locations, the first nearest-neighbour candidate is generally the observation at the same spatial location and thus has $h = 0$. Such colocated temporal pairs are retained. More generally, `neighb = m` represents the candidate set containing the first m nearest neighbours at the selected temporal lag. This is the same candidate set returned by `GeoNeighIndex`; no additional filtering of zero spatial lags is applied inside `GeoScatterplot`. Both fixed and dynamic spatial locations are supported. When distance classes are requested, all spatial or spatio-temporal pairs within `maxdist` are constructed for the selected temporal lags and then divided into `numbins` classes.

When `scale = "Uniform"`, the plotted observations are $U_i = F_i(Y_i)$ and the fitted contour is the corresponding copula density $c_h(u_1,u_2)$. When `scale = "Gaussian"`, the plotted observations are $Z_i = \Phi^{-1}\{F_i(Y_i)\}$ and the contour is

$$c_h\{\Phi(z_1),\Phi(z_2)\}\phi(z_1)\phi(z_2),$$

that is, the fitted copula represented with standard Gaussian margins. The plotted observations on these two scales are obtained internally by calling GeoPit with `type = "Uniform"` or `type = "Gaussian"`; therefore the marginal CDF definitions are shared by the two diagnostic functions rather than being duplicated in `GeoScatterplot`.

For models fitted with `copula = "Gaussian"`, `"SkewGaussian"`, or `"Clayton"`, the uniform- and Gaussian-scale contours are evaluated directly from the fitted copula density. The response marginal distribution is not used in this contour calculation. This direct evaluation is numerically more stable and ensures that the Gaussian-scale plot represents the same fitted copula as the uniform-scale plot. For native non-copula random-field models, the copula density is recovered from the fitted response and marginal densities.

These transformations preserve the copula and are useful for displaying reflection symmetry or asymmetry without confounding from the fitted marginal distribution. In particular, reflection symmetry corresponds to invariance under $(u_1,u_2) \mapsto (1-u_1, 1-u_2)$ on the uniform scale and under $(z_1,z_2) \mapsto (-z_1,-z_2)$ on the Gaussian scale. By default, Gaussian-score panels use square plotting regions and the interval $[-3,3]^2$.

Native fitted contours are available for the continuous models `"Gaussian"`, `"SkewGaussian"`, `"Tukeyh"`, `"StudentT"`, `"SinhAsinh"`, `"LogGaussian"`, `"Gamma"`, and `"Weibull"`. Fitted contours are also available for `"Gaussian"`, `"SkewGaussian"`, and `"Clayton"` copulas with the continuous marginal models supported by the pairwise copula implementation: `"Gaussian"`, `"StudentT"`, `"LogGaussian"`, `"Gamma"`, `"Weibull"`, `"Beta"`, `"Beta2"`, `"Kumaraswamy"`, `"Kumaraswamy2"`, `"Logistic"`, and `"SkewLaplace"`.

Fitted contours are restricted to univariate, non-misspecified spatial or spatio-temporal `GeoFit` objects. If `scale = "Original"`, `residuals = FALSE`, and the fitted location varies among sites, a unique original-scale bivariate contour is not defined. In that case, use `residuals = TRUE` or select the uniform or Gaussian scale. The `residuals` argument is not used with the uniform or Gaussian scale.

Value

The function is primarily called for its graphical output.

With the purely spatial original data-and-coordinate interface, the matched call is returned invisibly. With spatio-temporal raw data, an invisible list is returned with components `call`, `spacetime`, `time_lags`, `time_values`, and `panels`. With a `GeoFit` object, an invisible list is returned with components `call`, `model`, `copula`, `independence`, `contour`, `scale`, `residuals`, `spacetime`, `time_lags`, and `panels`. For a marginal independence fit, `copula` is reported as `"Independence"` and `independence` is `TRUE`. Each panel contains the plotted paired values, representative spatial lag, and temporal lag. When contours are requested, it also contains the evaluation grid, density matrix, contour levels, and the fitted correlation used by the panel when available. Nearest-neighbour panels additionally contain the pair indices, individual spatial lags, and individual temporal lags.

Spatio-temporal ordering

For raw fixed-location input, `data` is a $T \times N$ matrix with times in rows and sites in columns. For raw dynamic input, `data` and `coordx_dyn` are lists of equal length and are matched within each time block. Pair indices and selected `time.lag` values refer to the resulting time-wise concatenation. A supplied `GeoFit` object uses the same conventions. See GeoModels-spacetime-ordering.

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)

####################################
### example 1 : Weibull random field
####################################
NN <- 1000
coords <- cbind(runif(NN), runif(NN))

corrmodel <- "GenWend"
model <- "Weibull"

param <- list(mean = 0, shape = 8, nugget = 0,
              scale = 0.5, smooth = 0, power2 = 4)

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

```
## Original interface: scatterplots on the response scale
GeoScatterplot(data, coords, neighb = c(1, 2))

## Fit
fit <- GeoFit(data = data, coordx = coords,
              corrmodel = corrmodel, model = "Weibull",
              likelihood = "Marginal", type = "Pairwise",
              neighb = 4,
              start = list(mean = 0, shape = 6,
                            scale = 0.3),
              fixed = list(nugget = 0, smooth = 0,
                           power2 = 4))

GeoScatterplot(fit, neighb = c(1, 2),nlevels=8,scale = "Original")

## Gaussian-score scale: useful for assessing reflection asymmetry
GeoScatterplot(fit, neighb = c(1, 2),nlevels=8, scale = "Gaussian")


#############################################################
### example 2 : beta random field from a skew-Gaussian copula
#############################################################
NN <- 1000
coords <- cbind(runif(NN), runif(NN))

corrmodel <- "GenWend"
model <- "Beta2"
copula <- "SkewGaussian"
nu=0.95
## Main example: Beta2 marginal model with a Skew-Gaussian copula
param <- list(mean = 0, shape = 8, min = 0, max = 1,
              nu = nu, nugget = 0,
              scale = 0.5, smooth = 0, power2 = 4)

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

## The simulated model is Beta2 with copula = "SkewGaussian"
## Original interface: scatterplots on the response scale
GeoScatterplot(data, coords, neighb = c(1, 2))

## Fit the Beta2 model with a Skew-Gaussian copula
fit <- GeoFit(data = data, coordx = coords,
              corrmodel = corrmodel, model = model,
              copula = copula,
              likelihood = "Marginal", type = "Pairwise",
              neighb = 4,
              start = list(mean = 0, shape = 6,
                            scale = 0.3),
              fixed = list(nugget = 0, smooth = 0,nu=nu,
                           power2 = 4, min = 0, max = 1),
              lower = list(mean = -Inf, shape = 0, scale = 0),
```

```
              upper = list(mean = Inf, shape = Inf, scale = Inf),
              optimizer = "nlminb")

## Response scale: marginal shape and copula are both visible
GeoScatterplot(fit, neighb = c(1, 2),
               scale = "Original")

## Gaussian-score scale: useful for assessing reflection asymmetry
GeoScatterplot(fit, neighb = c(1, 2),
               scale = "Gaussian")


#############################################################
### example 3 : spatio-temporal Gaussian random field
#############################################################

set.seed(89)
coordt <- 1:5
coords <- cbind(runif(200), runif(200))

corrmodel <- "Matern_Matern"
param <- list(mean = 0, sill = 1, nugget = 0,
              scale_s = 0.2 / 3, scale_t = 2 / 3,
              smooth_s = 0.5, smooth_t = 0.5)

data_st <- GeoSim(coordx = coords, coordt = coordt,
                  corrmodel = corrmodel,
                  model = "Gaussian", param = param)$data

fit_st <- GeoFit(data = data_st, coordx = coords, coordt = coordt,
                 corrmodel = corrmodel, model = "Gaussian",
                 likelihood = "Marginal", type = "Pairwise",
                 neighb = 3, maxtime = 1,
                 start = list(mean = 0, sill = 1,
                              scale_s = 0.1, scale_t = 0.5),
                 fixed = list(nugget = 0,
                              smooth_s = 0.5, smooth_t = 0.5))

## Four panels: two nearest-neighbour candidate sets at temporal lags 0 and 1.
## With fixed locations, the panel neighb = 1, time.lag = 1 contains the
## colocated temporal pairs and therefore has representative spatial lag h = 0.
out_st <- GeoScatterplot(fit_st, neighb = c(1, 3),
                         time.lag = c(0, 1),
                         scale = "Gaussian")

## The same space-time pair selection is available without fitting a model.
out_raw_st <- GeoScatterplot(data_st, coordx = coords, coordt = coordt,
                             neighb = c(1, 3), time.lag = c(0, 1),
                             contour = FALSE)

## Distance classes at selected temporal instants.
out_raw_bins <- GeoScatterplot(data_st, coordx = coords, coordt = coordt,
                               times = c(1, 3, 5), time.lag = c(0, 2),
```

```
                     maxdist = 0.3, numbins = 3,
                     contour = FALSE)
```

GeoScores *Computation of predictive scores*

Description

The function computes predictive scores for observed validation values from point predictions, kriging prediction objects, or conditional-simulation objects. Gaussian predictive scores are available from GeoKrig/GeoKrigloc through the pred/mse convention, while GeoSimcond objects are scored from their empirical conditional predictive distribution.

Usage

```
GeoScores(data_to_pred,
 probject = NULL,
 pred = NULL,
 mse = NULL,
 score = c("pe", "crps", "intscore", "coverage"),
 lower95 = NULL,
 upper95 = NULL,
 threshold = 0.5,
 na.rm = TRUE)
```

Arguments

data_to_pred
: Numeric vector, matrix or array containing the observed validation values. Values are internally coerced to a numeric vector.

probject
: Optional object of class GeoKrig, GeoKrigloc, or GeoSimcond. For GeoKrig/GeoKrigloc, point predictions and prediction variances are extracted from probject$pred and probject$mse. Requested probabilistic scores are computed from the Gaussian predictive convention $N(\widehat{Y}, MSE)$ for every marginal model. For a non-Gaussian model this is a Gaussian approximation based on the optimal linear predictor and its MSE; GeoScores reports it without deciding whether that approximation is appropriate for a particular analysis.

  For GeoSimcond, the conditional simulations in probject$condsim define an empirical predictive distribution. Point scores use probject$cond_mean; CRPS, PIT, Brier score, prediction intervals, interval score, and coverage are computed directly from the conditional draws, without a Gaussian approximation.

pred
: Numeric vector, matrix or array of point predictions. This argument is required when probject is not supplied.

`mse` Optional numeric vector, matrix or array of prediction variances. Predictive standard errors are computed as `sqrt(mse)`. This argument is not needed for `"pe"`, for interval score or coverage when `lower95` and `upper95` are supplied, or when `probject` is a `GeoSimcond` object. It is required for probabilistic scores based on the Gaussian `pred`/`mse` convention unless both interval bounds are supplied and used to infer a Gaussian standard error.

`score` Character vector specifying which predictive scores should be computed. Possible values are `"brie"`, `"crps"`, `"lscore"`, `"pit"`, `"pe"`, `"intscore"`, and `"coverage"`. Several values can be supplied.

`lower95` Optional numeric vector, matrix or array containing the lower bounds of the 95 percent prediction intervals. If supplied together with `upper95`, these bounds are used directly for `"intscore"` and `"coverage"`. For a `GeoSimcond` object they override the default empirical 2.5 and 97.5 percent conditional-simulation quantiles. If `mse` is not supplied in the Gaussian `pred`/`mse` path, they can also be used to infer a Gaussian predictive standard error.

`upper95` Optional numeric vector, matrix or array containing the upper bounds of the 95 percent prediction intervals. See `lower95`.

`threshold` Finite numeric scalar used to compute the Brier score `"brie"`. The binary event is $Y > threshold$.

`na.rm` Logical. If `TRUE`, invalid observations are removed separately for each requested score class. Thus an invalid predictive variance does not remove an otherwise valid observation from MAE/RMSPE, and an invalid interval does not affect point scores. If `FALSE`, invalid values relevant to any requested score produce an error.

Details

`GeoScores` dispatches predictive scoring according to the information contained in `probject`.

For a `GeoKrig` or `GeoKrigloc` object, `pred` is the point predictor and `mse` its mean squared prediction error. The option `"pe"` returns mean absolute error and root mean squared prediction error. Whenever probabilistic scores are requested, `GeoScores` uses the Gaussian predictive convention

$$Y_0 \mid Y \dot{\sim} N\{\widehat{Y}_0, MSE_0\}.$$

This convention is used for every marginal model. For a Gaussian random field it is the natural conditional predictive distribution (conditional on fitted parameters). For a non-Gaussian model, `GeoKrig`/`GeoKrigloc` still returns an optimal linear predictor, so CRPS, LogScore, PIT, Brier score, and Gaussian intervals obtained this way must be interpreted as Gaussian predictive approximations based on the OLP and its MSE. `GeoScores` does not suppress those quantities; their scientific interpretation is left to the user.

Predictions and prediction variances can equivalently be supplied directly through `pred` and `mse`; this uses the same Gaussian predictive convention.

For a `GeoSimcond` object, the replications stored in `probject$condsim` are treated as an empirical conditional predictive distribution at each prediction location. No Gaussian approximation is used. The point forecast for `"pe"` is `probject$cond_mean` (or the empirical conditional mean reconstructed from the simulations if needed). The Brier probability for the event $Y > threshold$ is the empirical conditional exceedance probability. PIT is the empirical conditional CDF at the validation observation. The default 95 percent predictive interval is given by the empirical 0.025 and 0.975 quantiles, unless explicit `lower95`/`upper95` values are supplied.

The empirical CRPS from conditional draws $x_1, \dots, x_B$ is

$$\widehat{CRPS}(F, y) = \frac{1}{B}\sum_{b=1}^{B} |x_b - y| - \frac{1}{2B^2}\sum_{b=1}^{B}\sum_{c=1}^{B} |x_b - x_c|.$$

It is evaluated internally with an equivalent sorted-sample formula requiring $O(B \log B)$ rather than $O(B^2)$ work per prediction location.

A logarithmic score is not computed automatically from a `GeoSimcond` object. Conditional draws identify the predictive distribution empirically but do not by themselves provide a predictive density evaluated exactly at the observed value. If `"lscore"` is requested from a `GeoSimcond` object, `LogScore = NA` is returned with a warning; no kernel-density approximation is introduced silently.

In the Gaussian `pred`/`mse` path, the Brier score uses the Gaussian predictive probability of $Y > threshold$; the logarithmic score is the mean negative Gaussian log predictive density; PIT uses the Gaussian CDF; and CRPS uses the closed-form Gaussian expression.

The 95 percent interval score is

$$(u - l) + \frac{2}{\alpha}(l - y)I(y < l) + \frac{2}{\alpha}(y - u)I(y > u), \qquad \alpha = 0.05.$$

For `GeoKrig`/`GeoKrigloc` without explicit interval bounds, l and u are Gaussian intervals based on `pred` and `mse`. For `GeoSimcond`, they are empirical conditional-simulation quantiles. Empirical coverage is the proportion of validation observations lying in the corresponding intervals.

Value

A list containing only the components requested through `score`:

`MAE`	Mean absolute error. For `GeoSimcond`, the empirical conditional mean is used as the point forecast.
`RMSPE`	Root mean squared prediction error.
`Brier`	Brier score for the event $Y > threshold$. It is Gaussian probability based for `GeoKrig`/`GeoKrigloc` and empirical conditional-simulation based for `GeoSimcond`.
`LogScore`	Mean negative Gaussian log predictive density for the `pred`/`mse` or `GeoKrig`/`GeoKrigloc` path. For a `GeoSimcond` object this component is `NA` if requested.
`PIT`	Gaussian PIT values for the `pred`/`mse` or kriging path, or empirical conditional CDF values for `GeoSimcond`.
`CRPS`	Mean Gaussian CRPS for the kriging/direct-prediction path, or mean empirical CRPS from conditional simulations for `GeoSimcond`.
`IS95`	Mean interval score for the 95 percent prediction intervals.
`Cvg95`	Empirical coverage of the 95 percent 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. E. (2007). *Strictly Proper Scoring Rules, Prediction, and Estimation*. Journal of the American Statistical Association, **102**, 359–378.

Heaton, M. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D. W., Sun, F., and Zammit-Mangion, A. (2019). *A Case Study Competition Among Methods for Analyzing Large Spatial Data*. Journal of Agricultural, Biological, and Environmental Statistics, **24**, 398–425.

Examples

```
library(GeoModels)

################################################################
######### Example of predictive score computation #############
################################################################

model <- "Gaussian"
set.seed(79)
N <- 1000

x <- runif(N, 0, 1)
y <- runif(N, 0, 1)
coords <- cbind(x, y)

# Set covariance 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

# Training and validation split
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]

# Pairwise likelihood fitting
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)

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)

# Prediction at validation locations
pr <- GeoKrig(fit,loc = coords_to_pred, data = data_est, mse = TRUE)

# Predictive scores from pred and mse
Pr_scores <- GeoScores(data_to_pred, pred = pr$pred, mse = pr$mse,
 score = c("pe", "brie", "crps", "lscore",
 "pit", "intscore", "coverage"),
 threshold = 0)

Pr_scores$MAE
Pr_scores$RMSPE
Pr_scores$Brier
Pr_scores$CRPS
Pr_scores$LogScore
Pr_scores$IS95
Pr_scores$Cvg95

# Conditional-simulation objects can be scored directly. For example,
# after obtaining cs <- GeoSimcond(..., nrep = 500), use
# GeoScores(data_to_pred, cs)
# to compute point scores, empirical CRPS, empirical 95 percent interval
# score, and empirical coverage from cs$condsim.
```

GeoSim *Simulation of Gaussian and non-Gaussian random fields.*

Description

Simulates a realization of a Gaussian or non-Gaussian spatial, spatio-temporal, or spherical random field, and Gaussian bivariate random fields, for a specified covariance or correlation model. The covariance parameters are supplied through `param`; available correlation models are documented in `GeoCovmatrix`.

Usage

```
GeoSim(coordx=NULL, 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,check.duplicates=FALSE)
```

Arguments

`coordx`	May be `NULL` when `coordx_dyn` or `spobj` is supplied. Otherwise, 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 the temporal coordinates at which the field is simulated. Optional argument; the default is `NULL`, in which case a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
`coordx_dyn`	For dynamic simulation sites, a list with one two- or three-column coordinate matrix per element of `coordt`. The output at time t follows the row order of `coordx_dyn[[t]]`. See `GeoModels-spacetime-ordering`.
`corrmodel`	String; the name of a correlation model, for the see `GeoCovmatrix` for the list of implemented correlation models.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the Euclidean distance. See `GeoFit` for details.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as spatial or spatio-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`	Positive integer size parameter. For Binomial it may be scalar or site-specific; for direct Negative Binomial it is the single common number r of successes, with Geometric corresponding to $r = 1$.
`param`	A list of parameter values required in the simulation procedure of random fields, 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 design matrix for the mean. For fixed locations, rows are ordered by time blocks, with all sites at the first time followed by all sites at the second time. For dynamic locations, supply either a stacked matrix in temporal-block order or a list with `X[[t]]` aligned with `coordx_dyn[[t]]`.

spobj
: An object of class `sp` or `spacetime`. For space-time objects, the current `sp2Geo()` conversion uses sequential temporal indices and does not preserve irregular original time spacing; to retain irregular temporal distances, use the explicit-coordinate interface with numeric `coordt`.

nrep
: Numeric; Numbers of indipendent replicates.

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

check.duplicates
: Logical. If `TRUE`, perform a fast exact scan for duplicated observation locations at this user entry point. The default is `FALSE`, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

Details

Bivariate simulation is currently implemented only for `model = "Gaussian"`; other bivariate marginal models are rejected explicitly. `Binary` and `Bernoulli` are aliases of a binomial field with `n = 1`; `Geom` and `Geometric` are aliases of a negative-binomial field with `n = 1`. Direct simulation of `model = "Beta2"` is not implemented by `GeoSim`; use `GeoSimCopula` with an explicit copula instead. The function stops explicitly rather than returning a latent Gaussian draw. Models whose names contain `Gaussian_misp_` are inferential working models and are rejected as data-generating mechanisms. Other model names that are valid elsewhere in GeoModels but do not have a direct simulator are also rejected explicitly.

Several direct non-Gaussian constructions use an integer number of independent latent Gaussian fields. Consequently, `Gamma` requires integer `shape`; `Beta` requires integer `shape1` and `shape2`; and for `StudentT`, `SkewStudentT`, and `TwoPieceStudentT`, `1/df` must be an integer at least 3. For `PoissonGamma` and `PoissonGammaZIP`, `2*shape` must be a positive integer. These parameters are never rounded silently. These restrictions apply to the direct latent-field constructions in `GeoSim`; they do not apply to quantile-transformed copula margins in `GeoSimCopula`.

`nrep` must be a positive integer. For binomial and negative-binomial fields, `n` must contain positive integers and may be scalar or have one value per simulated observation. The returned `param` component preserves the parameter list supplied by the user rather than the standardized latent-Gaussian working parameters.

For univariate simulation, the location parameter is $\mu = X\beta$, with coefficients named `mean`, `mean1`, and so on in the order of the columns of `X`. If `X=NULL`, the simulation is intercept-only and uses scalar `mean`. Alternatively, `param$mean` may be a vector with one value per simulated observation; this external mean is mutually exclusive with `X`.

For the Tukey transformed-Gaussian models, `Tukeyh` requires `0 <= tail < 0.5`, while `Tukeyh2` requires both `0 <= tail1 < 0.5` and `0 <= tail2 < 0.5`. In `Tukeyh2`, `tail1` is the right-tail parameter and `tail2` is the left-tail parameter. Zero is an allowed boundary and recovers the Gaussian transformation on the corresponding side. `SinhAsinh` requires a strictly positive `tail`.

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 simulated data. For fixed-location space-time simulation this is a matrix with times in rows and sites in columns; for dynamic locations it is a list with one vector per time, aligned with `coordx_dyn`. See `GeoModels-spacetime-ordering`.
`distance`	The type of spatial distance;
`method`	The method of simulation
`model`	The type of RF, see `GeoFit`.
`n`	The Binomial number of trials; for direct Negative Binomial, the common number r of successes.
`numcoord`	The number of spatial coordinates;
`numtime`	The number the temporal realisations of the RF;
`param`	The parameter list supplied to the simulation call;
`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;

Spatio-temporal ordering

With fixed locations, a simulated space-time realization is returned as a $T \times N$ matrix: rows correspond to `coordt` and columns to rows of `coordx`. The corresponding internal and `X` row order is `c(t(data))`, i.e. time then site.

With dynamic locations, the simulated realization is a list of length T. Element `data[[t]]` has one value per row of `coordx_dyn[[t]]`, in the same row order. Replicates, when requested, contain objects with this same fixed or dynamic layout. See `GeoModels-spacetime-ordering`.

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 for covariance matrix construction, GeoFit for parameter estimation, GeoSimcond and GeoSimapprox for conditional and approximate simulation.

Examples

```
library(GeoModels)


################################################################
###
### 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))
if (requireNamespace("fields", quietly = TRUE)) {
 fields::quilt.plot(coords, data1, main = "Matern", xlab = "", ylab = "")
}
if (requireNamespace("fields", quietly = TRUE)) {
 fields::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))

if (requireNamespace("fields", quietly = TRUE)) {
 fields::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
if (requireNamespace("fields", quietly = TRUE)) {
 fields::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
if (requireNamespace("fields", quietly = TRUE)) {
 fields::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))
```

```
if (requireNamespace("fields", quietly = TRUE)) {
 fields::quilt.plot(
  coords, data[1, ], col = terrain.colors(100), main = "1",
  xlab = "", ylab = ""
 )
}
if (requireNamespace("fields", quietly = TRUE)) {
 fields::quilt.plot(
  coords, data[2, ], col = terrain.colors(100), main = "2",
  xlab = "", ylab = ""
 )
}
par(opar)


################################################################
###
### Example 7. Simulation of a spatio temporal Gaussian random field.
### 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 random field.
### 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=NULL, coordy=NULL, coordz=NULL,coordt=NULL,
coordx_dyn=NULL,corrmodel, distance="Eucl",
grid=FALSE, max.ext=1,
method="TB", L=10000,model='Gaussian',parallel=FALSE,ncores=6,
n=1,param,anisopars=NULL, radius=6371,X=NULL,spobj=NULL,
nrep=1,progress=TRUE,check.duplicates=FALSE)
```

Arguments

coordx	May be NULL when coordx_dyn or spobj is supplied. Otherwise, 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 the temporal coordinates. Optional argument; the default is NULL, in which case a spatial random field is expected. For space-time method = "CE", coordt must be finite, strictly increasing, and equally spaced; space-time turning-bands simulation is not implemented.
coordx_dyn	For dynamic simulation sites, a list with one two- or three-column coordinate matrix per element of coordt. The output at time t follows the row order of coordx_dyn[[t]]. See GeoModels-spacetime-ordering.
corrmodel	String; the name of a correlation model, for the see GeoCovmatrix for the list of implemented correlation models.
parallel	Logical; default FALSE. For method = "TB", TRUE enables an adaptive scheduler that chooses among serial execution, parallelization over spatial chunks within each turning-bands simulation, and parallelization over independent replicates. Small workloads can remain serial when worker overhead is expected to dominate.
ncores	Positive integer or NULL; default 6. With parallel=TRUE, an explicit integer requests that many workers, capped by detected cores and the available TB jobs. Set ncores=NULL for automatic selection, capped at six workers and normally leaving one detected core free.
distance	String; the name of the spatial distance. The default is Eucl, the Euclidean distance. Turning-bands simulation requires distance="Eucl". See GeoFit for details.
grid	Logical; if FALSE (the default) the data are interpreted as spatial or spatio-temporal realisations on a set of non-equispaced spatial sites (irregular grid).
max.ext	Positive integer; maximum number of successive doubling attempts used by the spatial or temporal CE embedding.
method	String; the approximation method. The default is TB (turning bands). The alternative CE uses circulant embedding. Spatial CE requires an exact regular two-dimensional grid, Euclidean distance and no anisopars. Spatial TB accepts

explicit irregular two- or three-dimensional Euclidean coordinates for univariate fields. Spatio-temporal CE requires fixed spatial locations, an equally spaced time vector and a supported separable correlation model.

`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` Positive integer size parameter. For Binomial it may be scalar or site-specific; for direct Negative Binomial it is the single common number r of successes, with Geometric corresponding to $r = 1$.

`param` A list of parameter values required in the simulation procedure of random fields, 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 design matrix for the mean. For fixed locations, rows are ordered by time blocks, with all sites at the first time followed by all sites at the second time. For dynamic locations, supply either a stacked matrix in temporal-block order or a list with `X[[t]]` aligned with `coordx_dyn[[t]]`.

`spobj` An object of class `sp` or `spacetime`. For space-time objects, the current `sp2Geo()` conversion uses sequential temporal indices and does not preserve irregular original time spacing; to retain irregular temporal distances, use the explicit-coordinate interface with numeric `coordt`.

`nrep` Numeric; Numbers of indipendent replicates.

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

`check.duplicates`
Logical. If `TRUE`, perform a fast exact scan for duplicated observation locations at this user entry point. The default is `FALSE`, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

Details

For `method = "CE"`, spatial simulations require `grid = TRUE`, separate equally spaced axes in `coordx` and `coordy`, `distance = "Eucl"`, and `anisopars = NULL`. The simulated grid is exactly the grid supplied by the user. The `sill` and `nugget` parameters are applied once to the latent Gaussian field.

Spatio-temporal CE is a separable hybrid method: Cholesky decomposition is used for the spatial correlation matrix and circulant embedding/FFT for the temporal correlation. It requires fixed spatial locations and a finite, strictly increasing, equally spaced `coordt`. Dynamic coordinates supplied through `coordx_dyn` are not supported by the approximate methods; use `GeoSim(..., method = "cholesky")` instead. A single temporal instant is handled as a spatial draw at the supplied locations. Bivariate approximate simulation is not available with `method = "CE"`; use `method = "TB"` or `GeoSim()`. Turning bands are implemented only for purely spatial models; `method = "TB"` is rejected for all spatio-temporal correlation models and requires `distance="Eucl"`. Purely spatial univariate TB is restricted to the Matern, generalized-Wendland / hypergeometric, and Kummer families implemented by the spectral sampler; unsupported correlation families are rejected before entering the TB kernel. L must be a positive integer.

Bivariate approximate simulation is restricted to model="Gaussian", corrmodel="Bi_matern", common spatial support for the two variables, and method="TB". The two marginal sills and nugget effects are applied to the standardized bivariate TB output before the component means are added. Non-Gaussian bivariate approximate simulation is rejected explicitly.

Binary and Bernoulli are aliases of a binomial field with n = 1; Geom and Geometric are aliases of a negative-binomial field with n = 1. Direct simulation of model = "Beta2" is not implemented by GeoSimapprox; use GeoSimCopula with an explicit copula instead. The function stops explicitly rather than returning a latent Gaussian draw. Inferential Gaussian_misp_* model names and any other model without an implemented direct simulator are also rejected explicitly.

For the Tukey transformed-Gaussian models, Tukeyh requires 0 <= tail < 0.5, while Tukeyh2 requires both 0 <= tail1 < 0.5 and 0 <= tail2 < 0.5. In Tukeyh2, tail1 is the right-tail parameter and tail2 is the left-tail parameter. Zero is an allowed boundary and recovers the Gaussian transformation on the corresponding side. SinhAsinh requires a strictly positive tail.

The integer latent-field restrictions of GeoSim apply unchanged: Gamma requires integer shape; Beta requires integer shape1 and shape2; 1/df must be an integer at least 3 for the direct Student-t constructions; and 2*shape must be integer for the direct Poisson-Gamma constructions. No such parameter is rounded silently. Binomial and negative-binomial n values must be positive integers. The negative-binomial simulator uses per-location counters rather than storing the complete Bernoulli history, and count-process simulators accumulate counts without a growing event-indicator matrix. nrep must be a positive integer. The returned param component preserves the parameter list supplied by the user.

For univariate simulation, the location parameter is $\mu = X\beta$, with coefficients named mean, mean1, and so on in the order of the columns of X. If X=NULL, the simulation is intercept-only and uses scalar mean. Alternatively, param$mean may be a vector with one value per simulated observation; this external mean is mutually exclusive with X.

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 simulated data. For fixed-location space-time simulation this is a matrix with times in rows and sites in columns; for dynamic locations it is a list with one vector per time, aligned with coordx_dyn. See GeoModels-spacetime-ordering.
distance	The type of spatial distance;
method	The method of simulation
model	The type of RF, see GeoFit.

`n`	The Binomial number of trials; for direct Negative Binomial, the common number r of successes.
`numcoord`	The number of spatial coordinates;
`numtime`	The number the temporal realisations of the RF;
`param`	The parameter list supplied to the simulation call;
`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;

Spatio-temporal ordering

With fixed locations, a simulated space-time realization is returned as a $T \times N$ matrix: rows correspond to `coordt` and columns to rows of `coordx`. The corresponding internal and `X` row order is `c(t(data))`, i.e. time then site.

With dynamic locations, the simulated realization is a list of length T. Element `data[[t]]` has one value per row of `coordx_dyn[[t]]`, in the same row order. Replicates, when requested, contain objects with this same fixed or dynamic layout. See `GeoModels-spacetime-ordering`.

Three-dimensional coordinates

For a purely spatial univariate field, `method = "TB"` supports explicit irregular $N \times 3$ Euclidean coordinates with `grid = FALSE` and `anisopars = NULL`. Three-dimensional TB is not implemented for spatio-temporal or bivariate models. CE remains restricted to regular two-dimensional grids. For unsupported three-dimensional cases, use `GeoSim(..., method = "cholesky")` instead.

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
if (requireNamespace("fields", quietly = TRUE)) {
  fields::image.plot(matrix(data1, length(x), length(y), byrow = TRUE))
}

################################################################
###
### Example 2. Simulation of a large spatial Tukey-h RF
### with 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=0.06,nugget=0,tail=0.15)
# Simulation of a spatial Gaussian RF with Matern correlation function
data1 <- GeoSimapprox(coords, corrmodel="Matern", model="Tukeyh",
                      method="TB",L=1000,param=param)$data
if (requireNamespace("fields", quietly = TRUE)) fields::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))
#fields::quilt.plot(coords,data[1,],col=terrain.colors(100),main="1",xlab="",ylab="")
#fields::quilt.plot(coords,data[2,],col=terrain.colors(100),main="2",xlab="",ylab="")
```

GeoSimcond *Conditional simulation of spatial Gaussian and non-Gaussian random fields*

Description

Performs global or nearest-neighbour local conditional simulation for univariate spatial random fields at explicit prediction coordinates. Cholesky simulation is available together with the approximate turning-bands method when supported by the selected correlation model. The combination `local=TRUE` and `method="TB"` avoids global observation covariance matrices and is intended for large spatial datasets.

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 = 6, progress=FALSE, n_iter=25L,
 check.duplicates=FALSE, nloc=NULL, mcmc_thin=1L)
```

Arguments

`estobj`	Object of class `GeoFit` containing model information
`data`	Numeric vector/matrix/array of observed data
`coordx`	Numeric matrix with one row per observed location and two or three spatial coordinate columns
`coordy`	Optional numeric vector of y-coordinates
`coordz`	Optional numeric vector of z-coordinates
`coordt`	Retained for API compatibility. Space-time conditional simulation is not currently implemented.
`coordx_dyn`	Retained for API compatibility. Dynamic coordinates are not currently implemented.
`corrmodel`	String specifying correlation model name
`distance`	String specifying distance metric (default: "Eucl")
`grid`	Must currently be `FALSE`; supply explicit coordinates.
`loc`	Numeric matrix of prediction locations (n x 2)
`maxdist`	Optional maximum distance for local kriging
`maxtime`	Optional maximum temporal distance
`method`	Unconditional simulation method. Currently `"Cholesky"` and `"TB"` are available; `"CE"` is retained for API compatibility but rejected by the explicit-coordinate conditional workflow.
`model`	String specifying random field type (default: "Gaussian")
`n`	For direct `Binomial`, the number of trials, either scalar or one positive integer per observed location. For direct `BinomialNeg`, the common positive integer number r of successes.
`nrep`	Number of retained conditional simulation replicates (default: 1). For direct count models these are successive thinned draws from the latent Gibbs chain.
`local`	Logical. If `FALSE` (default), global conditioning is used. If `TRUE`, each prediction location is conditioned only on the observations selected by `neighb` and/or `maxdist`.

`L` Number of lines for turning bands method (default: 1000)

`neighb` Optional positive integer giving the number of nearest observed locations used for local conditional simulation. When `local=TRUE`, at least one of `neighb` or `maxdist` must be supplied.

`param` List of parameter values

`anisopars` List with anisotropy angle and ratio

`radius` Radius used by spherical distance calculations (default: 1)

`sparse` Must currently be `FALSE`.

`time` Retained for API compatibility. Space-time conditional simulation is not currently implemented.

`copula` Optional string specifying copula type

`X` Optional design matrix for the marginal location at the observed locations.

`Xloc` Optional design matrix for the marginal location at prediction locations. If `X` is supplied and the margin uses a location parameter, either `Xloc` or `Mloc` must be supplied.

`Mloc` Optional vector of known/fitted marginal location values at prediction locations; an alternative to `Xloc`.

`parallel` Logical; default `FALSE`. If `TRUE`, supported expensive stages use parallel workers. For `method="TB"` or `"CE"`, the setting is passed to `GeoSimapprox`; for `local=TRUE`, local kriging weights may also be computed in parallel. The final local substitution is evaluated in the main R process using a compact sparse weight operator and bounded batches, avoiding duplication of large simulation and weight objects across workers.

`ncores` Positive integer or `NULL`; default `6`. With `parallel=TRUE`, an explicit integer requests that many workers, capped by detected cores and available jobs. Set `ncores=NULL` for automatic selection, capped at six workers and normally leaving one detected core free. A value of 1 is treated as serial execution.

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

`n_iter` Positive integer; number of Gibbs sweeps used by latent conditional samplers. For direct `Binomial` and `BinomialNeg`, it is the burn-in length of the count-constrained Gibbs chain; retained simulations are then separated by `mcmc_thin` sweeps. The default for these direct count models is 25. For the other latent Gibbs engines, omitting `n_iter` retains the historical default of 1000 sweeps. An explicitly supplied value is always respected. It is ignored by the Gaussian copula conditional simulator.

`check.duplicates` Logical. If `TRUE`, perform a fast exact scan for duplicated observation locations at this user entry point. The default is `FALSE`, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.

`nloc` Prediction-side count size. For direct `Binomial`, a scalar or one positive integer per prediction location; it is required when observation-side `n` varies by site. For direct `BinomialNeg`, if supplied it must equal the common r.

`mcmc_thin` Positive integer thinning interval for the direct Binomial and Negative-Binomial count-constrained latent Gibbs sampler. The default is 1.

Details

The currently validated domain is univariate, purely spatial, non-dynamic and non-grid conditional simulation. Both global and nearest-neighbour local conditioning are available for the Gaussian latent-substitution paths described below. Unsupported domain combinations are rejected before covariance matrices or unconditional simulations are constructed.

Without a copula, the currently supported models are `Gaussian`, `LogGaussian`, `Tukeyh`, `Tukeyh2`, `SinhAsinh`, `SkewGaussian`, `Gamma`, `Weibull`, `Binomial`, and `BinomialNeg`. Direct non-copula `Gamma` conditional simulation uses the finite Gaussian-square construction and therefore requires `param$shape` to be a positive integer; non-integer values are rejected rather than rounded. For direct and Gaussian-copula Tukey margins, `Tukeyh` requires `0 <= tail < 0.5`; `Tukeyh2` requires `0 <= tail1 < 0.5` and `0 <= tail2 < 0.5`, with `tail1` the right-tail and `tail2` the left-tail parameter. Zero is valid and recovers the Gaussian transformation on that side. `SinhAsinh` requires a strictly positive `tail`.

For the direct `Binomial` and `BinomialNeg` random fields, conditional simulation uses the repeated latent Gaussian threshold construction defining the models. At each observed site the latent Gaussian copies are updated as one count-constrained block. Their signs are sampled exactly from a conditional Bernoulli distribution by a log-scale dynamic program, after which the latent Gaussian magnitudes are sampled from the corresponding univariate truncated normal full conditionals. For `Binomial`, the constraint is

$$\sum_{l=1}^{n_i} I\{Z_l(s_i) > 0\} = y_i.$$

For `BinomialNeg`, writing $t_i = y_i + r$, the constraint is

$$\sum_{l=1}^{t_i-1} I\{Z_l(s_i) > 0\} = r - 1, \qquad Z_{t_i}(s_i) > 0.$$

The native sampler maintains the dense Gaussian precision update in compiled code and directly generates the prediction-side latent fields, so the full $L \times n_{obs} \times n_{rep}$ latent array is not returned to R. The first `n_iter` full site sweeps are discarded and retained conditional realisations are separated by `mcmc_thin` sweeps. This direct count path currently requires `method="Cholesky"` and `local=FALSE`.

With `copula="Gaussian"`, conditional simulation is implemented for the continuous margins `Gaussian`, `StudentT`, `LogGaussian`, `Gamma`, `Weibull`, `Beta`, `Beta2`, `Kumaraswamy`, `Kumaraswamy2`, `Logistic`, `SkewLaplace`, `Tukeyh`, `Tukeyh2`, and `SinhAsinh`. With `copula="Clayton"` or `copula="SkewGaussian"`, the currently validated continuous margins are the same list except for `Tukeyh`, `Tukeyh2`, and `SinhAsinh`. For copula-based `Gamma`, `Weibull`, `Beta`, `Beta2`, `Kumaraswamy`, and `Kumaraswamy2` margins, `param$sill` is not required: their marginal dispersion is determined by the corresponding shape parameters. The `sill` parameter is required only for copula margins whose marginal scale is explicitly parameterized by it. For the skew–Gaussian copula, `param$nu` is the bounded asymmetry parameter $\eta \in (-1, 1)$; the latent Gibbs sampler uses $\gamma_\eta = \eta/\sqrt{1-\eta^2}$.

For the constructive Clayton-like copula, `param$nu` must be a positive integer. If u_i is the observed copula uniform, the sampler uses $r_i = u_i^{2/\nu}$ and the latent representation

$$r_i = \frac{\sum_{k=1}^{\nu} Z_{ki}^2}{\sum_{k=1}^{\nu} Z_{ki}^2 + W_{1i}^2 + W_{2i}^2}.$$

At each observed site the $\nu + 2$ Gaussian latent variables are updated with a Gibbs kernel that preserves this ratio exactly. Directional full conditionals are von Mises–Fisher and the common latent radius is updated from a one-dimensional log-concave full conditional. The retained finite-sweep state is therefore a Monte Carlo approximation to the Clayton latent conditional distribution after `n_iter` sweeps. Conditional Gaussian simulation of the latent fields at `loc` is then followed by the Clayton ratio reconstruction and the location-specific marginal quantile.

Discrete copula margins are not routed through the former mid-PIT approximation, because that approximation is not exact conditioning on the latent intervals.

For Gaussian fields, conditional simulations use the substitution method

$$Z_c(s_0) = E\{Z(s_0) \mid Z(s) = z\} + Z^*(s_0) - WZ^*(s),$$

where the unconditional residual field Z^* is generated with zero mean and the covariance specified by `param`. The declared orientation of the kriging weights is used explicitly.

With `local=TRUE`, the same substitution identity is applied separately at each prediction location using only its selected neighborhood N_j:

$$Z_c(s_j) = \hat{Z}_j + Z^*(s_j) - w_j^T Z^*_{N_j}.$$

The local systems are solved once, and GeoSimcond keeps a compact map of neighbor indices and weights. When possible this map is assembled once as a `spam` sparse matrix with n_{loc} rows and n_{obs} columns. Conditional corrections for multiple realizations are then evaluated in bounded matrix batches rather than by repeating the local graph reduction for every replicate. The sparse representation contains only the selected local weights; GeoSimcond does not retain one local covariance matrix per prediction site. Prediction locations with no observation inside a `maxdist`-only neighborhood receive zero conditioning correction, so their local conditional approximation reduces to the marginal mean plus the unconditional residual at that location.

Local conditioning is currently validated for `Gaussian`, `LogGaussian`, `Tukeyh`, `Tukeyh2`, and `SinhAsinh` without a copula, and for all supported continuous margins with `copula="Gaussian"`. In particular, `Tukeyh`, `Tukeyh2` and `SinhAsinh` are also supported as continuous margins under the Gaussian copula and use the same latent-Gaussian fast path. The Gibbs-based constructive `SkewGaussian`, `Gamma`, `Weibull`, `Binomial`, and `BinomialNeg` paths and the `Clayton` and `SkewGaussian` copulas still require `local=FALSE`.

For the sinh–arcsinh model, the implementation uses the monotone transformation

$$Y(s) = \mu(s) + \sqrt{\sigma^2}\ \sinh\{[\operatorname{asinh}\{Z(s)\} + \eta]/\nu\},$$

and conditions on the Gaussian scale using its exact inverse

$$Z(s) = \sinh\{\nu\ \operatorname{asinh}[(Y(s) - \mu(s))/\sqrt{\sigma^2}] - \eta\}.$$

Thus `LogGaussian`, `Tukeyh`, `Tukeyh2` and `SinhAsinh` require only one latent Gaussian conditional simulation per replicate; no non-Gaussian Gibbs step is introduced.

For margins with a location parameter, location-specific covariates are preserved: if `X` is supplied for the observations, prediction requires either `Xloc` or `Mloc`; the function does not silently replace a varying prediction-side location by the intercept.

For continuous copula margins, observations are transformed by their fitted marginal CDF to the copula scale, conditional simulation is performed on the latent copula construction, and each realization is transformed back with the location-specific marginal quantile. For the Gaussian copula this is standard Gaussian conditional simulation. The skew–Gaussian case uses its latent Gibbs sampler, while the Clayton-like case uses the Gaussian-square ratio Gibbs sampler described above. In both MCMC cases, a retained finite-sweep draw is a Monte Carlo approximation to the target conditional distribution after `n_iter` sweeps.

`method="TB"` makes the unconditional simulation step approximate. For large datasets it can be combined with `local=TRUE`; the turning-bands simulation then avoids a dense covariance factorization for the unconditional field, while local conditioning avoids the global observation covariance system. For a fixed neighborhood size m, local preprocessing consists of small $m \times m$ kriging systems and the substitution correction uses only the retained local weights. For multiple conditional realizations, the same sparse local weight operator is reused and applied in bounded matrix batches. This optimization is shared automatically by the one-Gaussian monotone transformations and by Gaussian-copula margins. When `parallel=TRUE`, approximate unconditional simulation and local-weight construction honor `ncores`; the final sparse substitution remains in the main R process to avoid copying the unconditional simulations and weight map to additional worker processes. Spatial circulant embedding is not currently available because it requires a regular grid, whereas the validated conditional workflow combines explicit observation and prediction coordinates. The nugget, distance, radius and anisotropy settings are propagated to the latent Gaussian simulations.

Value

An object of class `GeoSimcond` containing `condsim`, a list with one element per successful conditional replicate. Each element is a numeric vector of length `numloc`, in prediction-location order. Thus, if a dense matrix is needed for post-processing, `do.call(rbind, lapply(x$condsim, as.numeric))` produces a matrix with replications in rows and prediction locations in columns. The object also contains `cond_mean` and `cond_var`, computed across replicates using a streaming update without materializing an additional full replicate-by-location matrix, together with the model, parameter, conditioning (`local`, `neighb`) and simulation-method metadata. With a single replicate, `cond_var` contains `NA` values because a sample variance cannot be estimated.

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.

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

Caamaño-Carrillo, C., Bevilacqua, M., López, C. and Morales-Oñate, V. (2024). Nearest neighbors 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*, 191, 107887.

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

# Empirical pointwise intervals from the conditional simulations.
# Rows are replications and columns are prediction locations.
sim_mat <- do.call(rbind, lapply(sim_result$condsim, as.numeric))
interval_summary <- t(apply(sim_mat, 2, quantile,
                            probs = c(0.025, 0.5, 0.975)))
colnames(interval_summary) <- c("q025", "q500", "q975")
```

```
head(interval_summary)

par(mfrow=c(1,3))
if (requireNamespace("fields", quietly = TRUE)) fields::quilt.plot(coords, data)
if (requireNamespace("fields", quietly = TRUE)) fields::quilt.plot(loc_to_sim, cond_mean)
if (requireNamespace("fields", quietly = TRUE)) fields::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))
if (requireNamespace("fields", quietly = TRUE)) fields::quilt.plot(coords, data)
```

```
if (requireNamespace("fields", quietly = TRUE)) fields::quilt.plot(loc_to_sim,cond_mean)
if (requireNamespace("fields", quietly = TRUE)) fields::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 univariate spatial and spatio-temporal random fields using Gaussian, skew-Gaussian, Clayton-like or AMH constructions. Bivariate correlation models are not supported. The function returns a realization of a random field for a given covariance model and covariance parameters. Exact simulation is available through Cholesky or SVD decomposition; for supported purely spatial correlation models, turning-bands simulation is available through `method="TB"`.

Usage

```
GeoSimCopula(coordx=NULL, 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,check.duplicates=FALSE,
 L=10000,parallel=FALSE,ncores=6)
```

Arguments

coordx	Optional when `coordx_dyn` or `spobj` is supplied. 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 the temporal coordinates at which the field is simulated. Optional argument; the default is `NULL`, in which case a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
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 see `GeoCovmatrix` for the list of implemented correlation models.
distance	String; the name of the spatial distance. The default is `Eucl`, the Euclidean distance. See `GeoFit` for details.

grid	Logical; if FALSE (the default) the data are interpreted as spatial or spatio-temporal realisations on a set of non-equispaced spatial sites (irregular grid).
method	String; simulation engine for the latent fields. Use "cholesky" (default) or "svd" for exact simulation through GeoSim, and "TB" for turning-bands simulation through GeoSimapprox.
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 random fields. The number of successes in the negative Binomial random fields. Default is 1.
param	A list of parameter values required in the simulation procedure of random fields, 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; one of "Gaussian", "SkewGaussian", "Clayton", or "AMH".
X	Numeric; Matrix of space-time covariates.
spobj	An object of class sp or spacetime. For space-time objects, the current sp2Geo() conversion uses sequential temporal indices and does not preserve irregular original time spacing; to retain irregular temporal distances, use the explicit-coordinate interface with numeric coordt.
nrep	Numeric; Numbers of indipendent replicates.
progress	Logic; If TRUE then a progress bar is shown.
check.duplicates	Logical. If TRUE, perform a fast exact scan for duplicated observation locations at this user entry point. The default is FALSE, so no duplicate-location scan is imposed. Internal bootstrap, cross-validation, and refitting calls do not repeat the scan.
L	Positive integer; number of turning-band lines used when method="TB". Ignored by exact methods.
parallel	Logical; default FALSE. For method="TB", set TRUE to allow GeoSimapprox to select a parallel turning-bands execution path.
ncores	Positive integer or NULL; default 6. With parallel=TRUE and method="TB", an explicit integer requests that many workers for the TB backend, subject to detected-core and job-count limits. Set ncores=NULL for automatic selection.

Details

Only univariate correlation models are accepted. Dynamic coordinates are supported and the returned data preserve the list structure and row order of coordx_dyn.

With method="TB", GeoSimCopula keeps the copula and marginal construction unchanged but generates each required latent field through GeoSimapprox(method="TB"). Consequently, TB is available only for purely spatial models supported by the turning-bands backend, requires distance="Eucl", and inherits the TB restrictions on correlation families and coordinate dimension. The sparse argument is ignored by the TB backend. Exact "cholesky" and "svd" simulations continue to use GeoSim.

For the constructive Clayton-like simulator, param$nu must be a positive integer and is never rounded silently. For copula="SkewGaussian", param$nu is the bounded reflection-asymmetry parameter $\eta \in (-1, 1)$ and is internally mapped to $\gamma_\eta = \eta/\sqrt{1-\eta^2}$. The continuous margins audited for copula simulation and pairwise copula fitting with the Gaussian, Clayton-like and skew–Gaussian copulas are Gaussian, StudentT, LogGaussian, Gamma, Weibull, Beta, Beta2, Kumaraswamy, Kumaraswamy2, Logistic, and SkewLaplace. The Gaussian copula additionally has audited simulation and pairwise fitting for Poisson, Binomial, and BinomialNeg. Exact conditional simulation for these discrete copula margins is not currently implemented. For model="StudentT", param$df is the reciprocal degrees-of-freedom parameter: the Student-t degrees of freedom are exactly 1/param$df, without integer rounding. The same no-rounding convention is used for the Student-t degrees of freedom of a "SkewStudentT" marginal. Because copula margins are obtained through inverse CDFs, the integer latent-field restrictions used by direct GeoSim / GeoSimapprox Gamma, Beta, and Student-t constructions do not apply here. Gamma margins require a finite positive shape and use shape shape/2 and scale 2/shape, hence the multiplicative factor has unit mean. With dynamic coordinates, X may be a row-stacked matrix or a list of matrices aligned with coordx_dyn. For SkewStudentT, skew is the native $\delta \in (-1, 1)$ parameter and is converted to the skew-t shape $\alpha = \delta/\sqrt{1-\delta^2}$. The Tukey-g-and-h implementation uses its continuous limit when the skew parameter is zero. Kumaraswamy margins use the inverse of the CDF employed by GeoPit.

Aliases accepted by the main modelling API, including Gauss, SkewGauss, LogGauss, and TwoPieceGauss, are canonicalized before the marginal transform. nrep must be a positive integer. For the AMH construction, nu is checked to be a finite scalar; no additional theoretical parameter range is imposed by this simulator. The returned param component is the parameter list supplied by the user.

Binary and Bernoulli are aliases of Binomial with n=1; Geom and Geometric are aliases of BinomialNeg with n=1.

For univariate simulation, the marginal location is $\mu = X\beta$, with coefficients named mean, mean1, and so on in the order of the columns of X. If X=NULL, the model is intercept-only. A vector param$mean supplies a known location value at each observation and cannot be combined with X.

Value

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

bivariate	Always FALSE; bivariate correlation models are rejected.
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 simulated data. Dynamic space-time output is a list with one vector per time, aligned with coordx_dyn; fixed-location space-time output preserves the matrix layout of GeoSim.
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 random fields;the number of successes in a negative Binomial random fields;
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

if (requireNamespace("fields", quietly = TRUE)) fields::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)
if (requireNamespace("fields", quietly = TRUE)) fields::quilt.plot(coords,data1)
```

GeoTestIndependence *Parametric bootstrap test for absence of spatial dependence*

Description

Performs a parametric bootstrap likelihood-ratio test of spatial independence against a spatially dependent random-field model. Under the null hypothesis, the observations at distinct sites are fitted using the marginal independence likelihood. Under the alternative hypothesis, a full spatial likelihood is fitted with the nugget proportion estimated below its independence boundary. The bootstrap is required because the null hypothesis corresponds to a boundary point and the spatial correlation parameters are not identified under the null.

Usage

```
GeoTestIndependence(data, coordx,
 start, fixed = NULL,
 corrmodel = "Matern",
 model = "Gaussian",
 optimizer = "bobyqa",
 optimizer_ind = "Nelder-Mead",
 lower = NULL, upper = NULL,
 X = NULL, n = 1,
 distance = "Eucl", radius = 1,
```

```
    anisopars = NULL,
    est.aniso = c(FALSE, FALSE),
    sparse = FALSE,
    B = 1000,
    parallel = FALSE,
    ncores = 6,
    progress = TRUE,
    seed = NULL)
```

Arguments

`data`	Numeric vector of observations.
`coordx`	Matrix or object coercible to a matrix containing the spatial coordinates. The number of rows must equal `length(data)`. Duplicate locations are not allowed.
`start`	Non-empty named list containing the parameters to be estimated under the spatial alternative. It must include, either directly or through `fixed`, all parameters required by `model` and `corrmodel`. The parameter `nugget` is always estimated under the alternative; if it is supplied in `fixed`, it is automatically moved to `start`.
`fixed`	Optional named list containing parameters kept fixed under the spatial alternative. Parameters cannot occur in both `start` and `fixed`.
`corrmodel`	Correlation model used under the spatial alternative. The default is `"Matern"`.
`model`	Marginal model for the observations. The default is `"Gaussian"`.
`optimizer`	Optimization method passed to `GeoFit` for the full spatial fit under H_1. The default is `"bobyqa"`.
`optimizer_ind`	Optimization method used for the marginal independence fit under H_0. The default is `"Nelder-Mead"`. If the requested method fails, the function also tries `"Nelder-Mead"` as a fallback and retains the successful fit with the largest finite log-likelihood.
`lower, upper`	Optional named lists of lower and upper bounds for the parameters estimated under the spatial alternative. Names must refer only to parameters in `start`, after the possible promotion of `nugget` from `fixed`. The upper bound for `nugget` must be one, or omitted, so that the independence model belongs to the closure of the alternative.
`X`	Optional design matrix for the mean or marginal regression model. Its number of rows must equal `length(data)`.
`n`	Model-specific parameter passed to `GeoFit` and `GeoSim`. The default is `1`.
`distance`	Distance metric passed to `GeoFit`, `GeoDistances`, and `GeoSim`. The default is `"Eucl"`.
`radius`	Positive radius used for spherical or geodesic distances. The default is `1`.
`anisopars`	Optional list of anisotropy parameters passed to the spatial fit under the alternative.
`est.aniso`	Logical vector of length two indicating whether the anisotropy angle and ratio are estimated under the alternative. The default is `c(FALSE, FALSE)`.

`sparse`	Logical value passed to the full-likelihood spatial fit. The default is `FALSE`.
`B`	Number of parametric bootstrap replications. The default is `1000`. Values below `99` produce a coarse bootstrap *p*-value and generate a warning.
`parallel`	Logical; default `FALSE`. Set `TRUE` to evaluate bootstrap fits in parallel when the required packages are available.
`ncores`	Positive integer or `NULL`; default `6`. With `parallel=TRUE`, an explicit integer requests that many workers, capped by detected cores and the number of bootstrap replications. Set `ncores=NULL` for automatic selection, capped at six workers and normally leaving one detected core free.
`progress`	Logical value indicating whether progress bars should be displayed through **progressr**.
`seed`	Optional finite numeric seed. When supplied, the previous global random-number-generator state is restored when the function exits.

Details

The test compares

$$H_0 : \text{absence of spatial dependence}$$

with

$$H_1 : \text{a spatially dependent random-field model.}$$

Under H_0, the function fits `GeoFit(..., likelihood = "Marginal", type = "Independence")`. Only marginal and regression parameters are estimated. Spatial correlation parameters and the nugget proportion are absent from the null fit because they are not identified under independence. The null fit uses `optimizer_ind`; if necessary, a `"Nelder-Mead"` fallback is attempted. The optimizer associated with the selected null fit is returned in `optimizer_independence`.

Under H_1, the function fits `GeoFit(..., likelihood = "Full", type = "Standard")` using `corrmodel`. The nugget parameter is estimated on $[0, 1)$. Independence is represented by the boundary `nugget` $=$ 1, where the spatial correlation contribution vanishes. The upper optimization bound is internally set to a value immediately below one because `GeoFit` requires `nugget < 1`.

The observed likelihood-ratio statistic is

$$\Lambda_{obs} = 2\{\ell_1 - \ell_0\},$$

where ℓ_0 is the maximized independence log-likelihood and ℓ_1 is the maximized full spatial log-likelihood. If the unrestricted spatial fit reaches the independence boundary, the statistic is set to zero and the function returns `pvalue = 1` without running the bootstrap.

The usual chi-squared likelihood-ratio approximation is not used because the null value is on the boundary and the spatial correlation parameters are not identified under H_0. Instead, the null distribution is approximated by parametric bootstrap. The bootstrap *p*-value is

$$\widehat{p} = \frac{1 + \sum_{b=1}^{B_{valid}} I(\Lambda_b^* \geq \Lambda_{obs})}{B_{valid} + 1}.$$

Bootstrap datasets are generated exactly independently at the observed sites. Because `GeoSim` does not accept `nugget = 1`, the implementation uses an auxiliary `GenWend` correlation model with compact support equal to one half of the minimum inter-site distance. Consequently, all off-diagonal correlations at the observed locations are exactly zero. This auxiliary model is used only as a simulation device; the fitted null model remains the marginal independence model.

The observed parameter estimates are used as starting values in the bootstrap refits. If the first spatial fit fails to attain at least the independence log-likelihood, adaptive fallback fits are attempted using alternative initial values for `nugget`. When the best spatial fit is numerically equivalent to the null fit, the bootstrap statistic is treated as a boundary value equal to zero.

When `parallel = TRUE`, bootstrap fits are evaluated with **future.apply**. Each simulated dataset is written temporarily to a separate RDS file so that workers do not receive the complete simulation object. The user's previous **future** plan, **progressr** handlers, and random-number- generator state are restored on exit.

The current implementation is restricted to univariate models with a single nugget proportion named `nugget`. It compares the independence likelihood only with the full spatial likelihood. It does not directly compare the independence likelihood with a pairwise composite likelihood, because the two criteria are not on the same scale.

Value

An invisible list with components:

`lambda_obs` Observed likelihood-ratio statistic.

`pvalue` Parametric bootstrap p-value.

`nugget_hat` Estimated nugget proportion under the spatial alternative.

`boundary_observed`
: Logical value indicating whether the unrestricted spatial fit was classified as an independence-boundary solution.

`optimizer_independence`
: Optimizer used by the selected independence fit under H_0. This can differ from `optimizer_ind` when the `"Nelder-Mead"` fallback is selected.

`optimizer_spatial`
: Optimizer used for the full spatial fit under H_1.

`seed` Seed supplied by the user, or `NULL`.

`B_requested` Requested number of bootstrap replications.

`bootstrap_successful`
: Number of valid bootstrap replications.

`bootstrap_failed`
: Number of failed bootstrap replications.

`bootstrap_success_rate`
: Fraction of successful bootstrap replications. This is `NA` when the observed statistic is zero and the bootstrap is not run.

`fallback_observed`
: Logical value indicating whether adaptive starting values were required for the observed spatial fit.

`fallback_bootstrap`
: Number of bootstrap replications in which adaptive starting values were used for the spatial fit.

`boundary_bootstrap`
: Number of valid bootstrap replications classified as independence-boundary solutions.

`simulation_support`
: Compact-support value used by the auxiliary `GenWend` model to simulate independent observations. This component is absent when the observed statistic is zero and the bootstrap is skipped.

`B_rep`
: Vector of valid bootstrap likelihood-ratio statistics.

`fit_H0`
: `GeoFit` object fitted under spatial independence.

`fit_H1`
: `GeoFit` object fitted under the spatial alternative.

Note

For Gaussian random fields, the null hypothesis is equivalent to mutual independence of the observations at distinct spatial locations. For non-Gaussian models, the interpretation as independence requires that zero latent spatial dependence imply independence in the corresponding observed random-field construction.

A small bootstrap *p*-value indicates evidence of spatial dependence. For final analyses, a value such as `B = 999` or larger is recommended.

Author(s)

Moreno Bevilacqua <moreno.bevilacqua89@gmail.com> https://sites.google.com/view/moreno-bevilacqua/home

See Also

`GeoFit`, `GeoSim`, `GeoDistances`, `GeoTestsupp_space`, `GeoTestIsotropy`

Examples

```
## Not run:
library(GeoModels)

set.seed(3)
N <- 300
coords <- cbind(runif(N), runif(N))

X <- cbind(1, runif(N))

param <- list(
  mean = 0.2,
  mean1 = -0.5,
  sill = 1,
  nugget = 0,
  scale = 0.2 / 3,
  smooth = 0.5
)

data <- GeoSim(
```

```
  coordx = coords,
  corrmodel = "Matern",
  model = "Gaussian",
  param = param,
  X = X
)$data

start <- list(
  mean = 0,
  mean1 = 0,
  sill = 1,
  nugget = 0.1,
  scale = 0.1
)

fixed <- list(smooth = 0.5)

lower <- list(
  mean = -Inf,
  mean1 = -Inf,
  sill = 1e-6,
  nugget = 0,
  scale = 1e-6
)

upper <- list(
  mean = Inf,
  mean1 = Inf,
  sill = Inf,
  nugget = 1,
  scale = 1
)

ans <- GeoTestIndependence(
  data = data,
  coordx = coords,
  X = X,
  start = start,
  fixed = fixed,
  corrmodel = "Matern",
  model = "Gaussian",
  optimizer = "bobyqa",
  optimizer_ind = "Nelder-Mead",
  lower = lower,
  upper = upper,
  B = 199,
  parallel = TRUE,
  seed = 123
)

ans$pvalue
ans$lambda_obs
ans$nugget_hat
```

```
ans$bootstrap_success_rate

## End(Not run)
```

GeoTestIsotropy *Parametric Bootstrap Tests for Spatial Isotropy*

Description

Performs two parametric-bootstrap tests of spatial isotropy by comparing an isotropic model under the null hypothesis with a geometrically anisotropic model under the alternative. The first test (MWCLT) is based on the estimated anisotropy ratio. The second test (WCLRT) is based on a likelihood-ratio or composite-likelihood-ratio statistic. Their null distributions are estimated by simulation from the fitted isotropic model.

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,
 method = c("cholesky", "TB"),
 L = 10000,
 n = 1,
 angle_starts = c(pi/4, 3*pi/4),
 batch_size = NULL,
 max_attempts = NULL,
 parallel = FALSE,
 ncores = 6,
 progress = TRUE,
 seed = NULL)
```

Arguments

`data`	Numeric vector of observations.
`coordx`	Numeric matrix with exactly two columns containing the spatial coordinates. The number of rows must equal `length(data)`.
`start`	Named list containing the covariance and marginal parameters to be estimated. Parameters `angle` and `ratio` must not be included, because they are handled internally by the isotropy test.

`fixed`	Named list containing the covariance and marginal parameters kept fixed. Together, `start` and `fixed` must contain exactly the parameters required by `CorrParam(corrmodel)` and `NuisParam(model)`.
`optimizer`	Optimization algorithm passed to `GeoFit`.
`model`	Marginal random-field model.
`corrmodel`	Spatial correlation model.
`lower, upper`	Named lists of lower and upper bounds for the parameters in `start`. Missing bounds are replaced by `-Inf` and `Inf`, respectively.
`B`	Number of valid parametric-bootstrap replications used to calibrate each test. Failed fitting attempts are replaced by newly simulated samples.
`likelihood`	Likelihood type passed to `GeoFit`. If both `likelihood` and `type` are `NULL`, full likelihood is used for at most 2000 observations and marginal pairwise composite likelihood is used otherwise.
`type`	Likelihood type passed to `GeoFit`; typically `"Standard"` for full likelihood or `"Pairwise"` for composite likelihood. It must be supplied together with `likelihood`.
`copula`	Optional copula specification. With a supported purely spatial correlation model, both `method="cholesky"` and `method="TB"` are available.
`neighb`	Number of nearest neighbours used when `type = "Pairwise"`. Ignored otherwise.
`method`	Simulation method under the null hypothesis. Use `"cholesky"` for exact simulation or `"TB"` for approximate turning-bands simulation. When `copula` is supplied, the corresponding latent engine is selected internally by `GeoSimCopula`.
`L`	Number of spectral components used when `method = "TB"`.
`n`	Common number of trials/successes used by models that require an integer `n`, such as Binomial-type models. It can be a scalar or, where supported, a vector with one value per observation. For `"Bernoulli"` and `"Binary"`, `n` is internally fixed to 1.
`angle_starts`	Finite numeric vector of starting angles used for the anisotropic fit. All supplied angles are tried (modulo π) and the fit with the largest objective value is retained. The default uses two starts, $\pi/4$ and $3\pi/4$, for both the observed fit and every bootstrap refit.
`batch_size`	Number of bootstrap datasets simulated and processed at one time. If `NULL`, a modest batch size is selected automatically. Batching limits memory use for large spatial datasets.
`max_attempts`	Maximum number of simulated bootstrap datasets allowed in order to collect exactly `B` valid replications. If `NULL`, the default is `max(B + 20, ceiling(1.5 * B))`. The function stops if this limit is reached before `B` valid replications are obtained.
`parallel`	Logical; if `TRUE`, bootstrap refits are distributed across multiple R sessions. With `method="TB"`, bootstrap simulation may also use multiple workers when supported by the simulation routine.

`ncores`	Positive integer or `NULL`. With `parallel=TRUE`, an explicit integer requests that many workers, capped by detected cores and the current number of jobs. Set `ncores=NULL` for automatic selection.
`progress`	Logical; if `TRUE`, display the same **progressr** text progress bar used by other parallel **GeoModels** functions. A single bar is shown for the whole bootstrap and counts valid bootstrap replications in both sequential and parallel execution.
`seed`	Optional finite numeric seed. The previous random-number-generator state is restored when the function exits.

Details

The null and alternative hypotheses are

$$H_0 : \mathrm{ratio} = 1,$$

and

$$H_1 : \mathrm{ratio} > 1,$$

with the anisotropy angle estimated under the alternative. Under H_0, the angle is not identifiable and the ratio is on the boundary of its parameter space. Consequently, a standard chi-squared reference distribution is not assumed; both tests are calibrated by parametric bootstrap.

The MWCLT statistic is the unconstrained anisotropy-ratio estimate $\widehat{\tau}$. If $\widehat{\tau}_b^*$ denotes the corresponding estimate in bootstrap replication b, the upper-tail bootstrap p-value is

$$\widehat{p}_\tau = \frac{1 + \sum_{b=1}^{B} I(\widehat{\tau}_b^* \geq \widehat{\tau})}{B+1}.$$

The WCLRT statistic is

$$T_{\mathrm{obs}} = \max\{0, 2(\ell_1 - \ell_0)\},$$

where ℓ_0 and ℓ_1 are the maximized likelihoods or composite likelihoods under the isotropic and anisotropic models. Its bootstrap p-value is

$$\widehat{p}_T = \frac{1 + \sum_{b=1}^{B} I(T_b \geq T_{\mathrm{obs}})}{B+1}.$$

The returned components `statistic` and `pvalue` remain aliases for this WCLRT statistic and p-value for backward compatibility.

For every observed or bootstrap anisotropic fit, all angles in `angle_starts` are used as starting values and the valid fit with the largest objective value is retained. During bootstrap refitting, the fitted observed anisotropy angle is also included as a starting angle. If the best numerical anisotropic fit has an objective value below the isotropic fit by more than numerical tolerance, the boundary solution is selected, giving ratio 1 and WCLRT statistic 0.

For pairwise composite likelihood, the isotropic and anisotropic fits must use the same retained pairs. The function checks this condition before computing the WCLRT statistic.

Bootstrap datasets are generated from the fitted isotropic model in batches. Failed bootstrap fits do not enter the empirical null distribution: new samples are generated until exactly `B` valid replications have been collected, or until `max_attempts` is reached. This makes the denominator of both bootstrap p-values exactly `B+1`.

The argument `n` is propagated consistently to `GeoFit`, `GeoSim`, `GeoSimapprox`, and `GeoSimCopula`. Direct latent Gaussian simulation of some non-Gaussian models imposes additional structural constraints. In particular, direct `"Gamma"` simulation requires the fitted `shape` to be a positive integer because it is the number of squared latent Gaussian fields. For the direct Gamma model used in the latent construction, it is therefore recommended to keep `shape` fixed at a positive integer during the isotropy test. This integer restriction does not apply merely because the marginal distribution is Gamma when a supported copula construction is used.

Value

An invisible list containing, among others:

`statistic`, `pvalue`
: Backward-compatible aliases for the observed WCLRT statistic and its bootstrap p-value.

`statistic_lrt`, `pvalue_lrt`
: Observed WCLRT statistic and p-value.

`statistic_ratio`, `pvalue_ratio`
: Observed MWCLT anisotropy-ratio statistic and p-value.

`pvalues`
: Named vector containing the MWCLT and WCLRT p-values.

`ratio_hat`, `angle_hat`
: Estimated anisotropy ratio and angle. The angle is `NA` when the boundary solution is selected.

`parallel`, `ncores`
: Whether parallel bootstrap refitting was used and the number of workers selected by the function.

`fit_H0`, `fit_H1`
: Fitted isotropic and best anisotropic `GeoFit` objects for the observed data.

`B_rep`, `B_rep_lrt`
: The `B` valid bootstrap WCLRT statistics.

`B_rep_ratio`
: The `B` valid bootstrap anisotropy-ratio estimates.

`B_rep_angle`
: Bootstrap angle estimates; boundary solutions are `NA`.

`bootstrap_B`
: Requested and obtained number of valid bootstrap replications.

`bootstrap_total_attempts`
: Total number of simulated datasets needed to obtain `B` valid replications.

`bootstrap_failed_attempts`
: Number of failed attempts that were replaced.

`bootstrap_failures`
: Data frame describing failed attempts.

`bootstrap_boundary_solutions`
: Number of valid bootstrap replications for which the boundary solution was selected.

`observed_multistart_angles`
: Starting angles used for the observed H1 fit.

`batch_size`, `max_attempts`
: Bootstrap batching and retry settings used by the function.

Author(s)

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

See Also

GeoFit, GeoSim, GeoSimapprox, GeoVariogramDir

Examples

```
## Not run:
## Isotropy tests for a Gaussian random field simulated under H0
set.seed(123)
nsite <- 500
coords <- cbind(runif(nsite), runif(nsite))

param <- list(
  mean = 0,
  nugget = 0,
  sill = 1,
  scale = 0.20,
  smooth = 0.5
)

z <- GeoSim(
  coordx = coords,
  corrmodel = "Matern",
  model = "Gaussian",
  param = param
)$data

start <- list(
  mean = 0,
  sill = 1,
  scale = 0.20,
  smooth = 0.5
)
fixed <- list(nugget = 0)

lower <- list(mean = -5, sill = 0.01, scale = 0.01, smooth = 0.05)
upper <- list(mean =  5, sill = 5,    scale = 1,    smooth = 2.5)

iso_test <- GeoTestIsotropy(
  data = z,
  coordx = coords,
  start = start,
  fixed = fixed,
  model = "Gaussian",
  corrmodel = "Matern",
```

```
  lower = lower,
  upper = upper,
  B = 99,
  likelihood = "Full",
  type = "Standard",
  method = "cholesky",
  seed = 321
)

iso_test$pvalue_ratio  # MWCLT
iso_test$pvalue_lrt    # WCLRT

## End(Not run)
```

GeoTests *Statistical Hypothesis Tests for Nested Models*

Description

Performs Wald and likelihood-ratio tests for successively nested GeoFit models. Standard likelihood fits can be compared using Wald or Wilks tests, whereas composite likelihood fits can be compared using Wald or Satterthwaite-adjusted composite likelihood-ratio tests.

Usage

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

Arguments

object1	An unrestricted fitted model of class GeoFit.
object2	A restricted fitted model of class GeoFit, nested within object1.
...	Further fitted models of class GeoFit, ordered from less restricted to more restricted. Adjacent models are compared.
statistic	A character string specifying the test statistic. Available choices are "Wald", "Wilks", and "WilksS". See Details.

Details

The models must be supplied from the unrestricted model to successively more restricted models. For each adjacent pair, the tested parameters are the parameters estimated in the unrestricted model but fixed in the restricted model. Each tested parameter must therefore occur explicitly in the fixed component of the restricted fit.

The function checks that adjacent models use the same data, coordinates, design matrix, marginal model, correlation model, likelihood type, pair construction, and other relevant fitting options. Common fixed parameters must have the same values in the two models.

The following statistics are available:

- `"Wald"`: available for both standard and composite likelihood fits. The statistic is

$$W_{\mathrm{Wald}} = (\widehat{\boldsymbol{\psi}} - \boldsymbol{\psi}_0)^\top \widehat{V}_{\psi\psi}^{-1} (\widehat{\boldsymbol{\psi}} - \boldsymbol{\psi}_0),$$

  where $\boldsymbol{\psi}$ is the vector of tested parameters and $\widehat{V}_{\psi\psi}$ is the corresponding block of the estimated covariance matrix of the unrestricted estimator. Under the null hypothesis, the statistic is asymptotically chi-squared with degrees of freedom equal to the number of restrictions. For composite likelihood fits, `varcov` must contain the Godambe covariance matrix, usually obtained by applying `GeoVarest` to the unrestricted fit.

- `"Wilks"`: available only for standard full-likelihood fits. The statistic is

$$W_{\mathrm{Wilks}} = 2\{\ell(\widehat{\boldsymbol{\theta}}) - \ell(\widetilde{\boldsymbol{\theta}})\},$$

  where $\widehat{\boldsymbol{\theta}}$ and $\widetilde{\boldsymbol{\theta}}$ are the unrestricted and restricted maximum likelihood estimates, respectively. Under standard regularity conditions, the statistic is asymptotically chi-squared with degrees of freedom equal to the number of restrictions.

- `"WilksS"`: available only for composite likelihood fits, including marginal, conditional, and difference composite likelihoods. Let

$$W_{CL} = 2\{\ell_{CL}(\widehat{\boldsymbol{\theta}}) - \ell_{CL}(\widetilde{\boldsymbol{\theta}})\}.$$

  Its asymptotic null distribution is generally a weighted sum

$$W_{CL} \dot{\sim} \sum_{j=1}^{r} \lambda_j \chi^2_{1,j},$$

  where r is the number of restrictions. In the implementation, $\lambda_1, \ldots, \lambda_r$ are the eigenvalues of

$$H_{\mathrm{eff}} V_{\psi\psi}, \qquad H_{\mathrm{eff}} = \{(H^{-1})_{\psi\psi}\}^{-1},$$

  with H the sensitivity matrix and $V = H^{-1} J H^{-1}$ the Godambe covariance matrix from the unrestricted fit. The Satterthwaite moment-matching constants are

$$\nu = \frac{(\sum_j \lambda_j)^2}{\sum_j \lambda_j^2}, \qquad \kappa = \frac{\sum_j \lambda_j^2}{\sum_j \lambda_j},$$

  and the adjusted statistic

$$W_S = \frac{W_{CL}}{\kappa}$$

  is compared with a chi-squared distribution having ν degrees of freedom. The unrestricted fit must contain both `sensmat` and the Godambe covariance matrix `varcov`; these can be obtained using `sensitivity = TRUE` in `GeoFit`, followed by `GeoVarest`.

For a composite likelihood-ratio comparison, the unrestricted and restricted fits must be based on exactly the same retained pairs. This is automatic for deterministic pair constructions. With stochastic thinning, the two fits must use the same thinning realization; `GeoTests` checks that their `rowidx` and `colidx` components are identical and stops otherwise.

The matrices used by `"WilksS"` are evaluated at the unrestricted fit. Under the null hypothesis, this is a consistent plug-in evaluation because the unrestricted and restricted estimators converge to the same parameter value.

Value

A data frame containing one row for each supplied model. The first row corresponds to the unrestricted model and contains missing test results. Each subsequent row reports the comparison between that model and the model in the preceding row. The columns are:

Num.Par Number of estimated parameters in the model.

Diff.Par Number of restrictions relative to the preceding model.

Df Degrees of freedom used for the reference chi-squared distribution. For "WilksS", this is the Satterthwaite effective degrees of freedom and need not be an integer.

Chisq Observed value of the test statistic.

Pr(>chisq) Upper-tail p-value from the chi-squared reference distribution with the reported 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

Kent, J. T. (1982). Robust Properties of Likelihood Ratio Tests. *Biometrika*, **69**, 19–27.

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, GeoVarest.

Examples

```
library(GeoModels)

################################################################
### Example 1. Test of a regression coefficient using
### conditional pairwise composite likelihood
################################################################
set.seed(342)
model <- "Gaussian"

NN <- 1500
x <- runif(NN, 0, 1)
y <- runif(NN, 0, 1)
coords <- cbind(x, y)
```

```
mean <- 1
mean1 <- -1.25
nugget <- 0
sill <- 1

X <- cbind(1, runif(nrow(coords)))

corrmodel <- "Wend0"
power2 <- 4
c_supp <- 0.15

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: regression mean with a nonzero slope
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)

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
)

## H0: constant mean, that is, mean1 = 0
fixed0 <- list(power2 = power2, nugget = nugget, mean1 = 0)
start0 <- list(mean = mean, scale = c_supp, sill = sill)
lower0 <- list(mean = -I, scale = 0, sill = 0)
upper0 <- list(mean = I, scale = I, sill = I)

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 = start0, fixed = fixed0
)

## Not run:
## GeoVarest is needed on the unrestricted fit to estimate its
## Godambe covariance matrix. Increase K for final analyses.
fitH1 <- GeoVarest(fitH1, K = 100, seed = 1234)
```

```
GeoTests(fitH1, fitH0, statistic = "Wald")
GeoTests(fitH1, fitH0, statistic = "WilksS")

## End(Not run)

################################################################
### Example 2. Test of Gaussianity for a Sinh--arcsinh random field
### using the standard full likelihood
################################################################
set.seed(99)
model <- "SinhAsinh"

NN <- 200
x <- runif(NN, 0, 1)
y <- runif(NN, 0, 1)
coords <- cbind(x, y)

mean <- 0
nugget <- 0
sill <- 1
skew <- 0
tail <- 1

corrmodel <- "Wend0"
power2 <- 4
c_supp <- 0.2

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: unrestricted Sinh--arcsinh 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)

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
)

## H0: Gaussianity, that is, skew = 0 and tail = 1
fixed0 <- list(
```

```
  power2 = power2, nugget = nugget, mean = mean,
  tail = 1, skew = 0
)
start0 <- list(scale = c_supp, sill = sill)
lower0 <- list(scale = 0, sill = 0)
upper0 <- list(scale = 2, sill = 5)

fitH0 <- GeoFit(
  data = data, coordx = coords, corrmodel = corrmodel,
  model = model, likelihood = "Full", type = "Standard",
  varest = TRUE, lower = lower0, upper = upper0,
  optimizer = "nlminb", start = start0, fixed = fixed0
)

GeoTests(fitH1, fitH0, statistic = "Wald")
GeoTests(fitH1, fitH0, statistic = "Wilks")
```

GeoTestsupp_space *Parametric Bootstrap Test for Spatial Support*

Description

Performs a parametric bootstrap likelihood-ratio test for the spatial *scale*, interpreted as the compact-support radius of the "GenWend" correlation model. The null model either constrains the scale by $\mathrm{scale} \leq h_0$ or, when h0 = NULL, represents spatial independence by fixing the support below the minimum observed inter-site distance. The alternative estimates the scale without the null upper constraint.

Usage

```
GeoTestsupp_space(data, coordx,
 start, fixed,
 model = "Gaussian",
 h0 = NULL,
 optimizer = "bobyqa",
 lower = NULL, upper = NULL,
 neighb = 5,
 B = 10000,
 likelihood = NULL,
 type = NULL,
 method = c("cholesky", "TB"),
 L = 10000,
 parallel = FALSE,
 ncores = 6,
 progress = TRUE,
 seed = NULL)
```

Arguments

`data`
: Numeric vector containing the spatial observations. Missing or non-finite values are not allowed.

`coordx`
: Numeric matrix whose rows contain the spatial coordinates. Its number of rows must equal `length(data)`. Duplicated locations are not allowed because the minimum inter-site distance must be positive.

`start`
: Non-empty named list containing parameters to be estimated. Together with `fixed`, it must contain exactly the parameters returned by `CorrParam("GenWend")` and `NuisParam(model)`.

`fixed`
: Named list containing parameters kept fixed, or `NULL`. A parameter cannot appear in both `start` and `fixed`. If `scale` is supplied in `fixed`, it is internally moved to the estimated-parameter list for construction of the unrestricted model.

`model`
: Marginal model passed to `GeoFit`. Default is `"Gaussian"`.

`h0`
: Positive finite threshold defining $H_0 : \text{scale} \leq h_0$. If `NULL`, the support is fixed at `max(1e-6, d_min - 1e-6)`, where d_{min} is the minimum inter-site distance. This removes dependence among the sampled locations; for Gaussian random fields it is equivalent to independence of the sampled observations. It should not be confused with the process-level null hypothesis tested by `GeoTestIndependence`.

`optimizer`
: Optimization method passed to `GeoFit`. Default is `"bobyqa"`.

`lower, upper`
: Named lists containing bounds for parameters in `start`. Missing bounds are replaced internally by finite fallback bounds. The null upper bound for `scale` is always reimposed after any numerical widening of the parameter box.

`neighb`
: Number of nearest neighbors used for pairwise composite likelihood. It is ignored for full likelihood. When full likelihood is selected automatically, it is internally set to `NULL`.

`B`
: Positive integer giving the requested number of parametric bootstrap replications. Default is `10000`. Values below 99 are allowed but generate a warning because they give a coarse bootstrap *p*-value.

`likelihood`
: Likelihood passed to `GeoFit`. Supported values are `"Full"`, `"Marginal"`, and `"Conditional"`. Either both `likelihood` and `type` must be supplied, or both must be `NULL`.

`type`
: Likelihood type passed to `GeoFit`. Full likelihood must use `"Standard"`; marginal or conditional composite likelihood must use `"Pairwise"`.

`method`
: Simulation method used under the fitted null model. `"cholesky"` calls `GeoSim`; `"TB"` calls `GeoSimapprox`.

`L`
: Number of spectral components used by `GeoSimapprox` when `method = "TB"`. Default is `10000`. Ignored for Cholesky simulation.

`parallel`
: Logical; default `FALSE`. Set `TRUE` to evaluate bootstrap fits in parallel when the required packages are available.

`ncores`
: Positive integer or `NULL`; default `6`. With `parallel=TRUE`, an explicit integer requests that many workers, capped by detected cores and the number of bootstrap replications. Set `ncores=NULL` for automatic selection, capped at six workers and normally leaving one detected core free.

`progress`	Logical value indicating whether progress information should be displayed through **progressr**.
`seed`	Optional finite numeric seed. When supplied, the test is reproducible and the user's previous random-number-generator state is restored when the function exits.

Details

The correlation model is fixed internally to `"GenWend"`. Therefore, the parameter called `scale` is interpreted as the compact-support radius.

Likelihood selection. When both `likelihood` and `type` are `NULL`, the function uses full likelihood with `type = "Standard"` for at most 2000 observations, and marginal pairwise composite likelihood for larger datasets. User-supplied combinations are checked for consistency.

Null and alternative models. For a numeric `h0`, the null model estimates `scale` subject to $\mathrm{scale} \leq h_0$, while the unrestricted model uses a data-dependent upper bound strictly larger than `h0`. For `h0 = NULL`, the null model fixes the support radius just below the minimum observed inter-site distance. Since the generalized Wendland correlation is compactly supported, all distinct observations are then uncorrelated under the null model.

The lower bound under the alternative must be smaller than `h0`, and its upper bound must be larger than `h0`, so that the null parameter space is nested in the alternative. The function stops if these conditions are not satisfied.

Likelihood-ratio statistic. Let ℓ_0 and ℓ_1 denote the fitted log-likelihood or log-composite-likelihood values under the null and unrestricted models. The observed statistic is

$$\Lambda_{\mathrm{obs}} = 2(\ell_1 - \ell_0).$$

Tiny negative differences attributable to numerical tolerance are set to zero. A materially smaller unrestricted likelihood is treated as an optimization failure rather than automatically converted to a zero statistic.

For pairwise composite likelihood, the function verifies that the null and unrestricted fits use identical `rowidx` and `colidx` pair sets. A likelihood-ratio comparison based on different pair sets is not accepted.

Adaptive unrestricted-fit fallback. The unrestricted model is first fitted once using the primary starting values. Only if this fit fails or returns a likelihood smaller than the null likelihood beyond numerical tolerance is a second fit attempted. The fallback starts from the fitted null parameters and places `scale` slightly above `h0`. The better valid unrestricted fit is retained. The same adaptive rule is used for each bootstrap dataset.

Parametric bootstrap. Bootstrap datasets are simulated from the fitted null model. Each dataset is refitted under both hypotheses, using the observed null and unrestricted estimates as starting values. Replications with failed fits, incompatible pair sets, non-finite likelihoods, or an unresolved ordering $\ell_1 < \ell_0$ are discarded. If B_{valid} replications are successful, the bootstrap *p*-value is

$$\widehat{p} = \frac{1 + \sum_{b=1}^{B_{\mathrm{valid}}} I(\Lambda_b \geq \Lambda_{\mathrm{obs}})}{B_{\mathrm{valid}} + 1}.$$

At least `min(20, B)` valid replications are required. A warning is issued when fewer than 80 percent of the requested replications are successful.

Early exit. If the unrestricted estimate itself satisfies the null constraint, the likelihood-ratio statistic is zero and the function returns `pvalue = 1` without running the bootstrap.

Parallel computation. For `method = "TB"`, simulation may use parallel computation inside `GeoSimapprox`. Bootstrap refits are performed in a separate phase. In the parallel bootstrap, each simulated dataset is written to a separate RDS file so that each worker reads only its own replicate. The user's **future** plan, **progressr** handlers, RNG state, and temporary files are restored or removed on exit.

Value

An invisible list. When the bootstrap is run, it contains:

`d_min` Minimum inter-site distance.

`h0` Threshold actually used under the null hypothesis.

`lambda_obs` Observed likelihood-ratio statistic.

`pvalue` Parametric-bootstrap p-value.

`seed` Seed supplied by the user, or `NULL`.

`method` Simulation method used.

`L` Number of turning-bands components when `method = "TB"`, otherwise `NULL`.

`B_requested` Requested number of bootstrap replications.

`bootstrap_successful` Number of valid bootstrap replications.

`bootstrap_failed` Number of failed or discarded bootstrap replications.

`bootstrap_success_rate` Fraction of requested replications that were valid.

`fallback_observed` Logical value indicating whether the adaptive unrestricted-fit fallback was used for the observed data.

`fallback_bootstrap` Number of bootstrap replications in which the adaptive unrestricted-fit fallback was attempted.

`B_rep` Vector of valid bootstrap likelihood-ratio statistics.

`fit_H0` Fitted `GeoFit` object under the null hypothesis.

`fit_H1` Fitted `GeoFit` object under the unrestricted alternative.

For an early exit, `lambda_obs = 0`, `pvalue = 1`, `bootstrap_successful = 0`, `bootstrap_success_rate = NA`, and `B_rep` is empty. In the current implementation, `L` and `bootstrap_failed` are omitted from this early-return object.

Note

The test concerns the compact-support radius of the generalized Wendland correlation model and is not a generic range-parameter test for arbitrary correlation functions. For final inference, a substantially larger value of `B` than that used in quick examples is recommended.

Author(s)

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

See Also

GeoFit, GeoSim, GeoSimapprox, GeoTestIndependence, GeoTestIsotropy, GeoTests

Examples

```
## Not run:
library(GeoModels)

set.seed(123)
n <- 300
coords <- cbind(runif(n), runif(n))

param <- list(
  mean = 0,
  nugget = 0,
  sill = 1,
  scale = 0.20,
  smooth = 0.5,
  power2 = 4
)

z <- GeoSim(
  coordx = coords,
  corrmodel = "GenWend",
  model = "Gaussian",
  param = param
)$data

start <- list(
  mean = 0,
  sill = 1,
  scale = 0.15
)
fixed <- list(
  nugget = 0,
  smooth = 0.5,
  power2 = 4
)

lower <- list(mean = -Inf, sill = 1e-6, scale = 1e-6)
upper <- list(mean = Inf, sill = Inf, scale = 1)

## Test H0: scale <= 0.10 against scale > 0.10.
ans <- GeoTestsupp_space(
```

```
  data = z,
  coordx = coords,
  start = start,
  fixed = fixed,
  h0 = 0.10,
  lower = lower,
  upper = upper,
  B = 199,
  likelihood = "Full",
  type = "Standard",
  parallel = FALSE,
  seed = 321
)

ans$pvalue
ans$lambda_obs
ans$bootstrap_success_rate

## Independence test: support fixed below the minimum inter-site distance.
ans_ind <- GeoTestsupp_space(
  data = z,
  coordx = coords,
  start = start,
  fixed = fixed,
  h0 = NULL,
  lower = lower,
  upper = upper,
  B = 199,
  likelihood = "Full",
  type = "Standard",
  parallel = FALSE,
  seed = 654
)

## End(Not run)
```

`GeoVarest` *Score-based variance estimation for* `GeoFit` *objects*

Description

The function updates a fitted `GeoFit` object by estimating the variability matrix of the composite likelihood score through parametric simulation. The fitted model is used to generate `K` independent datasets. For each simulated dataset, the composite likelihood score is evaluated at the original estimate $\hat{\theta}$, without refitting the model. The empirical variance of these simulated scores provides an estimate of the variability matrix J. Together with the sensitivity matrix H, computed by `GeoFit` when `sensitivity = TRUE`, this yields the Godambe sandwich covariance matrix

$$G^{-1} = H^{-1} J H^{-1}.$$

The updated object contains standard errors, Wald confidence intervals, p-values, the estimated matrices J, H^{-1} and G^{-1}, and composite likelihood information criteria based on the penalty $\mathrm{tr}(H^{-1}J)$.

Usage

```
GeoVarest(fit, K = 100, sparse = FALSE,
 method = c("cholesky", "TB", "CE"),
 alpha = 0.95, L = 10000,
 parallel = FALSE, ncores = 6, progress = TRUE, seed = NULL,
 min_success_rate = 0.8)
```

Arguments

`fit`	A fitted object obtained from `GeoFit`. The object must contain the sensitivity matrix, hence `GeoFit` should be called with `sensitivity = TRUE`. Full/Standard likelihood fits are not accepted; use `GeoFit(..., likelihood = "Full", type = "Standard", varest = TRUE)` for Hessian-based standard errors.
`K`	The number of simulations used in the parametric score bootstrap.
`sparse`	Logical; if `TRUE`, then Cholesky decomposition is performed using sparse matrix algorithms.
`method`	String; the method of simulation. The default is `"cholesky"`. For large data sets the options `"TB"` and `"CE"` call approximate simulation methods; see `GeoSimapprox`.
`alpha`	Numeric; the level of the confidence interval.
`L`	Numeric; the number of lines in the turning bands method.
`parallel`	Logical; default `FALSE`. If `TRUE`, the score evaluation step is parallelized.
`ncores`	Positive integer or `NULL`; default `6`. With `parallel=TRUE`, an explicit integer requests that many workers, capped by detected cores and the number of bootstrap jobs. Set `ncores=NULL` for automatic selection, capped at six workers and normally leaving one detected core free.
`progress`	Logical; if `TRUE`, progress information is shown.
`seed`	Optional integer seed for reproducibility of the simulated samples.
`min_success_rate`	Minimum fraction of successful score-bootstrap replications required to return variance estimates. The default is 0.8.

Details

For spatio-temporal fits, the original numeric `coordt` values stored in `fit` are reused. Irregularly spaced times are supported with `method = "cholesky"`; approximate simulation methods retain the restrictions documented in `GeoSimapprox`.

Let $cl(\theta)$ denote the composite log-likelihood and let $U(\theta) = \nabla cl(\theta)$ be the corresponding composite score. The function simulates `K` data sets from the fitted model and evaluates the composite score at the fitted parameter value $\hat{\theta}$. The variability matrix is estimated as

$$\hat{J} = Var\{U_1(\hat{\theta}), \dots, U_K(\hat{\theta})\}.$$

If H is the sensitivity matrix stored in `fit$sensmat`, the inverse Godambe matrix is estimated by

$$\widehat{G^{-1}} = H^{-1}\hat{J}H^{-1}.$$

Standard errors are obtained from the square root of the diagonal of $\widehat{G^{-1}}$.

For composite likelihoods, the penalty used in the information criterion is

$$tr(H^{-1}\hat{J}),$$

and the composite likelihood information criterion is computed as

$$-2cl(\hat{\theta}) + 2tr(H^{-1}\hat{J}).$$

Differently from GeoVarestbootstrap, this function does not refit the model for each simulated data set. It estimates the variability matrix of the score and then computes the sandwich/Godambe covariance matrix.

For stochastic nearest-neighbor fits, the realized retained-pair graph stored in the original fit is reused unchanged for every simulated data set. Thus the score bootstrap estimates variability conditional on the selected pair graph, in agreement with the current implementation described in the stochastic-NN methodology.

Parallel multisession workers are started with the package-library paths of the calling R session, including the library containing the loaded GeoModels installation.

For `method = "TB"` without a copula, parallel workers are persistent and replications are streamed one at a time (simulate, evaluate, release). This keeps peak memory bounded when both `L` and `K` are large while still reusing the selected pair graph and static fitting context within each worker. For fitted copula models, `method="TB"` is simulated through the turning-bands backend of `GeoSimCopula`; the copula path is currently not streamed replicate-by-replicate. On platforms where **future** reports forked multicore execution as safe, GeoModels uses it automatically to reduce worker startup and serialization overhead; otherwise it falls back to multisession with the current GeoModels library path propagated explicitly.

Value

Returns an updated object of class `GeoFit`. The following components are added or updated:

`stderr`	Estimated standard errors obtained from the inverse Godambe matrix.
`varcov`	Estimated inverse Godambe matrix $\widehat{G^{-1}}$.
`godambe`	Estimated Godambe matrix.
`Jmat`	Estimated variability matrix of the composite score.
`Hinv`	Inverse, or generalized inverse, of the sensitivity matrix.
`claic`	Composite likelihood AIC-type criterion.
`clic`	Same value as `claic`.
`clbic`	Composite likelihood BIC-type criterion.
`clic_penalty`	Penalty term $tr(H^{-1}\hat{J})$.
`conf.int`	Wald-type confidence intervals based on the estimated standard errors.
`pvalues`	Wald-type p-values.

scores Matrix of successful bootstrap score evaluations.

score_logCompLik
Composite log-likelihood values corresponding to the successful score evaluations.

score_failures Data frame with failed score evaluations, if any.

Three-dimensional coordinates

For a purely spatial univariate fit with explicit irregular three-dimensional Euclidean coordinates, method = "TB" is supported when no anisopars were used. Method "CE" remains restricted to regular two-dimensional grids. Use method = "cholesky" for bivariate, spatio-temporal, anisotropic, or other unsupported three-dimensional cases.

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 for the fitted objects used as input

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

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

Description

The procedure updates a `GeoFit` object using a classical parametric bootstrap. The fitted model is simulated and refitted for each bootstrap replication; the empirical covariance matrix of the successful parameter estimates is used as the bootstrap covariance estimate. A faster score-bootstrap alternative is `GeoVarest`. Full/Standard likelihood fits are not accepted; obtain their Hessian-based standard errors directly with `GeoFit(..., varest = TRUE)`.

Usage

```
GeoVarestbootstrap(
  fit, K = 100, sparse = FALSE,
  optimizer = NULL, lower = NULL, upper = NULL,
  method = "cholesky", alpha = 0.95, L = 10000,
  parallel = FALSE, ncores = 6, progress = TRUE,
```

```
  seed = NULL, min_success_rate = 0.8
)
```

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; default `FALSE`. If `TRUE` then the estimation step is parallelized
`ncores`	Positive integer or `NULL`; default `6`. With `parallel=TRUE`, an explicit integer requests that many workers, capped by detected cores and the number of bootstrap jobs. Set `ncores=NULL` for automatic selection, capped at six workers and normally leaving one detected core free.
`progress`	Logic; If TRUE then a progress bar is shown.
`seed`	Optional integer seed for reproducibility of the simulated samples.
`min_success_rate`	Minimum fraction of successful bootstrap refits required to return variance estimates. The default is 0.8.

Details

For spatio-temporal fits, the original numeric `coordt` values stored in `fit` are reused. Irregularly spaced times are supported with `method = "cholesky"`; approximate simulation methods retain the restrictions documented in `GeoSimapprox`.

The function updates a `GeoFit` object by refitting each simulated data set. A sensitivity matrix is not required to estimate bootstrap standard errors and confidence intervals. If `fit$sensmat` is available, CLIC and CLBIC penalties are also computed. For stochastic nearest-neighbor fits, the realized retained-pair graph from the original fit is reused unchanged across all bootstrap refits.

Parallel multisession workers are started with the package-library paths of the calling R session, including the library containing the loaded GeoModels installation.

For `method = "TB"` without a copula, parallel workers are persistent and replications are streamed one at a time (simulate, evaluate, release). This keeps peak memory bounded when both `L` and `K` are large while still reusing the selected pair graph and static fitting context within each worker. For fitted copula models, `method="TB"` is simulated through the turning-bands backend of `GeoSimCopula`; the copula path is currently not streamed replicate-by-replicate. On platforms where **future** reports forked multicore execution as safe, GeoModels uses it automatically to reduce worker startup and serialization overhead; otherwise it falls back to multisession with the current GeoModels library path propagated explicitly.

Value

Returns an updated object of class `GeoFit`. The main updated components include `stderr`, `varcov`, `godambe` when invertible, `conf.int`, `pvalues`, and `estimates`. When a compatible `sensmat` is available, `claic/clic`, `clbic`, and `clic_penalty` are also updated.

Three-dimensional coordinates

For a purely spatial univariate fit with explicit irregular three-dimensional Euclidean coordinates, `method = "TB"` is supported when no `anisopars` were used. Method `"CE"` remains restricted to regular two-dimensional grids. Use `method = "cholesky"` for bivariate, spatio-temporal, anisotropic, or other unsupported three-dimensional cases.

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` for the fitted objects used as input

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, numbins_t=NULL, directed=FALSE)
```

Arguments

`data` A numeric vector of length d (a single spatial realisation), or a $d \times d$ matrix (a single realisation on a regular grid), or a $t \times d$ matrix (a single spatio-temporal realisation), or a $d \times d \times t$ array (a single spatio-temporal realisation 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. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.

`coordx_dyn` For dynamic locations, a list with one two- or three-column coordinate matrix per temporal instant. The list length must match `length(coordt)`; `data[[t]]` must have one value per row of `coordx_dyn[[t]]`, in the same order.

`cloud` Logical; if `TRUE` the semivariogram cloud is computed for a univariate purely spatial field. If `FALSE` (default), a binned empirical semivariogram is returned. For large data sets, combine `cloud=TRUE` with `maxdist`, `neighb`, or `subsample` to limit the number of returned pairs.

`distance` String specifying the spatial distance. Default is `"Eucl"` (Euclidean distance). See `GeoFit` for details.

`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` A positive finite numeric maximum spatial distance. In the bivariate case a scalar or a length-3 vector (first marginal, cross, second marginal) is accepted. Use `NULL` to let the function use the available spatial range. See Details.

`neighb` Numeric; an optional positive integer indicating the order of neighborhood (useful for large datasets). Neighborhood pair selection is available for spatial and spatio-temporal semivariograms. In the bivariate case a length-3 vector can be used for the first marginal, cross, and second marginal pair sets. See Details.

`maxtime` A positive finite maximum temporal lag, expressed in the same units as `coordt`, to be considered for spatio-temporal semivariograms. See Details.

`numbins` Historical GeoModels argument controlling the spatial bin grid. It is the number of spatial bin boundaries, so the number of empirical spatial classes is `numbins - 1`. The default is 13 boundaries (12 classes).

`numbins_t`	Optional positive integer giving the number of temporal classes for irregularly spaced `coordt`. If `NULL`, `numbins - 1` classes are used. For regularly spaced times the attainable temporal lags are generated directly from the common spacing and this argument is not used.
`directed`	Logical used only when `neighb` is supplied. The default `FALSE` removes reciprocal nearest-neighbour duplicates so that the empirical semivariogram is based on unordered pairs. Set `TRUE` to retain the directed nearest-neighbour graph used by pairwise composite-likelihood code.
`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 historical `numbins` argument sets the number of spatial bin boundaries; hence `numbins - 1` empirical spatial classes are formed when `cloud=FALSE`.

The `maxdist` argument sets a strictly positive finite maximum spatial distance. If no pair falls below the requested cutoff, a valid empty semivariogram object is returned instead of constructing decreasing bins.

The `maxdist` option can be combined with `neighb` to reduce the number of pairs when handling large datasets, by restricting computations to local neighborhoods. By default reciprocal directed nearest-neighbour edges are deduplicated before binning; `directed=TRUE` retains the directed graph. For chordal and geodesic distances, neighbour ordering is obtained from three-dimensional unit-sphere coordinates and the reported lags are then evaluated in the requested metric.

Spatial and temporal bins are left-closed and right-open, except for the last bin which is closed on the right. Thus a pair exactly at the maximum retained lag is not discarded.

The `maxtime` argument sets the maximum temporal lag considered for spatio-temporal semivariograms. For regularly spaced times, attainable temporal lags are generated in linear memory/time from the common spacing; no $T \times T$ matrix of all time differences is formed. For irregular times a controlled grid of `numbins_t` temporal classes is used. The returned spatio-temporal surface is rectangular on `centers` by `centert`; cells with no valid pairs are returned as `NA` with count zero rather than being removed.

For dynamic sites, the temporal marginal is the empirical $\gamma(0, u)$ and therefore uses only spatially collocated locations (up to numerical tolerance). Nearby but non-collocated locations contribute to the positive- distance space-time surface, not to the temporal marginal. Consequently, in a fully dynamic design with no spatial locations repeated across temporal instants, `variogramt` is `NA` in the affected temporal classes. This is an expected property of the sampling design, not a failure of the space-time variogram: the positive-distance surface $\gamma(h, u)$ remains empirically estimable. In this case `plot.GeoVariogram` labels the panel "Temporal marginal unavailable (no repeated spatial locations)". Repeated spatial locations across times are needed only when the empirical temporal marginal itself is required.

In the bivariate case the two marginal semivariograms and the cross-semivariogram are evaluated on the same spatial bins. The cross-semivariogram uses the classical increment-product estimator on unordered positive-distance pairs. The trivial collocated contribution at lag zero is not mixed into the first positive spatial class. If two dynamic/support coordinate sets are supplied, they must be aligned to define this estimator unambiguously.

The `subsample` and `subsample_t` arguments provide additional control for large datasets by using only a proportion of spatial locations and/or time points. With `grid=TRUE`, the grid is first normalized to explicit spatial locations, so full-grid and subsampled calculations follow the same path.

Missing values `NA`/`NaN` are allowed and pairs involving them are skipped consistently. Infinite observations are rejected.

Value

Returns an object of class `"GeoVariogram"`. The list contains, as applicable:

`bins`	Spatial bin boundaries when `cloud=FALSE`; spatial pair distances when `cloud=TRUE`.
`bint`	Temporal lag representatives for a spatio-temporal variogram.
`bivariate`	Logical indicating a bivariate empirical variogram.
`cloud`	Logical indicating a variogram cloud.
`centers`	Spatial bin centers.
`centert`	Temporal lag representatives used by the rectangular spatio-temporal surface.
`distance`	Spatial distance type used to construct the empirical variogram.
`radius`	Sphere radius used for chordal or geodesic distances.
`grid`	Logical recording whether the original input was supplied as a grid.
`neighb`	Neighborhood order used for pair selection, or `NULL`.
`directed`	Whether reciprocal nearest-neighbour pairs were retained.
`lenbins`	Numbers of pairs in the spatial bins. In the bivariate case this is a two-row matrix.
`lenbinst`	Numbers of pairs in the cross/spatio-temporal bins. For space-time objects this follows the same row-major ordering as `variogramst`.
`lenbint`	Numbers of pairs in the temporal bins.
`maxdist`	Maximum spatial distance requested by the user.

`maxtime`	Maximum temporal lag requested by the user.
`numbins_t`	Requested number of temporal classes for irregular times, or `NULL`.
`regular`	Logical indicating regularly spaced temporal coordinates for a space-time object.
`time.breaks`	Internal temporal bin boundaries used for space-time binning.
`spacetime_dyn`	Logical indicating dynamic spatial coordinates.
`temporal.margin`	For space-time objects, a label indicating that the temporal margin is based on same-site/collocated pairs.
`subsample`	Spatial subsampling proportion.
`subsample_t`	Temporal subsampling proportion.
`variograms`	Empirical spatial semivariogram; a two-row matrix in the bivariate case.
`variogramst`	Empirical cross-semivariogram in the bivariate case, or the rectangular spatio-temporal surface stored in row-major order. Empty cells are `NA`.
`variogramt`	Empirical temporal marginal semivariogram.
`type`	Type of empirical variogram.

Spatio-temporal ordering

For fixed sites, `data` is a $T \times N$ matrix with times in rows and sites in columns, and is internally read in the order `c(t(data))`. For dynamic sites, `data` and `coordx_dyn` are aligned lists; the function concatenates complete temporal blocks in list order. See `GeoModels-spacetime-ordering`.

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` for model fitting, `GeoWLS` for weighted least-squares estimation from empirical variograms, `GeoCovariogram` for fitted covariance and variogram values, `plot.GeoVariogram` for plotting empirical variograms.

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.

Three-dimensional coordinates

With three-dimensional Euclidean coordinates, lag distances use all three coordinates, but direction classes are azimuths in the xy plane. Distances `"Geod"` and `"Chor"` require exactly two coordinate columns. The automatic maximum lag includes the extent of the z coordinate.

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 Gaussian random fields*

Description

the function returns the parameter estimates of a Gaussian random field obtained by the weighted 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, optimizer='Nelder-Mead',
 numbins=NULL, radius=1, start=NULL, weighted=FALSE,optimization=TRUE,
 numbins_t=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 an $(d \times d \times t \times n)$-array (a single spatio-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 one dimension of temporal coordinates. Optional argument; the default is `NULL`, in which case a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
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 GeoFit for 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 GeoFit for details.
neighb	Numeric; an optional positive integer indicating the order of neighborhood. See GeoFit for details
maxtime	Numeric; an optional positive maximum temporal lag, expressed in the same units as coordt. See GeoFit for details.
optimizer	String; the optimization algorithm (see optim for details). 'Nelder-Mead' is the default.
numbins	Historical GeoModels argument giving the number of spatial bin boundaries; numbins - 1 empirical spatial classes are used. See Details.
numbins_t	Optional positive integer giving the number of temporal classes for irregularly spaced coordt. If NULL, numbins - 1 is used. Regularly spaced times use their directly attainable temporal lags.
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

GeoWLS is defined for Gaussian random fields and therefore does not expose a model argument. Its native least-squares kernels compare the empirical variogram with the Gaussian variogram parameterized by nugget, sill, and the selected correlation model.

The historical numbins parameter gives the number of spatial bin boundaries, so numbins - 1 spatial classes are used. For spatio-temporal data, regular temporal coordinates are handled from their common spacing without constructing all pairwise temporal differences; irregular temporal coordinates use numbins_t temporal classes.

Empty empirical cells are retained in the rectangular spatial-temporal lag grid and are skipped by the least-squares objective through their zero pair count. This preserves the correspondence between empirical cells and their spatial/temporal lags.

The maxdist parameter indicates the positive finite maximum distance below which pairs are considered in the (weighted) 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;
numbins_t	The requested number of temporal classes for irregular time coordinates, or NULL;
message	Extra message passed from the numerical routines;
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 spatio-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*. Springer-Verlag, 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 one dimension of temporal coordinates. Optional argument; the default is `NULL`, in which case a spatial random field is expected. Temporal coordinates may be irregularly spaced; temporal lags are computed from the supplied coordinate values.
`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 spatio-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 returns 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 spatio-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 spatio-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 bivariate semivariograms*

Description

Plot an empirical spatial, spatio-temporal, or bivariate semivariogram returned by `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

Spatial objects are shown as a standard lag plot. Spatio-temporal objects show the empirical surface and its spatial and temporal margins. Bivariate objects show the two marginal semivariograms and the cross-semivariogram; missing empirical bins are handled as `NA`.

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 embedding simulation on a regular spatial grid*

Description

Internal subroutine used by `GeoSimapprox` to simulate a Gaussian random field on an exact, regular two-dimensional grid by circulant embedding.

Usage

```
SimCE(M, N, x, y, z = NULL, corrmodel, param,
      mean.val = 0, max.ext = 1L)
```

Arguments

`M`	Integer; number of coordinates in `x`.
`N`	Integer; number of coordinates in `y`.
`x`	A finite, strictly increasing, equally spaced numeric vector of length `M`.
`y`	A finite, strictly increasing, equally spaced numeric vector of length `N`.
`z`	Must be `NULL`. Three-dimensional CE grids are not implemented.
`corrmodel`	String; the name of a spatial correlation model.
`param`	A list of correlation parameters. `sill` is the marginal variance and `nugget` is the nugget proportion.
`mean.val`	A finite scalar or a vector with one value per grid point.
`max.ext`	Positive integer; maximum number of successive doubling attempts for the embedding.

Details

The simulated coordinates are exactly `expand.grid(x, y)`. The method uses Euclidean distance. If the embedding still has substantial negative eigenvalues at the maximum extension, they are truncated to zero and a warning is issued.

Value

A list containing `X`, the simulated values, `grid.points`, the exact grid coordinates, `k0`, the selected extension attempt, `approx.flag`, and the minimum embedding eigenvalue.

Author(s)

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

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 an **sp** or **spacetime** object and returns information used by GeoModels functions. Spatial objects should have a coordinate reference system (CRS) defined; if the CRS is missing, `sp2Geo` issues a warning because projected/geographic status cannot be determined reliably. For `STFDF` and `STIDF` objects, the current conversion represents the temporal positions by the sequential indices `1, 2, ..., T`; the original spacing of an irregular time index is therefore not preserved. To retain irregular temporal distances, use the explicit-coordinate interfaces of the relevant GeoModels functions and supply the numeric `coordt` values directly rather than passing `spobj`.

Value

A list with spatio-temporal information

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

```
if (requireNamespace("sp", quietly = TRUE)) {
# 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))

crs <- sp::CRS("EPSG:4326")
obj1 <- sp::SpatialPoints(coords, proj4string = crs)
obj2 <- sp::SpatialPointsDataFrame(
  coords, data = data.frame(X, Y), proj4string = crs
)

# 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 one dimension of temporal coordinates. Temporal coordinates may be irregularly spaced; temporal lags used during initialization are computed from the supplied coordinate values.
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 Details for the accepted options.
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 Details for more information.
maxtime	Numeric; an optional non-negative maximum temporal lag, expressed in the same units as coordt, considered in the composite-likelihood initialization.
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	Low-level compatibility flag. The GeoFit and WlsStart workflows always use TRUE; direct use is not recommended.
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 "bernoulli" (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 one dimension of temporal coordinates. Temporal coordinates may be irregularly spaced; temporal lags used during initialization are computed from the supplied coordinate values.
`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 non-negative maximum temporal separation, expressed in the same units as coordt, considered in the composite-likelihood initialization.
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	Deprecated compatibility argument. WlsStart always precomputes the selected pair structure; FALSE produces a warning and is treated as TRUE.
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 "bernoulli" (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