Package ‘CompRandFld’

May 21, 2013

Version 1.0.3

Date 2013-05-20

Title Composite-likelihood based Analysis of Random Fields

Author Simone Padoan <simone.padoan@unibocconi.it>, Moreno Bevilacqua <moreno.bevilacqua@uv.cl>

Maintainer Simone Padoan <simone.padoan@unibocconi.it>

Depends R (>= 2.12.0)

Description The aim of this package is to collect a set of procedures for the analysis of Random Fields by Composite Likelihood methods. Spatial analysis often involves dealing with large dataset. Therefore even simple studies may be too computationally demanding. Composite likelihood based methods are emerging as useful tools for mitigating such computational problems and show satisfactory results when compared with other techniques such as, for example the tapering method. Moreover, composite likelihood (and related quantities) have some good properties similar to those of the standard likelihood.

Suggests RandomFields, spam, scatterplot3d, fields, mapproj

License GPL (>= 2)

URL http://cran.r-project.org/web/packages/CompRandFld/index.html,

https //r-forge.r-project.org/projects/comprandfld/,

http //faculty.unibocconi.it/simonepadoan

Repository CRAN

Archs x86_64

R topics documented:

`CheckCorrModel` *Checking Correlation Model*

Description

Subroutine called by InitParam. The procedure controls if the correlation model inserted is correct.

Usage

```
CheckCorrModel(corrmodel)
```

Arguments

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

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, `http://faculty.unibocconi.it/simonepadoan`; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, `http://staff.deuv.cl/bevilacqua/`.

See Also

`FitComposite`

`CheckInput` *Checking Input*

Description

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

Usage

```
CheckInput(coordx, coordy, coordt, corrmodel, data, distance, fcall, fixed,
           grid, likelihood, margins, maxdist, maxtime, model, numblock,
           optimizer, param, replicates, start, taper, tapsep, threshold, type,
           varest, vartype, weighted)
```

Arguments

`coordx`	A numeric (d×2)-matrix (where `d` is the number of points) assigning 2-dimensions of coordinates or a numeric vector assigning 1-dimension of coordinates.
`coordy`	A numeric vector assigning 1-dimension of coordinates; `coordy` is interpreted only if `coordx` is a numeric vector otherwise it will be ignored.
`coordt`	A numeric vector assigning 1-dimension of temporal coordinates.
`corrmodel`	String; the name of a correlation model, for the description see `FitComposite`.
`data`	A numeric vector or a $(n \times d)$-matrix or $(d \times d \times n)$-matrix of observations.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details**.
`fcall`	String; `Fitting` to call the fitting procedure and `simulation` to call the simulation.
`fixed`	A named list giving the values of the parameters that will be considered as known values. The listed parameters for a given correlation function will be not estimated, i.e. if `list(nugget=0)` the nugget effect is ignored.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as a vector or a $(n \times d)$-matrix, instead if `TRUE` then $(d \times d \times n)$-matrix is considered.
`likelihood`	String; the configuration of the composite likelihood. `Marginal` is the default.
`margins`	String; the type of the marginal distribution of the max-stable field.
`maxdist`	Numeric; an optional positive value indicating the maximum spatial distance considered in the composite-likelihood computation.
`maxtime`	Numeric; an optional positive value indicating the maximum temporal lag separation in the composite-likelihood.
`model`	String; the density associated to the likelihood objects. `Gaussian` is the default.
`numblock`	Numeric; the number of observation of the underlying randfom field (only for max-stable simulations). See `RFsim`.
`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 `RFsim`.

`replicates`	Logical; if `FALSE` (the default) one spatial random field is considered, instead if `TRUE` the data are considered as iid replicates of a field.
`start`	A named list with the initial values of the parameters that are used by the numerical routines in maximization procedure. `NULL` is the default.
`taper`	String; the name of the tapered correlation function.
`tapsep`	Numeric; an optional value indicating the separabe parameter in the space time quasi taper (see **Details**).
`threshold`	Numeric; a value indicating a threshold for the binary random field.
`type`	String; the type of the likelihood objects. If `Pairwise` (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods.
`varest`	Logical; if `TRUE` the estimate' variances and standard errors are returned. `FALSE` is the default.
`vartype`	String; the type of estimation method for computing the estimate variances, see `FitComposite`.
`weighted`	Logical; if `TRUE` the likelihood objects are weighted. If `FALSE` (the default) the composite likelihood is not weighted.

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, `http://faculty.unibocconi.it/simonepadoan`; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, `http://staff.deuv.cl/bevilacqua/`.

See Also

`FitComposite`

`CheckLikelihood` *Checking Composite-likelihood Type*

Description

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

Usage

```
CheckLikelihood(likelihood)
```

Arguments

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

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, `http://faculty.unibocconi.it/simonepadoan`; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, `http://staff.deuv.cl/bevilacqua/`.

See Also

`FitComposite`

`CheckModel` *Checking Random Field type*

Description

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

Usage

```
CheckModel(model)
```

Arguments

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

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

See Also

FitComposite

`CheckType` *Checking Likelihood Objects*

Description

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

Usage

```
CheckType(type)
```

Arguments

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

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

See Also

`FitComposite`

CheckVarType — *Checking Variance Estimates Type*

Description

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

Usage

```
CheckVarType(type)
```

Arguments

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

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, `http://faculty.unibocconi.it/simonepadoan`; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, `http://staff.deuv.cl/bevilacqua/`.

See Also

`FitComposite`

CompLikelihood — *Optimizes the Composite log-likelihood*

Description

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

Usage

```
CompLikelihood(coordx, coordy, corrmodel, data, distance,
               flagcorr, flagnuis, fixed, grid, likelihood,
               lower, model, namescorr, namesnuis, namesparam,
               numparam, numparamcorr, optimizer, param,
               spacetime, threshold, type, upper, varest,
               vartype, winconst, winstp)
```

Arguments

coordx	A numeric ($d \times 2$)-matrix (where d is the number of points) assigning 2-dimensions of coordinates or a numeric vector assigning 1-dimension of coordinates.
coordy	A numeric vector assigning 1-dimension of coordinates; coordy is interpreted only if coordx is a numeric vector otherwise it will be ignored.
corrmodel	Numeric; the id of the correlation model.
data	A numeric vector or a ($n \times d$)-matrix or ($d \times d \times n$)-matrix of observations.
distance	String; the name of the spatial distance. The default is Eucl, the euclidean distance. See the Section **Details**.
flagcorr	A numeric vector of binary values denoting which paramerters of the correlation function will be estimated.
flagnuis	A numeric vector of binary values denoting which nuisance paramerters will be estimated.
fixed	A numeric vector of parameters that will be considered as known values.
grid	Logical; if FALSE (the default) the data are interpreted as a vector or a ($n \times d$)-matrix, instead if TRUE then ($d \times d \times n$)-matrix is considered.
likelihood	String; the configuration of the compositelikelihood, see FitComposite.
lower	A numeric vector with the lower bounds of the parameters' ranges.
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.
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.
param	A numeric vector of parameters' values.
spacetime	Logical; if TRUE the random field is spatial-temporal otherwise is a spatial field.
threshold	Numeric; a value indicating a threshold for the binary random field, see FitComposite.
type	String; the type of the likelihood objects. If Pairwise (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods.
upper	A numeric vector with the upper bounds of the parameters' ranges.
varest	Logical; if TRUE the estimate' variances and standard errors are returned. FALSE is the default.
vartype	String; the type of estimation method for computing the estimate variances, see FitComposite.
winconst	Numeric; a positive value – if vartype=SubSamp – determines the window size in the sub-sampling estimates of the variances, see FitComposite.
winstp	Numeric; a positive value – if vartype=SubSamp – determines the window step in the sub-sampling estimates of the variances, see FitComposite.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

See Also

FitComposite

CorrelationParam *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

```
CorrelationParam(corrmodel)
```

Arguments

corrmodel	String; the name of a correlation model. See Covmatrixfor details

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

See Also

FitComposite

Covariogram *Computes covariance, variogram and extremal coefficient functions*

Description

The procedure computes and/or plots the covariance, the variogram or the extremal coefficient functions and the practical range estimated fitting a Gaussian or max-stable random field with the composite-likelihood or using the weighted least square method. Allows to add to the variogram or extremal coefficient plots the empirical estimates.

Usage

```
Covariogram(fitted, lags=NULL, lagt=NULL, answer.cov=FALSE, answer.vario=FALSE,
            answer.extc=FALSE, answer.range=FALSE, fix.lags=NULL, fix.lagt=NULL,
            show.cov=FALSE, show.vario=FALSE, show.extc=FALSE, show.range=FALSE,
            add.cov=FALSE, add.vario=FALSE, add.extc=FALSE, pract.range=95,
            vario, ...)
```

Arguments

`fitted`	A fitted object obtained from the `FitComposite` or `WLeastSquare` procedures.
`lags`	A numeric vector of distances.
`lagt`	A numeric vector of temporal separations.
`answer.cov`	Logical; if `TRUE` a vector with the estimated covariance function is returned; if `FALSE` (the default) the covariance is not returned.
`answer.vario`	Logical; if `TRUE` a vector with the estimated variogram is returned; if `FALSE` (the default) the variogram is not returned.
`answer.extc`	Logical; if `TRUE` a vector with the estimated extremal coefficient is returned; if `FALSE` (the default) the variogram is not returned.
`answer.range`	Logical; if `TRUE` the estimated pratical range is returned; if `FALSE` (the default) the pratical range is not returned.
`fix.lags`	Integer; a positive value denoting the spatial lag to consider for the plot of the temporal profile.
`fix.lagt`	Integer; a positive value denoting the temporal lag to consider for the plot of the spatial profile.
`show.cov`	Logical; if `TRUE` the estimated covariance function is plotted; if `FALSE` (the default) the covariance function is not plotted.
`show.vario`	Logical; if `TRUE` the estimated variogram is plotted; if `FALSE` (the default) the variogram is not plotted.
`show.extc`	Logical; if `TRUE` the estimated extremal coefficient is plotted; if `FALSE` (the default) the extremal coefficient is not plotted.
`show.range`	Logical; if `TRUE` the estimated pratical range is added on the plot; if `FALSE` (the default) the pratical range is not added.
`add.cov`	Logical; if `TRUE` the vector of the estimated covariance function is added on the current plot; if `FALSE` (the default) the covariance is not added.
`add.vario`	Logical; if `TRUE` the vector with the estimated variogram is added on the current plot; if `FALSE` (the default) the correlation is not added.
`add.extc`	Logical; if `TRUE` the vector with the estimated extremal coefficient is added on the current plot; if `FALSE` (the default) the correlation is not added.
`pract.range`	Numeric; the percent of the sill to be reached.
`vario`	A `Variogram` object obtained from the `EVariogram` procedure.
`...`	other optional parameters which are passed to plot functions.

Value

The returned object is eventually a list with:

`covariance`	The vector of the estimated covariance function;
`variogram`	The vector of the estimated variogram function;
`extrcoeff`	The vector of the estimated extremal coefficient function;
`pratical.range`	The estimated practial range.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

References

Cooley, D., Naveau, P. and Poncet, P. (2006) *Variograms for spatial max-stable random fields*. Dependence in Probability and Statistics, p. 373–390.

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

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

Smith, R. L. (1990) Max-Stable Processes and Spatial Extremes. *Unpublished manuscript*, University of North California.

See Also

FitComposite, WLeastSquare

Examples

```
library(CompRandFld)
library(RandomFields)
library(scatterplot3d)
set.seed(31231)

# Set the coordinates of the points:
x <- runif(100, 0, 10)
y <- runif(100, 0, 10)


################################################################
###
### Example 1. Plot of covariance and variogram functions
### estimated from a Gaussian random field with exponent
### correlation. One spatial replication is simulated.
###
###
###############################################################

# Set the model's parameters:
corrmodel <- "exponential"
mean <- 0
sill <- 1
nugget <- 0
scale <- 3

# Simulation of the Gaussian random field:
data <- RFsim(x, y, corrmodel=corrmodel, param=list(mean=mean,
              sill=sill, nugget=nugget, scale=scale))$data

# Maximum composite-likelihood fitting of the Gaussian random field:
fit <- FitComposite(data, x, y, corrmodel=corrmodel,
                    type='Difference')

# Empirical estimation of the variogram:
```

```
vario <- EVariogram(data, x, y)

# Plot of covariance and variogram functions:
par(mfrow=c(1,2))
Covariogram(fit, show.cov=TRUE, show.range=TRUE,
            show.vario=TRUE, vario=vario)


################################################################
##
### Example 2. Plot of covariance and extremal coefficient
### functions estimated from a max-stable random field with
### exponential correlation. n idd spatial replications are
### simulated.
###
###############################################################

set.seed(1126)
# Simulation of the max-stable random field:
data <- RFsim(x, y, corrmodel=corrmodel, model="ExtGauss", replicates=20,
              param=list(mean=mean,sill=sill,nugget=nugget,scale=scale))$data

# Maximum composite-likelihood fitting of the max-stable random field:
fit <- FitComposite(data, x, y, corrmodel=corrmodel, model='ExtGauss',
                    replicates=20, varest=TRUE, vartype='Sampling',
                    margins="Frechet")

data <- Dist2Dist(data, to='sGumbel')

# Empirical estimation of the madogram:
vario <- EVariogram(data, x, y, type='madogram', replicates=20)

# Plot of correlation and extremal coefficient functions:
par(mfrow=c(1,2))
Covariogram(fit, show.cov=TRUE, show.range=TRUE, show.extc=TRUE,
            vario=vario, pract.range=84)


################################################################
###
### Example 3. Plot of covariance and variogram functions
### estimated from a Gaussian spatio-temporal random field with
### double-exp correlation.
### One spatio-temporal replication is simulated.
###
###############################################################

# Define the spatial-coordinates of the points:
x <- runif(20, 0, 1)
y <- runif(20, 0, 1)
# Define the temporal sequence:
time <- seq(0, 50, 1)

# Simulation of the spatio-temporal Gaussian random field:
data <- RFsim(x, y, time, corrmodel="exp_exp",param=list(mean=mean,
              nugget=nugget,scale_s=0.5,scale_t=1,sill=sill))$data
```

```
# Maximum composite-likelihood fitting of the space-time Gaussian random field:
fit <- FitComposite(data, x, y, time, corrmodel="exp_exp", maxtime=5,
                    likelihood="Marginal",type="Pairwise", fixed=list(
                    nugget=nugget, mean=mean),start=list(scale_s=0.2,
                    scale_t=1, sill=sill))

# Empirical estimation of spatio-temporal covariance:
vario <- EVariogram(data, x, y, time, maxtime=10)

# Plot of the fitted space-time covariace
Covariogram(fit,show.cov=TRUE)

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

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

################################################################
###
### Example 4. Plot of parametric and empirical lorelograms
### estimated from a Binary Gaussian random fields with
### exponential correlation. One spatial replication is
### simulated.
###
###############################################################

set.seed(1240)

# Define the spatial-coordinates of the points:
x <- seq(0,3, 0.1)
y <- seq(0,3, 0.1)

# Simulation of the Binary Gaussian random field:
data <- RFsim(x, y, corrmodel=corrmodel, model="BinaryGauss",
              threshold=0,param=list(nugget=nugget,mean=mean,
              scale=.6,sill=0.8))$data

# Maximum composite-likelihood fitting of the Binary Gaussian random field:
fit <- FitComposite(data, x, y, corrmodel=corrmodel, model="BinaryGauss",
                    maxdist=0.8, likelihood="Marginal", type="Pairwise",
                    start=list(mean=mean,scale=0.1,sill=0.1))

# Empirical estimation of the lorelogram:
vario <- EVariogram(data, x, y, type="lorelogram", maxdist=2)

# Plot of fitted and empirical lorelograms:
Covariogram(fit, vario=vario, show.vario=TRUE, lags=seq(0.1,2,0.1))
```

Covmatrix *Spatio-temporal (tapered) Covariance Matrix*

Description

The function computes the (tapered) covariance matrix for a spatial (temporal or spatio-temporal) covariance model and a set of spatial (temporal or spatio-temporal) points.

Usage

```
Covmatrix(coordx, coordy=NULL, coordt=NULL, corrmodel, distance="Eucl",
          grid=FALSE, iskrig=FALSE, maxdist=NULL, maxtime=NULL, param ,
          taper=NULL, tapsep=NULL, type="Standard")
```

Arguments

`coordx`	A numeric ($d \times 2$)-matrix (where `d` is the number of spatial sites) giving 2-dimensions of spatial coordinates or a numeric d-dimensional vector giving 1-dimension of spatial coordinates.
`coordy`	A numeric vector giving 1-dimension of spatial coordinates; `coordy` is interpreted only if `coordx` is a numeric vector or `grid=TRUE` otherwise it will be ignored. Optional argument, the default is `NULL` then `coordx` is expected to be numeric a ($d \times 2$)-matrix.
`coordt`	A numeric vector giving 1-dimension of temporal coordinates. At the moment implemented only for the Gaussian case. Optional argument, the default is `NULL` then a spatial random field is expected.
`corrmodel`	String; the name of a correlation model, for the description see the Section **Details**.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See `FitComposite`.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as spatial or spatial-temporal realisations on a set of non-equispaced spatial sites (irregular grid). See `FitComposite`.
`iskrig`	Logical: the default value is `FALSE`. It is `TRUE` if the function is called by the function `Kri`.
`maxdist`	Numeric; an optional positive value indicating the marginal spatial compact support. See `FitComposite`.
`maxtime`	Numeric; an optional positive value indicating the marginal temporal compact support. See `FitComposite`.
`param`	A list of parameter values required for the correlation model. See `FitComposite` and `CorrelationParam`.
`taper`	String; the name of the taper correlation function if type is `Tapering`, see the Section **Details**.
`tapsep`	Numeric; an optional value indicating the separabe parameter in the space-time quasi taper (see **Details**).
`type`	String; the type of covariance matrix `Standard` (the default) or `Tapering` for tapered covariance matrix

Details

The parameter `param` is a list including all the parameters of a covariance function model.
In particular, the covariance models share the following paramaters: the `sill` that represents the common variance of the random field, the `nugget` that represents the local variation (white noise) at the origin. For each correlation model you can check the list of the specific parameters using `CorrelationParam`.
Here there is the list of all the implemented space and space-time correlation models. The list of space-time correlation functions includes separable and non-separable models.

- Purely spatial correlation models:
  1. `cauchy`
  $$R(h) = \left(1 + h^2\right)^{-\beta}$$
  The parameter β is positive. It is a special case of the `gencauchy` model.
  2. `exponential`
  $$R(h) = e^{-h}, \quad h \geq 0$$
  This model is a special case of the `whittle` and the `stable` model.
  3. `gauss`
  $$R(h) = e^{-h^2}$$
  This model is a special case of the `stable` model.
  4. `gencauchy` (generalised `cauchy`)
  $$R(h) = (1 + h^\alpha)^(- \beta/\alpha)$$
  The parameter α is in (0,2], and β is positive.
  5. `spherical`
  $$R(h) = (1 - 1.5h + 0.5h^3)1_{[0,1]}(h)$$
  This isotropic covariance function is valid only for dimensions less than or equal to 3.
  6. `stable`
  $$R(h) = e^{-h^\alpha}, \quad h \geq 0$$
  The parameter α is in $(0, 2]$.
  7. `wave`
  $$R(h) = \frac{\sin h}{h}, \quad h > 0 \qquad \text{and } R(0) = 1$$
  This isotropic covariance function is valid only for dimensions less than or equal to 3.
  8. `matern`
  $$R(h) = 2^{1-\nu}\Gamma(\nu)^{-1}x^\nu K_\nu(h)$$
  The parameter ν is positive.
  This is the model of choice if the smoothness of a random field is to be parametrised: if $\nu > m$ then the graph is m times differentiable.
- Spatio-temporal correlation models:
  - Non-separable models:
    1. `gneiting` (non-separabel space time model)
    $$R(h, u) = exp(-h^\nu/(1 + u^\lambda)^(0.5sep\nu))/(1 + u^\lambda)$$
    The parameters ν and λ take values in $[0, 2]$; the parameter sep take values in $[0, 1]$. For $sep = 0$ it is a separable model.
    2. `gneiting_GC` (non-separabel space time model with great circle distances)
    $$R(h, u) = exp(-u^\lambda/(1 + h^\nu)^(0.5sep\lambda))/(1 + h^\nu);$$
    3. `iacocesare` (non-separabel space time model)
    $$R(h, u) = (1 + h^\nu + u^\lambda)^{-\delta}$$
    The parameters ν and λ take values in $[1, 2]$; the parameters δ must be greater than or equal to half the space-time dimension.

4. `porcu` (non-separabel space time model)

$$R(h,u) = (0.5(1+h^{\nu})^(sep) + 0.5(1+u^{\lambda})^(sep))^(-1/sep)$$

   The parameters ν and λ take values in $[0,2]$; the paramete sep take values in $[0,1]$. The limit of the correlation model as sep tends to zero leads to a separable model.

5. `porcu2` (non-separabel space time model)

$$R(h,u) = exp(-h^{\nu}(1+u^{\lambda})^(0.5sep\nu))/(1+u^{\lambda})^(1.5)$$

   The parameters ν and λ take values in $[0,2]$; the parameter sep take values in $[0,1]$. For $sep = 0$ it is a separable model.

– Separable models.

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

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

  Several combinations are possible:
  1. `exp_exp`: spatial exponential model and temporal exponential model
  2. `exp_cauchy`: spatial exponential model and temporal cauchy model
  3. `matern_cauchy`: spatial matern model and temporal cauchy model
  4. `stable_stable`: spatial stabel model and temporal stable model

  Note that some models are nested. (The `exp_exp` with the `stable_stable` for instance.)

• Spatial taper function models.

  For spatial covariance tapering the tapered correlation functions are:
  1. `Wendland1`
$$R(h) = (1-h)^2(1+0.5h)1_{[0,1]}(h)$$
  2. `Wendland2`
$$R(h) = (1-h)^4(1+4h)1_{[0,1]}(h)$$
  3. `Wendland3`
$$R(h) = (1-h)^6(1+6h+35h^2/3)1_{[0,1]}(h)$$

• Spatio-tempora tapered correlation models.

  For space-time covariance tapering likelihood the taper functions are obtained as the product of a spatial and a temporal taper (Separable taper). Several combinations are possible:
  – `Wendlandi_Wendlandj`: spatial `Wendlandi` taper and temporal `Wendlandj` taper with i,j=1,2,3.

• Space-time non separable quasi-taper with dynamically space-time compact support is:
  – `qt_time` and `qt_space`. In The case of `qt_time` the space-time quasi taper is:

$$T(h,u) = (arg)^(-6)(1+7x)(1-x)^7 1_{[0,maxtime/arg]}(u)$$

$$arg = (1+h/maxdist)^{\beta}, x = uarg/maxtime$$

  where $0 <= \beta <= 1$ is a fixed parameter of separability (`tapsep`), `maxtime` the fixed temporal compact support and `maxdist` the fixed spatial scale parameter. The quasi-taper `qt_space` is the same taper but changing the time with the space.

Remarks:

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

$$C(h) = sill + nugget, if h = 0$$

$$C(h) = sill * R(h/scale, , ...), if h > 0$$

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

$$C(h, u) = sill + nugget, if h = 0, u = 0$$

$$C(h, u) = sill * R(h/scale_s, u/scale_t, ...), if h > 0 or u > 0$$

Here '...' stands for additional parameters.

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

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

Similarly if `R(h,u)` is a spatio-temporal taper given in standard notation, then the taper function applied with arbitrary compact supports (maxdist, maxtime) equals to:

$$T(h, u) = R(h/maxdist, u/maxtime)$$

Then the tapered covariance matrix is obtained as:

$$C_t ap(h, u) = T(h, u)C(h, u)$$

Value

Returns an object of class `CovMat`. An object of class `CovMat` is a list containing at most 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;
- `covmatrix` — The covariance matrix if `type` is`Standard`. An object of class spam if `type` is `Tapering`
- `corrmodel` — String: the correlation model;
- `distance` — String: the type of spatial distance;
- `grid` — Logical:`TRUE` if the spatial data are in a regular grid, otherwise `FALSE`;
- `nozero` — In the case of tapered matrix the percentage of non zero values in the covariance matrix. Otherwise is NULL.
- `maxdist` — Numeric: the marginal spatial compact support if `type` is `Tapering`;
- `maxtime` — Numeric: the marginal temporal compact support if `type` is `Tapering`;

`namescorr`	String: The names of the correlation parameters;
`numcoord`	Numeric: the number of spatial coordinates;
`numtime`	Numeric: the number the temporal coordinates;
`param`	Numeric: The covariance parameters;
`tapmod`	String: the taper model if `type` is `Tapering`. Otherwise is NULL.
`spacetime`	`TRUE` if spatio-temporal and `FALSE` if spatial covariance model;

In the space-time case `covmatrix` is the covariance matrix of the random vector

$$Z(s_1, t_1), Z(s_1, t_2), ..Z(s_n, t_1), .., Z(s_n, t_m)$$

for `n` spatial locatione sites and `m` temporal instants.

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, http://staff.deuv.cl/bevilacqua/.

References

Bevilacqua, M., Mateu, J., Porcu, E., Zhang, H. and Zini, A. (2010). Weighted composite likelihood-based tests for space-time separability of covariance functions. *Statistics and Computing*, **20(3)**, 283-293.

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

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

Gneiting, T., Genton, M. G. and Guttorp, P. (2007). *Geostatistical space-time models, stationarity, separability and full symmetry*. In Finkenstadt, B., Held, L. and Isham, V. (eds.), Statistical Methods for Spatio-Temporal Systems, Chapman & Hall/CRC, Boca Raton, pp. 151-175

Schlather, M. (1999) *An introduction to positive definite functions and to unconditional simulation of random fields*. Technical report ST 99–10, Dept. of Maths and Statistics, Lancaster University

See Also

`Kri`, `RFsim`, `FitComposite`

Examples

```
library(CompRandFld)
library(spam)

################################################################
###
### Example 1. Covariance matrix associated to
### a Matern correlation model
###
###############################################################

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

```
matrix1 <- Covmatrix(x, y, corrmodel="matern", param=list(smooth=0.5,
                    sill=1,scale=0.2))
dim(matrix1$covmatrix)

################################################################
###
### Example 2. Tapered Covariance matrix associated to
### a Matern correlation model
###
###############################################################

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

matrix2 <- Covmatrix(x, y, corrmodel="matern", param=list(smooth=0.5,
                    sill=1,scale=0.2),maxdist=0.3,taper="Wendland1",
                    type="Tapering")
# Tapered covariance matrix
as.matrix(matrix2$covmatrix)[1:15,1:15]

# Percentage of no zero values in the tapered matrix
matrix2$nozero

################################################################
###
### Example 3. Covariance matrix associated to
### a space-time double exponential correlation model
###
###############################################################

# Define the temporal-coordinates:
times <- seq(1, 5, 1)

# Define correlation model
corrmodel="exp_exp"

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

# Simulation of a spatial Gaussian random field:
matrix3 <- Covmatrix(x, y, times, corrmodel=corrmodel,
                     param=param)

dim(matrix3$covmatrix)

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

# Simulation of a spatial Gaussian random field:
matrix4 <- Covmatrix(x, y, times, corrmodel=corrmodel, param=param, maxdist=0.3,
                     maxtime=2,taper="Wendland2_Wendland2",type="Tapering")
```

```
# Tapered space time covariance matrix
as.matrix(matrix4$covmatrix)[1:10,1:10]
```

`Dist2Dist` *Switches from an EV to Another EV Distribution*

Description

The function transforms observations belonging to the GEV class from one model to another.

Usage

```
Dist2Dist(data, from='Gev', to='sFrechet', loc=NULL, scale=NULL, shape=NULL)
```

Arguments

`data`	A numeric vector or a matrix of extreme values.
`from`	The name of the original extreme value distribution, i.e. `Gev` (the default), see the **Details** section.
`to`	The name of the desired extreme value distribution, i.e. `sFrechet` (the default), see the **Details** section.
`loc`	A numeric value or vector of location parameters.
`scale`	A numeric value or vector of scale parameters.
`shape`	A numeric value or vector of shape parameters.

Details

If `data` is a numeric vector of length `n` then the dataset is consider as a realisation from an univariate extreme value distribution. Instead, if `data` is a $(n \times d)$-matrix then the columns represent the different variables with extreme value distributions and the rows represent the iid replications. Finally, if `data` is a $(d \times d \times n)$-matrix then the columns and rows represent the different variables and the third dimension represents the iid replications.

The parameters `from` and `to` indicate the original extreme value distribution(s) from which the observations are drawn and the target extreme value distribution(s) that the transformed data will follow. The options are:

1. `from=Gev` (generalised extreme value distribution):
   - `to=Uniform`, which means uniform distribution;
   - `to=sFrechet`, which means standard (or unit) Frechet distribution, that is GEV(1,1,1);
   - `to=sGumbel`, which means standard Gumbel distribution, that is GEV(0,1,1);
   - `to=sWeibull`, which means standard Weibull distribution, that is GEV(1,1,-1);
   - `to=Gev`, which means generalised extreme value distribution. Note, that in this case, it is required to insert vectors of location, scale and shape parameters with dimension `n` in the univariate case, dimension `d` when `data` is $(n \times d)$-matrix and dimension $n \times d$ when `data` is $(d \times d \times n)$-matrix.
2. `from=sFrechet`

- to=Gev.

3. from=sGumbel
   - to=Gev.
4. from=sWeibull
   - to=Gev.

Value

A numeric vector or matrix of transformed values following the desired distribution.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

References

de Haan, L. and Ferreira, A. (2006) *Extreme Value Theory An Introduction*. Springer Verlang, New York.

See Also

FitGev

EVariogram *Empirical Variogram(variants) of Gaussian, Binary and Max-Stable Fields*

Description

The function returns an empirical estimate of the variogram (or its variants) for Gaussian, Binary and max-stable random field.

Usage

```
EVariogram(data, coordx, coordy, coordt=NULL, cloud=FALSE, distance='Eucl',
           grid=FALSE, gev=c(0,1,0), maxdist=NULL, maxtime=NULL, numbins=NULL,
           replicates=1, type='variogram')
```

Arguments

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

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

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

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

`cloud` Logical; if `TRUE` the variogram cloud is computed, otherwise if `FALSE` (the default) the empirical (binned) variogram is returned.

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

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

`gev` A numeric vector with the three GEV parameters;

`maxdist` A numeric value denoting the spatial maximum distance, see the Section **Details**.

`maxtime` A numeric value denoting the temporal maximum distance, see the Section **Details**.

`numbins` A numeric value denoting the numbers of bins, see the Section **Details**.

`replicates` Numeric; a positive integer denoting the number of independent and identically distributed (iid) replications of a spatial or spatial-temporal random field. Optional argument, the default value is 1 then a single realisation is considered.

`type` A String denoting the type of variogram. Four options are available: `variogram`, `madogram`, `Fmadogram` and `lorelogram`. It is returned respectively, the standard variogram with the first (Gaussian responses), the madogram with the second and third (extreme values), the lorelogram with the fourth (Binary data).

Details

We briefly report the definitions of variogram used in this function.

In the case of a spatial Gaussian random field the sample `variogram` estimator is defined by

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

where $N(h)$ is the set of all the sample pairs whose distances fall into a tolerance region with size h (equispaced intervalls are considered). Observe, that in the literature often the above definition is termed semivariogram (see e.g. the first reference). Nevertheless, here this defition has been used in order to be consistent with the variogram defition used for the extremes (see e.g. the third reference).

In the case of a spatial max-stable random field, the sample `madogram` estimator is defined similarly to the Gaussian case by

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

In the case of a spatial binary random field, the sample `lorelogram` estimator (the analogue of the correlation) is defined by

$$\hat{L}(h) = (N_{11}(h)N_{00}(h))/(N_{01}(h)N_{10}(h)).$$

where $N_{11}(h)$ is the number of pairs who are both equal to 1 and that falls in the bin h. Similarly are defined the other quantities.

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

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

where $N(h,u)$ is the set of all the sample pairs whose spatial distances fall into a tolerance region with size h and $|k-l| = u$. Note, that $Z(x_i, l)$ is the observation at site x_i and time l. Taking this in mind and given the above definition of lorelogram, the spatio-temporal extention is straightforward.

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

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

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

Value

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

`bins`	Adjacent intervals of grouped spatial distances if `cloud=FALSE`. Otherwise if `cloud=TRUE` all the spatial pairwise distances;
`bint`	Adjacent intervals of grouped temporal distances if `cloud=FALSE`. Otherwise if `cloud=TRUE` all the temporal pairwise distances;
`cloud`	If the variogram cloud is returned (`TRUE`) or the empirical variogram (`FALSE`);
`centers`	The centers of the spatial bins;
`distance`	The type of spatial distance;
`extcoeff`	The spatial extremal coefficient function. Available only if `type` is equal to `madogram` or `Fmadogram` (for the moment available only for a spatial random field);
`lenbins`	The number of pairs in each spatial bin;
`lenbinst`	The number of pairs in each spatial-temporal bin;
`lenbint`	The number of pairs in each temporal bin;
`srange`	The maximum and minimum spatial distances used for the calculation of the variogram;
`variograms`	The empirical spatial variogram;
`variogramst`	The empirical spatial-temporal variogram;
`variogramt`	The empirical temporal variogram;
`trange`	The maximum and minimum temporal distance used for the calculation of the variogram;
`type`	The type of estimated variogram: the standard variogram or the madogram.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

References

Cooley, D., Naveau, P. and Poncet, P. (2006) *Variograms for spatial max-stable random fields*. Dependence in Probability and Statistics, p. 373–390.

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

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

Heagerty, P. J., and Zeger, S. L. (1998). Lorelogram: A Regression Approach to Exploring Dependence in Longitudinal Categorical Responses. *Journal of the American Statistical Association*, **93**(441), 150–162

Smith, R. L. (1990) Max-Stable Processes and Spatial Extremes. *Unpublished manuscript*, University of North California.

See Also

FitComposite

Examples

```
library(CompRandFld)
library(RandomFields)
set.seed(514)

# Set the coordinates of the sites:
x <- runif(200, 0, 10)
y <- runif(200, 0, 10)


#################################################################
###
### Example 1. Empirical estimation of the variogram from a
### Gaussian random field with exponential correlation.
### One spatial replication is simulated.
###
###
################################################################

# Set the model's parameters:
corrmodel <- "exponential"
mean <- 0
sill <- 1
nugget <- 0
scale <- 3

# Simulation of the spatial Gaussian random field:
data <- RFsim(x, y, corrmodel=corrmodel, param=list(mean=mean,
              sill=sill, nugget=nugget, scale=scale))$data

# Empirical spatial variogram estimation:
fit <- EVariogram(data, x, y)
```

```
# Results:
plot(fit$centers, fit$variograms, xlab='h', ylab=expression(gamma(h)),
     ylim=c(0, max(fit$variograms)), xlim=c(0, fit$srange[2]), pch=20,
     main="variogram")


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

set.seed(331)
# Define the temporal sequence:
times <- seq(1,10,1)

# Simulation of a spatio-temporal Gaussian random field:
data <- RFsim(x, y, times, corrmodel="gneiting",
              param=list(mean=0,scale_s=0.4,scale_t=1,sill=sill,
              nugget=0,power_s=1,power_t=1,sep=0.5))$data

# Empirical spatio-temporal variogram estimation:
fit <- EVariogram(data, x, y, times, maxtime=5,maxdist=4)

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

# Results: Marginal temporal empirical  variogram
plot(fit$bint, fit$variogramt, xlab='t', ylab=expression(gamma(t)),
     ylim=c(0, max(fit$variograms)),xlim=c(0,max(fit$bint)),
     pch=20,main="Marginal temporal Variogram",cex.axis=.8)

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

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

# A smoothed version
par(mai=c(.2,.2,.2,.2),mgp=c(1,.3, 0))
persp(c(0,fit$centers), c(0,fit$bint), st.vario,
      xlab="h", ylab="u", zlab=expression(gamma(h,u)),
```

```
      ltheta=90, shade=0.75, ticktype="detailed", phi=30,
      theta=30,main="Space-time variogram",cex.axis=.8,
      cex.lab=.8)


################################################################
###
### Example 3. Empirical estimation of the madogram from a
### max-stable random field (Extremal Gaussian model) with
### exponential correlation.
### n iid spatial replications are simulated.
###
###############################################################

set.seed(7273)
# Simulation of the max-stable random field:
data <- RFsim(x, y, corrmodel=corrmodel, model="ExtGauss",
              param=list(mean=mean, sill=sill, nugget=nugget,
              scale=scale), replicates=40)$data
# Tranform data from from common unit Frechet to standard Gumbel margins:
data <- Dist2Dist(data, to='sGumbel')

# Empirical madogram estimation:
fit <- EVariogram(data, x, y, type='madogram', replicates=40, cloud=FALSE)

# Results:
par(mfrow=c(1,2), mai=c(.6,.6,.3,.3), mgp=c(1.6,.6, 0))
plot(fit$centers, fit$variograms, xlab='h', ylab=expression(nu(h)),
     ylim=c(0, max(fit$variograms)), xlim=c(0, fit$srange[2]), pch=20,
     main="madogram")
plot(fit$centers, fit$extcoeff, xlab='h', ylab=expression(theta(h)),
     ylim=c(1, 2), xlim=c(0, fit$srange[2]), pch=20,
     main="extremal coefficient")


################################################################
###
### Example 4. Empirical estimation of the lorelogram from a
### spatio-temporal Binary-Gaussian random field with double
### exponential correlation.
### One spatio-temporal replication is simulated.
###
###############################################################

#set.seed(5351)
# Define the temporal sequence:
#times <- seq(1,10,1)

# Simulation of othe space-time Binary Gaussian random field:

#data <- RFsim(x, y, times, corrmodel="exp_exp", model="BinaryGauss",
#        threshold=0, param=list(nugget=nugget,mean=mean,scale_t=1,
#        scale_s=1,sill=1))$data

# Empirical lorelogram estimation:
#fit <- EVariogram(data, x, y, times, type="lorelogram", maxtime=6,maxdist=4)
```

```
# Building space-time lorelogram
#st.vario <- matrix(fit$variogramst,length(fit$centers),length(fit$bint))
#tempvario <- rbind(fit$variogramt,st.vario)
#st.vario <- cbind(c(NA,fit$variograms),tempvario)

#par(mfrow=c(2,2))

# Results:
# Marginal spatial empirical lorelogram
#plot(fit$centers, fit$variograms, xlab='h', ylab="lorelogram",
#     xlim=c(0, max(fit$centers)), pch=20,main="Marginal spatial lorelgram")

# Marginal temporal empirical  lorelogram
#plot(fit$bint, fit$variogramt, xlab='t', ylab="lorelogram",
#     xlim=c(0,max(fit$bint)), pch=20,main="Marginal temporal lorelogram")

# 3d Spatio-temporal lorelogram:
#require(scatterplot3d)
#st.grid <- expand.grid(c(0,fit$centers), c(0,fit$bint))

#scatterplot3d(st.grid[,1], st.grid[,2], c(st.vario),
#              highlight.3d=TRUE, xlab = "h", ylab = "t",
#              zlab="Lorelogram",pch=20, main="Space-time lorelogram",
#              cex.axis=.7, mar=c(2,2,2,2), mgp=c(0,0,0),cex.lab=.7)

# smoothed version:
#par(mai=c(.2,.2,.2,.2),mgp=c(1.2,.5, 0))
#persp(c(0,fit$centers),c(0,fit$bint),st.vario,
#      xlab = "h", ylab = "t", zlab ="Lorelogram",
#      ltheta=90, shade=0.75, ticktype="detailed",
#      phi=30,theta=30, main="Space-time lorelogram",
#      cex.axis=.8, cex.lab=.8)
```

`FitComposite` *Max-Likelihood-Based Fitting of Gaussian, Binary and Max-Stable Fields*

Description

Maximum weighted composite-likelihood fitting of spatio-temporal Gaussian, binary and spatial max-stable random fields. For the spatio-temporal Gaussian random field, (restricted) maximum likelihood and tapered likelihood fitting is also avalable. The function returns the model parameters' estimates and the estimates' variances and allows to fix any of the parameters.

Usage

```
FitComposite(data, coordx, coordy=NULL, coordt=NULL, corrmodel,
             distance='Eucl', fixed=NULL, grid=FALSE, likelihood='Marginal',
             margins='Gev', maxdist=NULL, maxtime=NULL, model='Gaussian',
             optimizer='Nelder-Mead',replicates=1, start=NULL, taper=NULL,
             tapsep=NULL, threshold=NULL,type='Pairwise', varest=FALSE,
             vartype='SubSamp', weighted=FALSE, winconst, winstp)
```

Arguments

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

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

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

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

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

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

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

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

`likelihood` String; the configuration of the composite likelihood. `Marginal` is the default, see the Section **Details**.

`margins` String; the type of the marginal distribution of the max-stable field. `Gev` is the default, see the Section **Details**.

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

`maxtime` Numeric; an optional positive value indicating the maximum temporal separation considered in the composite or tapered likelihood computation (see **Details**).

`model` String; the type of random field and therefore the densities associated to the likelihood objects. `Gaussian` is the default, see the Section **Details**.

`optimizer` String; the optimization algorithm (see `optim` for details). 'Nelder-Mead' is the default.

`replicates` Numeric; a positive integer denoting the number of independent and identically distributed (iid) replications of a spatial or spatial-temporal random field. Optional argument, the default value is `1` then a single realisation is considered.

`start` An optional named list with the initial values of the parameters that are used by the numerical routines in maximization procedure. `NULL` is the default (see **Details**).

taper	String; the name of the type of covariance matrix. It can be `Standard` (the default value) or `Tapering` for taperd covariance matrix.
tapsep	Numeric; an optional value indicating the separabe parameter in the space time quasi taper (see **Details**).
threshold	Numeric; a value indicating a threshold for the binary random field. Optional in the case that `model` is `BinaryGauss`, see the Section **Details**.
type	String; the type of the likelihood objects. If `Pairwise` (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods (see **Details**).
varest	Logical; if `TRUE` the estimates' variances and standard errors are returned. `FALSE` is the default.
vartype	String; (`SubSamp` the default) the type of method used for computing the estimates' variances, see the Section **Details**.
weighted	Logical; if `TRUE` the likelihood objects are weighted, see the Section **Details**. If `FALSE` (the default) the composite likelihood is not weighted.
winconst	Numeric; a positive value for computing the sub-window size where observations are sampled in the sub-sampling procedure (if `vartype=SubSamp`). For increasing `winconst` increasing sub-window sizes are obtained. Optional argument, the default is 1. See **Details** for more information.
winstp	Numeric; a value in $(0, 1]$ for computing the sub-window step (in the subsampling procedure). This value denote the proportion of the sub-window size. Optional argument, the default is 0.5. See **Details** for more information.

Details

Note, that the standard likelihood may be seen as particular case of the composite likelihood. In this respect `FitComposite` provides maximum (restricted) likelihood fitting. Only composite likelihood estimation based on pairs are considered. Specifically marginal pairwise, conditional pairwise and difference pairwise. Covariance tapering is considered only for Gaussian random fields.

With `data`, `coordx`, `coordy`, `coordt`, `grid` and `replicates` parameters:

- If `data` is a numeric d-dimensional vector, `coordx` and `coordy` are two numeric d-dimensional vectors (or `coordx` is $(d \times 2)$-matrix and `coordy=NULL`), `coordt=NULL`, `grid=FALSE` and `replicates=1`, then the data are interpreted as a single spatial realisation observed on d spatial sites;
- If `data` is a numeric $(n \times d)$-matrix, `coordx` and `coordy` are two numeric d-dimensional vectors (or `coordx` is $(d \times 2)$-matrix and `coordy=NULL`), `coordt=NULL`, `grid=FALSE` and `replicates=n`, then the data are interpreted as `n` iid replications of a spatial random field observed on `d` spatial sites.
- If `data` is a numeric $(d \times d)$-matrix, `coordx` and `coordy` are two numeric d-dimensional vectors, `coordt=NULL`, `grid=TRUE` and `replicates=1`, then the data are interpreted as a single spatial random field realisation observed on `d` equispaced spatial sites (named regular grid).
- If `data` is a numeric $(d \times d \times n)$-array, `coordx` and `coordy` are two numeric d-dimensional vectors, `coordt=NULL`, `grid=TRUE` and `replicates=n`, then the data are interpreted as `n` iid realisations of a spatial random field observed on `d` equispaced spatial sites.
- If `data` is a numeric $(t \times d)$-matrix, `coordx` and `coordy` are two numeric d-dimensional vectors (or `coordx` is $(d \times 2)$-matrix and `coordy=NULL`), `coordt` is a numeric t-dimensional

vector, `grid=FALSE` and `replicates=1`, then the data are interpreted as a single spatial-temporal realisation of a random field observed on `d` spatial sites and for `t` times.

- If `data` is a numeric $(t \times d \times n)$-array, `coordx` and `coordy` are two numeric d-dimensional vectors (or `coordx` is $(d{\times}2)$-matrix and `coordy=NULL`), `coordt` is a numeric t-dimensional vector, `grid=FALSE` and `replicates=n`, then the data are interpreted as `n` iid realisations of a spatial-temporal random field observed on `d` spatial sites and for `t` times.
- If `data` is a numeric $(d \times d \times t)$-array, `coordx` and `coordy` are two numeric d-dimensional vectors, `coordt` is a numeric t-dimensional vector, `grid=TRUE` and `replicates=1`, then the data are interpreted as a single spatial-temporal realisation of a random field observed on `d` equispaced spatial sites and for `t` times.
- If `data` is a numeric $(d \times d \times t \times n)$-array, `coordx` and `coordy` are two numeric d-dimensional vectors, `coordt` is a numeric t-dimensional vector, `grid=TRUE` and `replicates=n`, then the data are interpreted as `n` iid realisation of a spatial-temporal random field observed on `d` equispaced spatial sites and for `t` times.

The `corrmodel` parameter allows to select a specific correlation function for the random field. (See `Covmatrix`).

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

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

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

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

The `margins` parameter concerns only max-stable fields and indicates how the margins are considered. The options are `Gev` or `Frechet`, where in the former case the marginals are supposed generalized extreme value distributed and in the latter case unit Frechet distributed.

The `maxdist` parameter set the maximum spatial distance below which pairs of sites with inferior distances are considered in the composite-likelihood. This can be inferior of the effective maximum spatial distance. **Note** that this corresponds to use a weighted composite-likelihood with binary weights. Pairs with distance less than `maxdist` have weight 1 and are included in the likelihood computation, instead those with greater distance have weight 0 and then excluded. The default is `NULL`, in this case the effective maximum spatial distance between sites is considered.

The same arguments of `maxdist` are valid for `maxtime` but here the weigthed composite-likelihood regards the case of spatial-temporal field. At the moment is implemented only for Gaussian random fields. The default is `NULL`, in this case the effective maximum temporal lag between pairs of observations is considered.

In the case of tapering likelihood `maxdist` and `maxtime` describes the spatial and temporal compact support of the taper model (see `Covmatrix`). If they are not specified then the maximum spatial and temporal distances are considered. In the case of space time quasi taper the `tapsep` parameter allows to specify the spatio temporal compact support (see `Covmatrix`).

The `model` paramter indicates the type of random field considered, for instance `model=Gaussian` denotes a Gaussian random field. Accordingly, this also determines the analytical expression of the finite dimensional distribution associated with the random field. The available options are:

- `Gaussian`, for a Gaussian random field (see i.e. Wackernagel, H. 1998);
- `BinaryGauss`, for a Binary random field (see Heagerty and Lele 1998)
- `BrowResn`, for a Brown-Resnick max-stable random field (see Kabluchko, Z. et al. 2009);
- `ExtGauss`, for an Extremal Gaussian max-stable random field (known also as Schlather model) (see Schlather, M. 2002);
- `ExtT`, for an Extremal `t` max-stable random field (see Davison, A. C. et al. 2012);

Note, that only for the `Gaussian` case the estimation procedure is implemented for spatial and spatial-temporal random fields.

The `start` parameter allows to specify starting values. If `start` is omitted the routine is computing the starting values using the weighted moment estimator.

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

The `threshold` parameter indicates the value (common for all the spatial sites) above which the values of the underlying Gaussian latent process are considered sucesses events (values below are instead failures). See e.g. Heagerty and Lele (1998) for more details.

The `type` parameter represents the type of likelihood used in the composite-likelihood definition. The possible alternatives are listed in the following scheme.

1. If a Gaussian random field is considered (`model=Gaussian`):
    - If the composite is formed by marginal likelihoods (`likelihood=Marginal`):
        - `Pairwise`, the composite-likelihood is defined by the pairwise likelihoods;
        - `Difference`, the composite-likelihood is defined by likelihoods which are obtained as difference of the pairwise likelihoods.
    - If the composite is formed by conditional likelihoods (`likelihood=Conditional`)
        - `Pairwise`, the composite-likelihood is defined by the pairwise conditional likelihoods.
    - If the composite is formed by a full likelihood (`likelihood=Full`):
        - `Standard`, the objective function is the classical multivariate likelihood;
        - `Restricted`, the objective function is the restricted version of the full likelihood (e.g. Harville 1977, see **References**);
        - `Tapering`, the objective function is the tapered version of the full likelihood (e.g. Kaufman et al. 2008, see **References**).

The `varest` parameter specifies if the standard error estimation of the estimated parameters must be computed. For Gaussian random field and standard (restricted) likelihood estimation, standard errrors are computed as square root of the diagonal elements of the Fisher Information matrix (asymptotic covariance matrix of the estimates under increasing domain). For Gaussian random field and tapered and composite likelihood estimation, standard errors estimate are computed as square root of the diagonal elements of the Godambe Information matrix. (asymptotic covariance matrix of the estimates under increasing domain (see Shaby, B. and D. Ruppert (2012) for tapering and Bevilacqua et. al. (2012) , Bevilacqua and Gaetan (2013) for weighted composite likelihood)). The `vartype` parameter specifies the method used to compute the estimates' variances in the composite likelihood case. In particular for estimating the variability matrix `J` in the Godambe expression matrix. This parameter is considered if `varest=TRUE`. The options are:

- `SubSamp` (the default), indicates the Sub-Sampling method;
- `Sampling`, indicates that the variability matrix is estimated by the sample contro-part (available only for n iid replications of the random field, i.e. `replicates=n`);

The `weighted` parameter specifies if the likelihoods forming the composite-likelihood must be weighted. If `TRUE` the weights are selected by opportune procedures that improve the efficient of the maximum composite-likelihood estimator (not implemented yet). If `FALSE` the efficient improvement procedure is not used.

For computing the standard errors by the sub-sampling procedure, `winconst` and `winstp` parameters represent respectively a positive constant used to determine the sub-window size and the the step with which the sub-window moves.

In the spatial case (subset of R^2), the domain is seen as a rectangle $B \times H$, therefore the size of the sub-window side b is given by $b = winconst \times \sqrt(B)$ (similar is of h). For a complete description see Lee and Lahiri (2002). By default `winconst` is set $B/(2 \times \sqrt(B))$. The `winstp` parameter is used to determine the sub-window step. The latter is given by the proportion of the sub-window size, so that when `winstp=1` there is not overlapping between contiguous sub-windows. In the spatial case by default `winstp=0.5`. The sub-window is moved by successive steps in order to cover the entire spatial domain. Observations, that fall in disjoint or overlapping windows are considered indipendent samples.

In the spatio-temporal case the subsampling is meant only in time as described by Li et al. (2007). Thus, `winconst` represents the lenght of the temporal sub-window. By default the size of the sub-window is computed following the rule established in Li et al. (2007). By default `winstp` is the time step.

Observe that in the spatio-temporal case, the returned values by `srange` and `trange`, represent respectively the minimum and maximum of the marginal spatial distances and those of the temporal separations. Thus, the minimum being not the overall (i.e. considering the spatio-temporal coordinates) is not zero, as one could be expect and the latter can be easily added by the user.

Value

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

`clic`	The composite information criterion, if the full likelihood is considered then it coincides with the Akaike information criterion;
`coordx`	A d-dimensional vector of spatial coordinates;
`coordy`	A d-dimensional vector of spatial coordinates;
`coordt`	A t-dimensional vector of temporal coordinates;
`convergence`	A string that denotes if convergence is reached;
`corrmodel`	The correlation model;
`data`	The vector or matrix or array of data;
`distance`	The type of spatial distance;
`fixed`	The vector of fixed parameters;
`iterations`	The number of iteration used by the numerical routine;
`likelihood`	The configuration of the composite likelihood;
`logCompLik`	The value of the log composite-likelihood at the maximum;
`message`	Extra message passed from the numerical routines;
`model`	The density associated to the likelihood objects;
`nozero`	In the case of tapered likelihood the percentage of non zero values in the covariance matrix. Otherwise is NULL.
`numcoord`	The number of spatial coordinates;

numrep	The number of the iid replicatations of the random field;
numtime	The number the temporal realisations of the random field;
param	The vector of parameters' estimates;
srange	The minimum and maximum spatial distance (see **Details**). The maximum is maxdist, if inserted, rather the effective maximum distance;
stderr	The vector of standard errors;
sensmat	The sensitivity matrix;
varcov	The matrix of the variance-covariance of the estimates;
varimat	The variability matrix;
vartype	The method used to compute the variance of the estimates;
trange	The minimum and maximum temporal separation (see **Details**). The maximum is maxtime, if inserted, rather then the effective maximum separation;
threshold	The threshold used in the binary random field.
type	The type of the likelihood objects.
winconst	The constant use to compute the window size in the sub-sampling procedure;
winstp	The step used for moving the window in the sub-sampling procedure

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

References

Maximum Restricted Likelihood Estimator:

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

Tapered likelihood:

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

Shaby, B. and D. Ruppert (2012). Tapered covariance: Bayesian estimation and asymptotics. *J. Comp. Graph. Stat.*, **21-2**, 433–452.

Composite-likelihood:

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

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

Weighted Composite-likelihood for binary random fields:

Patrick, J. H. and Subhash, R. L. (1998) A Composite Likelihood Approach to Binary Spatial Data. *Journal of the American Statistical Association, Theory & Methods*, **93**, 1099–1111.

Weighted Composite-likelihood for max-stable random fields:

Davison, A. C. and Gholamrezaee, M. M. (2012) Geostatistics of extremes. *Proceedings of the Royal Society of London, series A*, **468**, 581–608.

Padoan, S. A. (2008). *Computational Methods for Complex Problems in Extreme Value Theory*. PhD. Thesis, Department of Statistics, University of Padua.

Padoan, S. A. Ribatet, M. and Sisson, S. A. (2010) Likelihood-Based Inference for Max-Stable Processes. *Journal of the American Statistical Association, Theory & Methods*, **105**, 263–277.

Weighted Composite-likelihood for Gaussian random fields:

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

Bevilacqua, M. Gaetan, C. (2013) On composite likelihood inference based on pairs for spatial Gaussian random fields *Techical Report, Department of Statistics, de Valparaiso University*.

Spatial Extremes:

Davison, A. C., Padoan, S. A., and Ribatet, M. (2012) Statistical Modelling of Spatial Extremes, with discussion. *Statistical Science*, **27**, 161–186.

de Haan, L., and Pereira, T. T. (2006) Spatial Extremes: Models for the Stationary Case. *The Annals of Statistics*, **34**, 146–168.

Kabluchko, Z. (2010) Extremes of Independent Gaussian Processes. *Extremes*, **14**, 285–310.

Kabluchko, Z., Schlather, M., and de Haan, L. (2009) Stationary max-stable fields associated to negative definite functions. *The Annals of Probability*, **37**, 2042–2065.

Schlather, M. (2002) Models for Stationary Max-Stable Random Fields. *Extremes*, **5**, 33–44.

Smith, R. L. (1990) Max-Stable Processes and Spatial Extremes. *Unpublished manuscript*, University of North California.

Sub-sampling estimation:

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

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

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

Li, B., Genton, M. G. and Sherman, M. (2007). A nonparametric assessment of properties of space-time covariance functions. *Journal of the American Statistical Association*, **102**, 736–744

See Also

`Covmatrix`, `GaussRF`, `MaxStableRF`, `WLeastSquare`, `optim`

Examples

```
library(CompRandFld)
library(RandomFields)
library(spam)
set.seed(3132)

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

# Define the spatial-coordinates of the points:
x <- runif(100, 0, 10)
```

```
y <- runif(100, 0, 10)

# Set the covariance model's parameters:
corrmodel <- "exponential"
mean <- 0
sill <- 1
nugget <- 0
scale <- 1.5
param<-list(mean=mean,sill=sill,nugget=nugget,scale=scale)

# Simulation of the spatial Gaussian random field:
data <- RFsim(x, y, corrmodel=corrmodel, param=param)$data

# Fixed parameters
fixed<-list(mean=mean,nugget=nugget)

# Starting value for the estimated parameters
start<-list(scale=scale,sill=sill)


################################################################
###
### Example 1. Maximum likelihood fitting of
### Gaussian random fields with exponential correlation.
### One spatial replication.
### Likelihood setting: composite with
### marginal pairwise likelihood objects.
###
###############################################################


# Maximum composite-likelihood fitting of the random field:
fit <- FitComposite(data, x, y, corrmodel=corrmodel, maxdist=5,
                    varest=TRUE,start=start,fixed=fixed)

# Results:
print(fit)

################################################################
###
### Example 2. Maximum likelihood fitting of
### Gaussian random fields with exponential correlation.
### One spatial replication.
### Likelihood setting: standard full likelihood.
###
###############################################################

# Maximum composite-likelihood fitting of the random field:
fit <- FitComposite(data, x, y, corrmodel=corrmodel,likelihood="Full",
                    type="Standard",varest=TRUE,start=start,fixed=fixed)
# Results:
print(fit)

################################################################
###
### Example 3. Maximum likelihood fitting of
### Gaussian random fields with exponetial correlation.
```

```
### One spatial replication.
### Likelihood setting: tapered full likelihood.
###
###############################################################

# Maximum tapered likelihood fitting of the random field:
fit <- FitComposite(data, x, y, corrmodel=corrmodel,likelihood="Full",
                    type="Tapering",taper="Wendland1",maxdist=4,
                    varest=TRUE,start=start,fixed=fixed)

# Results:
print(fit)


################################################################
###
### Example 4. Maximum composite-likelihood fitting of
### max-stable random fields. Extremal Gaussian model with
### exponential correlation. n iid spatial replications.
### Likelihood setting: composite with marginal pairwise
### likelihood objects.
###
###############################################################

# Simulation of a max-stable random field in the specified points:
data <- RFsim(x, y, corrmodel=corrmodel, model="ExtGauss", replicates=30,
              param=list(mean=mean,sill=sill,nugget=nugget,scale=scale))$data

# Maximum composite-likelihood fitting of the random field:
fit <- FitComposite(data, x, y, corrmodel=corrmodel, model="ExtGauss",
                    replicates=30, varest=TRUE, vartype="Sampling",
                    margins="Frechet",start=list(sill=sill,scale=scale))

# Results:
print(fit)

################################################################
###
### Example 5. Maximum likelihood fitting of
### Binary-Gaussian random fields with exponential correlation.
### One spatial replication.
### Likelihood setting: composite with marginal pairwise
### likelihood objects.
###
###############################################################

set.seed(3128)

x <- runif(200, 0, 10)
y <- runif(200, 0, 10)

# Simulation of the spatial Binary-Gaussian random field:
data <- RFsim(x, y, corrmodel=corrmodel, model="BinaryGauss",
              threshold=0, param=list(mean=mean,sill=.8,
              nugget=nugget,scale=scale))$data

# Maximum composite-likelihood fitting of the random field:
```

```
fit <- FitComposite(data, x, y, corrmodel=corrmodel, threshold=0,
                    model="BinaryGauss", fixed=list(nugget=nugget,
                    mean=0),start=list(scale=.1,sill=.1))

# Results:
print(fit)


################################################################
######### Examples of spatio-temporal random fields ###########
################################################################

# Define the temporal sequence:
time <- seq(1, 80, 1)

# Define the spatial-coordinates of the points:
x <- runif(10, 0, 10)
y <- runif(10, 0, 10)
coords=cbind(x,y)

# Set the covariance model's parameters:
corrmodel="exp_exp"
scale_s=0.5
scale_t=1
sill=1
nugget=0
mean=0

param<-list(mean=0,scale_s=1,scale_t=1,sill=sill,nugget=nugget)

# Simulation of the spatial-temporal Gaussian random field:
data <- RFsim(coordx=coords,coordt=time,corrmodel=corrmodel,
              param=param)$data

# Fixed parameters
fixed<-list(mean=mean,nugget=nugget)

# Starting value for the estimated parameters
start<-list(scale_s=scale_s,scale_t=scale_t,sill=sill)

#################################################################
###
### Example 6. Maximum likelihood fitting of
### Gaussian random field with double-exponential correlation.
### One spatio-temporal replication.
### Likelihood setting: composite with conditional pairwise
### likelihood objects.
###
################################################################

# Maximum composite-likelihood fitting of the random field:
fit <- FitComposite(data=data,coordx=coords,coordt=time,corrmodel="exp_exp",
                    maxtime=2,maxdist=1,likelihood="Conditional",type="Pairwise",
                    start=start,fixed=fixed)

# Results:
print(fit)
```

FitGev *Max-likelihood Fitting of the GEV Distribution*

Description

the function returns the parameters' estimates and the variances of the estimates (if required) of the generalized extreme value distribution for a given dataset of extreme values.

Usage

```
FitGev(data, method='Nelder-Mead', start, varest=FALSE)
```

Arguments

`data`	A vector of extreme values.
`method`	The optimization method (see `optim` for details). 'Nelder-Mead' is the default.
`start`	A named list with the initial values for the parameters over which the likelihood is to be maximized.
`varest`	Logical; if `TRUE` the estimate' variances and the standard errors are returned, instead if `FALSE` (the default) only the estimate are computed.

Details

If `start` is omitted the routine is computing the starting values using moment estimators.

Value

The returned object is a list with:

`param`	The vector of parameters' estimates.
`varcov`	The matrix of the variance-covariance of the estimates.
`stderr`	The vector of the standard errors.

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, `http://faculty.unibocconi.it/simonepadoan`; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, `http://staff.deuv.cl/bevilacqua/`.

References

de Haan, L. and Ferreira, A. (2006) *Extreme Value Theory An Introduction*. Springer Verlang, New York.

Smith, R. L. (1985) Maximum likelihood estimation in a class of non-regular cases. *Biometrika*, **72**, 67–90.

See Also

`GevLogLik`, `optim`

GevLogLik *Log-Likelihood of the GEV Distribution*

Description

The function returns the log-likelihood value of the Generalized Extreme Value Distribution for a given set of data and parameters.

Usage

```
GevLogLik(data, numdata, param)
```

Arguments

`data`	A vector of extreme values.
`numdata`	The number of data observations.
`param`	The vector of GEV parameters (location, scale and shape).

Value

The log-likelihood value is returned.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

References

de Haan, L. and Ferreira, A. (2006) *Extreme Value Theory An Introduction*. Springer Verlang, New York.

HypoTest *Statistical Hypothesis Tests for Nested Models*

Description

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

Usage

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

Arguments

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

Details

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

1. Wald-type (`Wald`);
2. Score-type, also known as Rao-type (`Rao`);
3. Wilks-type; also known as the composite likelihood ratio statistic. Available are variants of the basic version, in particular:
   - Rotnitzky and Jewell adjustment (`WilksRJ`);
   - Satterhwaite adjustment (`WilksS`);
   - Chandler and Bate adjustment (`WilksCB`);
   - Pace, Salvan and Sartori adjustment (`WilksPSS`);

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

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

where $G_{\psi\psi}$ is the $q' \times q'$ submatrix of the Godambe information pertaining to ψ and $\hat{\theta}$ is the maximum likelihood estimator from the full model. The score-type statistic (Rao-type) is

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

where $H^{\psi\psi}$ is the $q' \times q'$ submatrix of the inverse of $H(\theta)$ pertaining to ψ (the same for G) and $\hat{\theta}_{\psi}$ is the constrained maximum likelihood estimate of θ for fixed ψ. These two statistics can be called from the routine `HypoTest` assigning at the argument `statistic` respectively the values: `Wald` and `Rao`.

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

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

The asymptotic distribution of the composite likelihood ratio statistic is given by

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

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

Value

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

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

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, http://staff.deuv.cl/bevilacqua/.

References

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

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

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

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

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

See Also

`FitComposite`.

Examples

```
library(CompRandFld)
library(RandomFields)
set.seed(3451)

# Define the spatial-coordinates of the points:
x <- runif(300, 0, 10)
```

```
y <- runif(300, 0, 10)

################################################################
###
### Example 1. Composite likelihood-based hypothesis testing.
### Simulation of a Gaussian spatial random field with
### stable correlation.
### Estimation by composite likelihood using the setting:
### marginal pairwise likelihood objects.
###
###############################################################

# Set the model's parameters:
corrmodel <- "stable"
mean <- 0
sill <- 1
nugget <- 1
scale <- 1
power <- 1.3

# Simulation of the spatial Gaussian random field:
data <- RFsim(x, y, corrmodel=corrmodel, param=list(mean=mean,
              sill=sill,nugget=nugget,scale=scale,power=power))$data

# Maximum composite-likelihood fitting of the random field, full model:
fit1 <- FitComposite(data, x, y, corrmodel=corrmodel, maxdist=5,
                    varest=TRUE,fixed=list(nugget=1))

# Maximum composite-likelihood fitting of the random field, first nasted model:
fit2 <- FitComposite(data, x, y, corrmodel=corrmodel, maxdist=5,
                    varest=TRUE,fixed=list(nugget=1,sill=1))

# Maximum composite-likelihood fitting of the random field, second nasted model:
fit3 <- FitComposite(data, x, y, corrmodel=corrmodel, maxdist=5,
                    varest=TRUE,fixed=list(nugget=1,sill=1,mean=0,power=1.3))

# Hypothesis testing results:
# composite Wald-type statistic:
HypoTest(fit1, fit2, fit3, statistic='Wald')

# composite score-type statistic:
HypoTest(fit1, fit2, fit3, statistic='Rao')

# composite likelihood ratio statistic with RJ adjustment:
HypoTest(fit1, fit2, fit3, statistic='WilksRJ')

# composite likelihood ratio statistic with S adjustment:
HypoTest(fit1, fit2, fit3, statistic='WilksS')

# composite likelihood ratio statistic with CB adjustment:
HypoTest(fit1, fit2, fit3, statistic='WilksCB')

# composite likelihood ratio statistic with PSS adjustment:
HypoTest(fit1, fit2, fit3, statistic='WilksPSS')
```

InitParam *Initializes the Parameters for Estimation Procedures*

Description

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

Usage

```
InitParam(coordx, coordy, coordt, corrmodel, data, distance, fcall, fixed,
          grid,likelihood, margins, maxdist, maxtime, model, numblock,
          param, parscale, paramrange, replicates, start, taper, tapsep,
          threshold, type, typereal, varest, vartype, weighted,
          winconst,winstp)
```

Arguments

`coordx`	A numeric ($d{\times}2$)-matrix (where `d` is the number of points) assigning 2-dimensions of coordinates or a numeric vector assigning 1-dimension of coordinates.
`coordy`	A numeric vector assigning 1-dimension of coordinates; `coordy` is interpreted only if `coordx` is a numeric vector otherwise it will be ignored.
`coordt`	A numeric vector assigning 1-dimension of temporal coordinates.
`corrmodel`	String; the name of a correlation model.
`data`	A numeric vector or a $(n \times d)$-matrix or $(d \times d \times n)$-matrix of observations.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details**.
`fcall`	String; "fitting" to call the fitting procedure and "simulation" to call the simulation procedure.
`fixed`	A named list giving the values of the parameters that will be considered as known values.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as a vector or a $(n \times d)$-matrix, instead if `TRUE` then $(d \times d \times n)$-matrix is considered.
`likelihood`	String; the configuration of the composite likelihood.
`margins`	String; the type of the marginal distribution of the max-stable field.
`maxdist`	Numeric; an optional positive value indicating the maximum spatial distance considered in the composite-likelihood computation.
`maxtime`	Numeric; an optional positive value indicating the maximum temporal lag considered in the composite-likelihood computation.
`model`	String; the density associated to the likelihood objects. `Gaussian` is the default.
`numblock`	Numeric; the observation size of the underlying random field. Only in case of max-stable random fields and in the simulation.
`param`	A numeric vector of parameter values required in the simulation procedure of random fields.

`parscale`	A numeric vector of scaling factor to improve the maximizing procedure, see `optim`.
`paramrange`	A numeric vector of parameters ranges, see `optim`.
`replicates`	Logical; if `FALSE` (the default) one spatial random field is considered, instead if `TRUE` the data are considered as iid replicates of a field.
`start`	A named list with the initial values of the parameters that are used by the numerical routines in maximization procedure.
`taper`	String; the name of the type of covariance matrix. It can be `Standard` (the default value) or `Tapering` for taperd covariance matrix.
`tapsep`	Numeric; an optional value indicating the separabe parameter in the space time quasi taper (see **Details**).
`threshold`	Numeric; a value indicating a threshold for the binary random field.
`type`	String; the type of likelihood objects. Temporary value set to be "WLeast-Square" (weigthed least-square) in order to compute the starting values.
`typereal`	String; the real type of likelihood objects. See `FitComposite`.
`varest`	Logical; if `TRUE` the estimates' variances and standard errors are returned. `FALSE` is the default.
`vartype`	String; the type of estimation method for computing the estimate variances, see the Section **Details**.
`weighted`	Logical; if `TRUE` the likelihood objects are weighted, see `FitComposite`.
`winconst`	Numeric; a positive real value indicating the sub-window in the sub-sampling procedure. See `FitComposite`.
`winstp`	Numeric; a value in $(0, 1]$ for defining the window step. See `FitComposite`.

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, http://staff.deuv.cl/bevilacqua/.

See Also

`FitComposite`

`Kri` *Spatial and spatio temporal simple and ordinary (tapered) kriging*

Description

The function computes simple or ordinary (tapered) kriging, in addition, for a set of unknown spatial location sites and temporal instants and a given space or space-time covariance model, it computes the Kriging variance.

Usage

```
Kri(data, coordx, coordy=NULL, coordt=NULL, corrmodel,distance="Eucl",
    grid=FALSE, loc, maxdist=NULL, maxtime=NULL, param, taper=NULL,
    tapsep=NULL, time=NULL, type="Standard",type_krig="Simple")
```

Arguments

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

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

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

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

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

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

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

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

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

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

`param` A list of parameter values required for the correlation model.See the Section **Details**.

`taper` String; the name of the taper correlation function, see the Section **Details**.

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

`time` A numeric $(m \times 1)$ vector (where `m` is the number of temporal instants) giving the temporal instants to be predicted; the default is `NULL` then only spatial prediction is performed.

`type` String; if `standard` then standard kriging is performed;if `Tapering` then kriging with covariance tapering is performed.

`type_krig` String; the type of kriging. If `Simple` (the default) then simple kriging is performed. If `ordinary` then ordinary kriging is performed. (See the Section **Details**).

Details

For a spatial or spatio-temporal dataset, given a set of locations and temporal istants and a correlation model `corrmodel` with some fixed parameters, the function computes simple or ordinary kriging, for the specified spatial locations `loc` and temporal instants `time`, providing also the respective standard error. For the choice of the spatial or spatio temporal correlation model see details in Covmatrix function. The parameter param specifies the covariance parameters, see CorrelationParam and Covmatrix 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 within list param (See examples). In addition, it is possible to perform kriging based on covariance tapering for simple kriging (Furrer et. al, 2008). In this case, space or space-time tapered function and spatial or spatio- temporal compact support must be specified. For the choice of a space or space-time tapered function see Covmatrix. When performing kriging with covariance tapering, sparse matrix algorithms are exploited using the package spam.

Value

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

coordx	A d-dimensional vector of spatial coordinates used for prediction;
coordy	A d-dimensional vector of spatial coordinates used for prediction;
coordt	A t-dimensional vector of temporal coordinates used for prediction;
corrmodel	String: the correlation model;
covmatrix	The covariance matrix if type is Standard. An object of class spam if type is Tapering
data	The vector or matrix or array of data used for prediction
distance	String: the type of spatial distance;
grid	TRUE if the spatial data used for prediction are observed in a regular grid, otherwise FALSE;
loc	A $(n \times 2)$-matrix of spatial locations to be predicted.
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;
param	Numeric: The covariance parameters;
pred	A $(m \times n)$-matrix of spatio or spatio temporal kriging prediction;
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 Ordinary).
varpred	A $(m \times n)$-matrix of spatio or spatio temporal variance kriging prediction;

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

References

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

See Also

Covmatrix

Examples

```
library(CompRandFld)
library(fields)

################################################################
############### Example of Spatial kriging  ####################
################################################################

# Define the spatial-coordinates of the points:
x <- runif(50, 0, 1)
y <- runif(50, 0, 1)

# Set the model's parameters:
corrmodel <- "exponential"
mean<-0
sill<-1
nugget<-0
scale<-0.5
param<-list(mean=mean,sill=sill,nugget=nugget,scale=scale)

# spatial matrix location sites
coords<-cbind(x,y)

# Simulation of the spatial Gaussian random field:
set.seed(3132)
data <- RFsim(coordx=coords, corrmodel=corrmodel,
              param=param)$data

# Maximum likelihood fitting :
fit <- FitComposite(data, coordx=coords, corrmodel=corrmodel,
                    likelihood='Full', type='Standard',
                    fixed=list(mean=mean,nugget=nugget))

# locations to predict
xx<-seq(0,1,0.02)
loc_to_pred<-as.matrix(expand.grid(xx,xx))

################################################################
###
### Example 1. Spatial simple kriging of n sites of a
### Gaussian random fields with exponential correlation.
###
###############################################################

pr<-Kri(loc=loc_to_pred,coordx=coords,corrmodel=corrmodel,
```

```
       param= as.list(c(fit$param,fit$fixed)), data=data)

################################################################
###
### Example 2. Spatial tapered simple kriging of n sites of a
### Gaussian random fields with exponential correlation.
###
###############################################################

pr_tap=Kri(loc=loc_to_pred,coordx=coords,corrmodel=corrmodel,data=data,
       param= as.list(c(fit$param,fit$fixed)),type="Tapering",
       maxdist=0.15,taper="Wendland1")

colour <- rainbow(100)

par(mfrow=c(2,2))
# simple kriging map prediction
image.plot(xx, xx, matrix(pr$pred,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Simple Kriging")

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

# simple tapered kriging map prediction
image.plot(xx, xx, matrix(pr_tap$pred,ncol=length(xx)),col=colour,
           xlab="",ylab="",main="Simple Tapered Kriging")

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


################################################################
########### Examples of Spatio-temporal kriging  ###############
################################################################

# Define the spatial-coordinates of the points:
x <- runif(15, 0, 1)
y <- runif(15, 0, 1)
coords<-cbind(x,y)
times<-1:7

#  Define the times to predict
times_to_pred<-8:10

# Define model correlation and associated parameters
corrmodel<-"exp_exp"
param<-list(nugget=0,mean=1,scale_s=1,scale_t=2,sill=2)

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

# Maximum likelihood fitting of the space time random field:
fit <- FitComposite(data, coordx=coords, coordt=times,
```

```
                    corrmodel=corrmodel, likelihood='Full',
                    type='Standard',fixed=list(mean=1,
                    nugget=0))

################################################################
###
### Example 3. Temporal ordinary kriging of one location site for
### a Gaussian random fields with estimated double exponential
### correlation.
###
################################################################

param<-as.list(c(fit$param,fit$fixed))
pr<-Kri(loc=c(.5,.5),time=times_to_pred,coordx=coords,coordt=times,
        corrmodel=corrmodel,param=param,data=data,type_krig="ordinary")

pr$pred       #  prediction
pr$varpred    #  prediction variance


################################################################
###
### Example 4. Spatio temporal simple kriging of n locations
### sites and m temporal instants for a Gaussian random fields
### with estimated double exponential correlation.
###
###############################################################

pr<-Kri(loc=loc_to_pred,time=times_to_pred,coordx=coords,coordt=times,
       corrmodel=corrmodel, param=param,data=data)

par(mfrow=c(3,2))

for(i in 1:3){
image.plot(xx, xx, matrix(pr$pred[i,],ncol=length(xx)),col=colour,
           main = paste("Kriging Time=" , i),ylab="")
image.plot(xx, xx, matrix(pr$varpred[i,],ncol=length(xx)),col=colour,
           main = paste("Std error Time=" , i),ylab="")
}



################################################################
###
### Example 5. Spatio temporal tapered simple kriging of n locations
### sites and m temporal instants for a Gaussian random fields
### with estimated double exponential correlation.
###
###############################################################

#pr_tap<-Kri(loc=loc_to_pred,time=times_to_pred,coordx=coords,coordt=times,
#       corrmodel=corrmodel, param=param,type="Tapering",maxdist=0.4,maxtime=4,
#       taper="Wendland2_Wendland2",data=data)

#par(mfrow=c(3,2))

#for(i in 1:3){
```

```
#image.plot(xx, xx, matrix(pr_tap$pred[i,],ncol=length(xx)),col=colour,
#           main = paste("Tapered Kriging Time=" , i),ylab="")
#image.plot(xx, xx, matrix(pr_tap$varpred[i,],ncol=length(xx)),col=colour,
#           main = paste("Tapered Std error Time=" , i),ylab="")
#}
```

`Likelihood` *Optimizes the Log Likelihood*

Description

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

Usage

```
Likelihood(corrmodel, data, fixed, flagcor, flagnuis, grid, lower,
           model, namescorr, namesnuis, namesparam, numcoord, numpairs,
           numparamcor, numrep, numtime, optimizer, param, setup,
           spacetime, varest, taper, type, upper)
```

Arguments

`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`	A numeric vector with the lower bounds of the parameters' ranges.
`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.
`numrep`	Numeric; the number of iid replications.
`numtime`	Numeric; the number of temporal observations.
`optimizer`	String; the optimization algorithm (see `optim` for details). 'Nelder-Mead' is the default.
`param`	A numeric vector of parameters.

`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`	A numeric vector with the upper bounds of the parameters' ranges.

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, `http://faculty.unibocconi.it/simonepadoan`; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, `http://staff.deuv.cl/bevilacqua/`.

See Also

`FitComposite`

`MomEst` *Estimation of the GEV parameters by the Method of Moments*

Description

Using the moment estimator, the function returns the parameter estimates of the generalized extreme value distribution for a given dataset of extreme values.

Usage

```
MomEst(data, n)
```

Arguments

`data`	A vector of extreme values.
`n`	The number of observations.

Value

The returned object is a list with:

`location`	The location estimate.
`scale`	The scale estimate.
`shape`	The shape estimate.

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, `http://faculty.unibocconi.it/simonepadoan`; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, `http://staff.deuv.cl/bevilacqua/`.

References

de Haan, L. and Ferreira, A. (2006) *Extreme Value Theory An Introduction*. Springer Verlang, New York.

See Also

GevLogLik, FitGev

NuisanceParam — *Lists the Nuisance Parameters of a Random Field*

Description

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

Usage

```
NuisanceParam(model)
```

Arguments

model	String; the name of a random field.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

See Also

FitComposite

RFsim — *Simulation of Gaussian, Binary and Max-stable Random Fields*

Description

Simulation of spatial and spatio-temporal Gaussian, binary and max-stable random fields. The function returns one or more replications of a random field for a given covariance model and covariance parameters.

Usage

```
RFsim(coordx, coordy=NULL, coordt=NULL, corrmodel, distance="Eucl",
      grid=FALSE, model='Gaussian', numblock=NULL, param,
      replicates=1, threshold=NULL)
```

Arguments

coordx	A numeric $(d \times 2)$-matrix (where d is the number of spatial sites) giving 2-dimensions of spatial coordinates or a numeric d-dimensional vector giving 1-dimension of spatial coordinates.
coordy	A numeric vector giving 1-dimension of spatial coordinates; coordy is interpreted only if coordx is a numeric vector or grid=TRUE otherwise it will be ignored. Optional argument, the default is NULL then coordx is expected to be numeric a $(d \times 2)$-matrix.
coordt	A numeric vector giving 1-dimension of temporal coordinates. At the moment implemented only for the Gaussian case. Optional argument, the default is NULL then a spatial random field is expected.
corrmodel	String; the name of a correlation model, for the description see the Section **Details**.
distance	String; the name of the spatial distance. The default is Eucl, the euclidean distance. See the Section **Details** of FitComposite.
grid	Logical; if FALSE (the default) the data are interpreted as spatial or spatial-temporal realisations on a set of non-equispaced spatial sites (irregular grid).
model	String; the type of random field and therefore the densities associated to the likelihood objects. Gaussian is the default, see the Section **Details**.
numblock	Numeric; the observation size of the underlying random field. Only in case of max-stable random fields.
param	A list of parameter values required in the simulation procedure of random fields, see **Examples**.
replicates	Numeric; a positive integer denoting the number of independent and identically distributed (iid) replications of a spatial or spatial-temporal random field. Optional argument, the default value is 1 then a single realisation is considered.
threshold	Numeric; a value indicating a threshold for the binary random field. Optional in the case that model is BinaryGauss, see the Section **Details**.

Details

Note that this function is also interfaced to the **R** package **RandomFields**, using fast routines therein developed for the simulation of random fields, see for example GaussRF, MaxStableRF, ect.

Value

Returns an object of class RFsim. An object of class RFsim is a list containing at most 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;
corrmodel	The correlation model; see Covmatrix.
data	The vector or matrix or array of data, see FitComposite;
distance	The type of spatial distance;
model	The type of random field, see FitComposite.
numcoord	The number of spatial coordinates;

`numtime`	The number the temporal realisations of the random field;
`param`	The vector of parameters' estimates;
`randseed`	The seed used for the random simulation;
`replicates`	The number of the iid replicatations of the random field;
`spacetime`	`TRUE` if spatio-temporal and `FALSE` if spatial random field;
`threshold`	The threshold for deriving the binary random field.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

See Also

Covmatrix, GaussRF, MaxStableRF

Examples

```
library(CompRandFld)
library(RandomFields)
library(mapproj)
library(fields)

################################################################
###
### Example 1. Simulation of a Gaussian random field.
### Gaussian random fields with Whittle-Matern correlation.
### One spatial replication.
###
###
###############################################################

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

set.seed(261)
# Simulation of a spatial Gaussian random field:
data <- RFsim(x, y, corrmodel="matern", param=list(smooth=0.5,
             mean=0,sill=1,scale=0.2,nugget=0))$data

################################################################
###
### Example 2. Simulation of a binary random field based on
### the latent Gaussian random field with exponential correlation.
### One spatial replication on a regular grid
###
###
###############################################################

# Define the spatial-coordinates of the points:
x <- seq(0, 1, 0.02)
y <- seq(0, 1, 0.02)
```

```
set.seed(251)

# Simulation of a spatial binary random field:
sim <- RFsim(x, y, corrmodel="exponential", grid=TRUE,
             model="BinaryGauss", threshold=0,
             param=list(nugget=0,mean=0,scale=.1,sill=1))

image.plot(x,y,sim$data,col=terrain.colors(100))

################################################################
###
### Example 3. Simulation of a max-stable random
### extremal-t type with exponential correlation.
### One spatial replication on a regular grid
###
###
###############################################################

set.seed(341)

# Simulation of a spatial binary random field:
sim <- RFsim(x, y, corrmodel="exponential", grid=TRUE, model="ExtT",
             numblock=500, param=list(nugget=0,mean=0,scale=.1,
             sill=1,df=5))

image.plot(x,y,log(sim$data))


################################################################
###
### Example 4. Simulation of a Gaussian random field.
### with double exponential correlation.
### One spatio-temporal replication.
###
###
###############################################################

# Define the spatial-coordinates of the points:
#x <- seq(0, 1, 0.1)
#y <- seq(0, 1, 0.1)
# Define the temporal-coordinates:
#times <- seq(1, 10, 1)
#
# Simulation of a spatial Gaussian random field:
#sim <- RFsim(x, y, times, corrmodel="exp_exp", grid=TRUE,
#             param=list(nugget=0,mean=0,scale_s=0.3,
#             scale_t=0.5,sill=1))

# For a simple animation movie
# library(animation)
# saveMovie(for (i in 1:length(times)) image(x,y,sim$data[,,i]),
#           height=300,width=500,interval=1,
#           outdir = "c:/",movie.name = "spacetime_animation.mpg")
```

```
################################################################
###
### Example 5. Simulation of a Gaussian random field
### with  exponential correlation on a portion of  the earth surface
### One spatial replication.
###
###
###############################################################

#lon_region<-c(-60,60)
#lat_region<-c(-60,60)
#
#lon<-seq(min(lon_region),max(lon_region),2)
#lat<-seq(min(lat_region),max(lat_region),2)
#
#data<-RFsim(coordx=lon,coordy=lat,corrmodel="exponential",
#          distance="Geod",grid=TRUE,param=list(nugget=0,mean=0
#          ,scale=8000,sill=1))$data
#
#image.plot(lon,lat,data,xlab="Longitude",ylab="Latitude")
#map(database="world",xlim=lon_region,ylim=lat_region,add=TRUE)
```

`us.coords` *Gauging Stations of the Annual Maximum Rainfalls in U.S.*

Description

A (46×3)-matrix containing the coordinates of the 46 gauging stations in U.S. where the daily precipitation are recorded.

Usage

```
data(usrain)
```

Format

A numerical matrix containing the longitude, the latitude and the altitude of the gauging stations.

Source

Padoan, S. A. Ribatet, M. and Sisson, S. A. (2010) Likelihood-Based Inference for Max-Stable Processes. *Journal of the American Statistical Association, Theory & Methods*, **105**, 263–277.

`usrain` *Annual Maximum Rainfalls in U.S.*

Description

A (91×46)-matrix containing annual maximum of daily rainfalls, in millimeters, for 91 years at 46 gauging stations in U.S.

Usage

```
data(usrain)
```

Format

A numerical matrix containing 4186 observations.

Source

Padoan, S. A. Ribatet, M. and Sisson, S. A. (2010) Likelihood-Based Inference for Max-Stable Processes. *Journal of the American Statistical Association, Theory & Methods*, **105**, 263–277.

`winds` *Irish Daily Wind Speeds*

Description

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

Usage

```
data(irishwinds)
```

Format

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

Source

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

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

Description

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

Usage

```
data(irishwinds)
```

Format

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

Source

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

`WLeastSquare` *WLS of Gaussian and Max-Stable Random Fields*

Description

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

Usage

```
WLeastSquare(data, coordx, coordy=NULL, coordt=NULL, corrmodel,
             distance="Eucl", fixed=NULL, grid=FALSE, maxdist=NULL,
             maxtime=NULL,  model='Gaussian', optimizer='Nelder-Mead',
             numbins=NULL, replicates=1, start=NULL, weighted=FALSE)
```

Arguments

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

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

coordy	A numeric vector giving 1-dimension of spatial coordinates; coordy is interpreted only if coordx is a numeric vector or grid=TRUE otherwise it will be ignored. Optional argument, the default is NULL then coordx is expected to be numeric a $(d \times 2)$-matrix.
coordt	A numeric vector giving 1-dimension of temporal coordinates. Optional argument, the default is NULL then a spatial random field is expected.
corrmodel	String; the name of a correlation model, for the description (see FitComposite).
distance	String; the name of the spatial distance. The default is Eucl, the euclidean distance. See the Section **Details** of FitComposite.
fixed	A named list giving the values of the parameters that will be considered as known values. The listed parameters for a given correlation function will be not estimated, i.e. if list(nugget=0) the nugget effect is ignored.
grid	Logical; if FALSE (the default) the data are interpreted as a vector or a $(n \times d)$-matrix, instead if TRUE then $(d \times d \times n)$-matrix is considered.
maxdist	A numeric value denoting the maximum distance, see **Details** and FitComposite.
maxtime	Numeric; an optional positive value indicating the maximum temporal lag considered in the composite-likelihood computation (see FitComposite.
model	String; the type of random field. Gaussian is the default, see FitComposite for the different types.
optimizer	String; the optimization algorithm (see optim for details). 'Nelder-Mead' is the default.
numbins	A numeric value denoting the numbers of bins, see the Section **Details**
replicates	Numeric; a positive integer denoting the number of independent and identically distributed (iid) replications of a spatial or spatial-temporal random field. Optional argument, the default value is 1 then a single realisation is considered.
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 FitComposite).
weighted	Logical; if TRUE then the weighted least square estimator is considered. If FALSE (the default) then the classic least square is used.

Details

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

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

Value

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

bins	Adjacent intervals of grouped distances;
bint	Adjacent intervals of grouped temporal separations
centers	The centers of the bins;
coordx	The vector or matrix of spatial coordinates;
coordy	The vector of spatial coordinates;

`coordt`	The vector of temporal coordinates;
`convergence`	A string that denotes if convergence is reached;
`corrmodel`	The correlation model;
`data`	The vector or matrix of data;
`distance`	The type of spatial distance;
`fixed`	The vector of fixed parameters;
`iterations`	The number of iteration used by the numerical routine;
`message`	Extra message passed from the numerical routines;
`model`	The type of random fields;
`numcoord`	The number of spatial coordinates;
`numrep`	The number of the iid replicatations of the random field;
`numtime`	The number the temporal realisations of the random field;
`param`	The vector of parameters' estimates;
`srange`	The minimum and maximum spatial distance;
`trange`	The minimum and maximum temporal separations;
`variograms`	The empirical spatial variogram;
`variogramt`	The empirical temporal variogram;
`variogramst`	The empirical spatial-temporal variogram;
`weighted`	A logical value indicating if its the weighted method;
`wls`	The value of the least squares at the minimum.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, http://faculty.unibocconi.it/simonepadoan; Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>, http://staff.deuv.cl/bevilacqua/.

References

Barry, J. T., Crowder, M. J. and Diggle, P. J. (1997) *Parametric estimation of the variogram*. Tech. Report, Dept Maths & Stats, Lancaster University.

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

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

Smith, R. L. (1990) Max-Stable Processes and Spatial Extremes. *Unpublished manuscript*, University of North California.

See Also

`FitComposite`, `optim`

Examples

```
library(CompRandFld)
library(RandomFields)
set.seed(2111)

# Set the coordinates of the sites:
x <- runif(100, 0, 10)
y <- runif(100, 0, 10)


################################################################
###
### Example 1. Least square fitting of a Gaussian random field
### with exponential correlation.
### One spatial replication is simulated.
### Unweighted version (all weights equals to 1).
###
###############################################################

# Set the model's parameters:
corrmodel <- "exponential"
mean <- 0
sill <- 1
nugget <- 0
scale <- 2

# Simulation of the Gaussian random field:
data <- RFsim(x, y, corrmodel=corrmodel, param=list(mean=mean,
              sill=sill, nugget=nugget, scale=scale))$data

fix=list(nugget=0)
ini=list(scale=scale,sill=sill)
# Least square fitting of the random field:
fit <- WLeastSquare(data, x, y, corrmodel=corrmodel,fixed=fix,start=ini)

# Results:
print(fit)


################################################################
###
### Example 2. Least square fitting of a max-stable random field
### (Extremal Gaussian model) with exponential correlation
### n iid spatial replications.
### Unweighted version (all weights equals to 1).
###
###############################################################


# Simulation of the max-stable random field:
data <- RFsim(x, y, corrmodel=corrmodel, model="ExtGauss",
              param=list(mean=mean, sill=sill, nugget=nugget,
              scale=scale), replicates=40)$data

# Least square fitting of the random field:
fit <- WLeastSquare(data, x, y, corrmodel=corrmodel, model="ExtGauss",
```

```
                    replicates=40)

# Results:
print(fit)

################################################################
###
### Example 3. Least square fitting of a spatio-temporal
### Gaussian random field with double exponential correlation.
### One replication is simulated.
### Weighted version (all weights equals to 1).
###
###############################################################

# Define the temporal sequence:
#time <- seq(1, 25, 1)

# Simulation of the Gaussian random field:
#data <- RFsim(x, y, time, corrmodel="exp_exp", param=list(mean=mean,
#             scale_s=scale,scale_t=1,sill=sill,nugget=nugget))$data

#fix<-list(nugget=nugget)
#ini<-list(scale_s=scale,scale_t=1,sill=1)
# Weighted least square estimation:
#fit <- WLeastSquare(data, x, y, time, corrmodel="exp_exp", maxdist=5,
#                    maxtime=5,fixed=fix,start=ini)

# Results
#print(fit)
```

WlsInit *Computes Starting Values based on Weighted Least Squares*

Description

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

Usage

```
WlsInit(coordx, coordy, coordt, corrmodel, data, distance, fcall, fixed,
        grid, likelihood, margins, maxdist, maxtime, model, numblock,
        param, parscale, paramrange, replicates, start, taper, tapsep,
        threshold, type, varest, vartype, weighted, winconst, winstp)
```

Arguments

`coordx`	A numeric $(d \times 2)$-matrix (where `d` is the number of points) assigning 2-dimensions of coordinates or a numeric vector assigning 1-dimension of coordinates.
`coordy`	A numeric vector assigning 1-dimension of coordinates; `coordy` is interpreted only if `coordx` is a numeric vector otherwise it will be ignored.
`coordt`	A numeric vector assigning 1-dimension of temporal coordinates.
`corrmodel`	String; the name of a correlation model, for the description.

`data`	A numeric vector or a $(n \times d)$-matrix or $(d \times d \times n)$-matrix of observations.
`distance`	String; the name of the spatial distance. The default is `Eucl`, the euclidean distance. See the Section **Details**.
`fcall`	String; "fitting" to call the fitting procedure and "simulation" to call the simulation procedure.
`fixed`	A named list giving the values of the parameters that will be considered as known values.
`grid`	Logical; if `FALSE` (the default) the data are interpreted as a vector or a $(n \times d)$-matrix, instead if `TRUE` then $(d \times d \times n)$-matrix is considered.
`likelihood`	String; the configuration of the composite likelihood.
`margins`	String; the type of the marginal distribution of the max-stable field.
`maxdist`	Numeric; an optional positive value indicating the maximum spatial distance considered in the composite-likelihood computation.
`maxtime`	Numeric; an optional positive value indicating the maximum temporal separation considered in the composite-likelihood computation.
`model`	String; the name of the model. Here the default is `NULL`.
`numblock`	Numeric; the observation size of the underlying random field. Only in case of max-stable random fields and in the simulation.
`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.
`replicates`	Logical; if `FALSE` (the default) one spatial random field is considered, instead if `TRUE` the data are considered as iid replicates of a field.
`start`	A numeric vector with starting values.
`taper`	String; the name of the type of covariance matrix. It can be `Standard` (the default value) or `Tapering` for taperd covariance matrix.
`tapsep`	Numeric; an optional value indicating the separabe parameter in the space time quasi taper (see **Details**).
`threshold`	Numeric; a value indicating a threshold for the binary random field.
`type`	String; the type of estimation method.
`varest`	Logical; if `TRUE` the estimates' variances and standard errors are returned. `FALSE` is the default.
`vartype`	String; the type of estimation method for computing the estimate variances, see the Section **Details**.
`weighted`	Logical; if `TRUE` the likelihood objects are weighted, see `FitComposite`.
`winconst`	Numeric; a positive real value indicating the window size used from the sub-sampling method for the estimation of the parameters variances..
`winstp`	Numeric; a value in $(0, 1]$ for defining the window step. See `FitComposite`.

Author(s)

Simone Padoan, `<simone.padoan@unibocconi.it>`, `http://faculty.unibocconi.it/simonepadoan`; Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`, `http://staff.deuv.cl/bevilacqua/`.

See Also

`FitComposite`, `WLeastSquare`.