Package ‘CompRandFld’

November 30, 2010

Version 0.9

Date 2010-11-30

Title Composite-likelihood based Analysis of Random Fields

Author Simone Padoan <simone.padoan@epfl.ch>, Moreno Bevilacqua <moreno.bevilacqua@unive.it>

Maintainer Simone Padoan <simone.padoan@epfl.ch>

Depends R (>= 2.9.0)

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

License GPL Version 2 or later

URL http://people.epfl.ch/simone.padoan

R topics documented:

`CheckCorrModel` *Check of the Correlation Model*

Description

Subroutine called by InitParam. The procedure controls if the correlation model inserted has been implemented.

Usage

```
CheckCorrModel(corrmodel)
```

Arguments

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

Author(s)

Simone Padoan, `<simone.padoan@epfl.ch>`, `http://people.epfl.ch/simone.padoan`.

See Also

`FitComposite`

`CheckInput` *Check of the input*

Description

Subroutine called by the fitting procedures. The procedure controls the input passed to the fitting procedures.

Usage

```
CheckInput(coordx, coordy, corrmodel, data, fixed,
           grid, likelihood, lonlat, model, optimizer,
           replicates, start, type, varest, vartype,
           weighted, weights, winconst)
```

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.
`corrmodel`	String; the name of a correlation model, for the description see `CovarianceFct`.
`data`	A numeric vector or a $(n \times d)$-matrix or $(d \times d \times n)$-matrix of observations.
`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.
`lonlat`	Logical; if `FALSE` (the default), `coordx` and `coordy` are interpreted as Cartesian coordinates otherwise they are considered as longitude and latitude.
`model`	String; the density associated to the likelihood objects. `Gaussian` is the default.
`optimizer`	String; the optimization algorithm (see `optim` for details). 'Nelder-Mead' is the default.
`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.
`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.
`weights`	A numeric vector of weights.
`winconst`	Numeric; a positive real value – if `vartype=Sub-Samp` – that determines the window size in the sub-sampling estimates of the variances, see `FitComposite`.

Author(s)

Simone Padoan, `<simone.padoan@epfl.ch>`, `http://people.epfl.ch/simone.padoan`.

See Also

`FitComposite`

CheckLikelihood *Check of the type of Composite-likelihood*

Description

Subroutine called by InitParam. The procedure controls the type of the composite-likelihood passed to the FitComposite procedure.

Usage

```
CheckLikelihood(likelihood)
```

Arguments

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

Author(s)

Simone Padoan, <simone.padoan@epfl.ch>, http://people.epfl.ch/simone.padoan.

See Also

FitComposite

CheckModel *Check of the type of Random Field*

Description

Subroutine called by InitParam. The procedure controls the type of random field passed to the fitting procedures.

Usage

```
CheckModel(model)
```

Arguments

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

Author(s)

Simone Padoan, <simone.padoan@epfl.ch>, http://people.epfl.ch/simone.padoan.

See Also

FitComposite

CheckParam — *Check of the Parameters*

Description

Subroutine called by InitParam. The procedure controls the validity of the correlation's parameters.

Usage

```
CheckParam(corrmodel, namesparam, numparam)
```

Arguments

`corrmodel`	String; the name of a correlation model.
`namesparam`	String; the names of the parameters.
`numparam`	Numeric; the number of the parameters.

Author(s)

Simone Padoan, `<simone.padoan@epfl.ch>`, `http://people.epfl.ch/simone.padoan`.

See Also

`FitComposite`

CheckParamRange — *Check of the Parameters' Ranges*

Description

Subroutine called by the fitting procedures. The procedure controls the range of the correlation's parameters.

Usage

```
CheckParamRange(param)
```

Arguments

`param`	Numeric; a vector of correlation's parameters.

Author(s)

Simone Padoan, `<simone.padoan@epfl.ch>`, `http://people.epfl.ch/simone.padoan`.

See Also

`FitComposite`

`CheckType` *Check of the type of likelihood objects*

Description

Subroutine called by InitParam. The procedure controls the type of likelihood objects that form the composite-likelihood .

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@epfl.ch>`, `http://people.epfl.ch/simone.padoan`.

See Also

`FitComposite`

`CheckVarType` *Check of the method for the computation of the estimates' variances*

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@epfl.ch>`, `http://people.epfl.ch/simone.padoan`.

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, flagcorr,
               flagnuis, fixed, grid, likelihood, lonlat,
               lower, model, namescorr, namesnuis, namesparam,
               numcoord, numdata, numparam, numparamcorr,
               optimizer, param, type, upper, varest, vartype,
               winconst)
```

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.
`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`.
`lonlat`	Logical; if `FALSE` (the default), `coordx` and `coordy` are interpreted as Cartesian coordinates otherwise they are considered as longitude and latitude.
`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.
`numdata`	Numeric; the number of data replications in time.
`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.
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 real value – if vartype=Sub-Samp – that determines the window size in the sub-sampling estimates of the variances, see FitComposite.

Author(s)

Simone Padoan, <simone.padoan@epfl.ch>, http://people.epfl.ch/simone.padoan.

See Also

FitComposite

CorrelationFct *Computes the Correlation function*

Description

Subroutine called by Covariogram. The procedure computes the estimated correlation function for a given fitted model.

Usage

```
CorrelationFct(corrmodel, lags, param)
```

Arguments

corrmodel	Numeric; the id of the correlation model.
lags	A numeric vector of distances between points.
param	A numeric vector with the parameter values.

Author(s)

Simone Padoan, <simone.padoan@epfl.ch>, http://people.epfl.ch/simone.padoan.

See Also

FitComposite

`CorrelationParam` *Lists the Parameters of the Correlation function*

Description

Subroutine called by DetectParam and InitParam. The procedure returns the list of the parameter for a given correlation model.

Usage

```
CorrelationParam(corrmodel)
```

Arguments

`corrmodel`	String; the name of a correlation model.

Author(s)

Simone Padoan, `<simone.padoan@epfl.ch>`, `http://people.epfl.ch/simone.padoan`.

See Also

`FitComposite`

`Covariogram` *Computes (and Plots) the covariance function and the variogram*

Description

The procedure computes and plots the estimated covariance function and the variogram from a fitted model obtained fitting a random field with the composite-likelihood or using the weighted least square method.

Usage

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

Arguments

`fitted`	The fitted object obtained from the `FitComposite` procedure.
`lags`	A numeric vector of distances.
`answer.cov`	Logical; if `TRUE` a vector with the estimated covariance function is returned; if `FALSE` (the default) the covariance is not returned.
`answer.vario`	Logical; if `TRUE` a vector with the estimated variogram is returned; if `FALSE` (the default) the variogram is not returned.

`answer.range` Logical; if `TRUE` the estimated pratical range is returned; if `FALSE` (the default) the pratical range is not returned.

`show.cov` Logical; if `TRUE` the estimated covariance function is plotted; if `FALSE` (the default) the covariance function is not plotted.

`show.vario` Logical; if `TRUE` the estimated variogram is plotted; if `FALSE` (the default) the variogram is not plotted.

`show.range` Logical; if `TRUE` the estimated pratical range is added on the plot; if `FALSE` (the default) the pratical range is not added.

`add.cov` Logical; if `TRUE` the vector of the estimated covariance function is added on the current plot; if `FALSE` (the default) the covariance is not added.

`add.vario` Logical; if `TRUE` the vector with the estimated variogram is added on the current plot; if `FALSE` (the default) the correlation is not added.

`pract.range` Numeric; the percent of the sill to be reached.

`vario` An object of type `Variogram` obtained from `EVariogram`.

`...` other optional parameters which are passed to plot function.

Value

The returned object is a list with:

`covariance` The vector of the estimated covariance function;

`variogram` The vector of the estimated variogram function;

`pratical.range` The estimated practial range.

Author(s)

Simone Padoan, <simone.padoan@epfl.ch>, http://people.epfl.ch/simone.padoan.

References

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

Examples

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

# Set the coordinates of the points:
x <- runif(100, 0, 20)
y <- runif(100, 0, 20)


################################################################
###
### Example 1. Plot of the estimated correlation function
### from a spatial realisation of a Gaussian random field.
###
###
###############################################################

# Set the model's parameters:
```

```
corrmodel <- "stable"
mean <- 0
variance <- 3
nugget <- 1
scale <- 10
power <- 1.5

# Simulation of the Gaussian random field in the specified points:
sim <- GaussRF(x=x, y=y, model=corrmodel, grid=FALSE,
               param=c(mean, variance, nugget, scale, power))

# Maximum composite-likelihood fitting of the random field:
fit <- FitComposite(x, y, corrmodel, sim)

# Plot of the Correlation function:
par(mfrow=c(1,2))
Covariogram(fit, show.cov=TRUE, show.range=TRUE, show.vario=TRUE)
```

DetectParam *Identifies the Parameters of the Correlation function*

Description

Subroutine called by Covariogram and others The procedure returns a list with the correlation model and the list of parameters.

Usage

```
DetectParam(corrmodel, fixed, param)
```

Arguments

corrmodel	String; the name of a correlation model.
fixed	A numeric vector with the fixed parameters.
param	A numeric vector with the parameters.

Author(s)

Simone Padoan, <simone.padoan@epfl.ch>, http://people.epfl.ch/simone.padoan.

See Also

FitComposite

`Dist2Dist` *Switches from an Extreme Value Distribution to Another Extreme Value Distribution*

Description

the function transforms a sequence of values following an extreme value distribution to a sequence with another desired extreme value distribution.

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@epfl.ch>, http://people.epfl.ch/simone.padoan

References

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

See Also

FitGev

EVariogram	*Empirical Estimation of the Variogram for Gaussian and Max-Stable Random Fields*

Description

the function returns an empirical estimate of the variogram for a Gaussian or a max-stable random field.

Usage

```
EVariogram(coordx, coordy, data, cloud=FALSE, extcoeff=FALSE, grid=FALSE,
           gev=c(0,1,0), lonlat=FALSE, maxdist=NULL, numbins=NULL,
           replicates=FALSE, type='variogram')
```

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.
`data`	A numeric vector or a $(n \times d)$-matrix or $(d \times d \times n)$-matrix of observations (see `FitComposite`).
`cloud`	Logical; if `TRUE` the variogram cloud is computed, otherwise if `FALSE` (the default) the empirical (binned) variogram is returned.
`extcoeff`	Logical; if the extremal coefficient estimates need to be returned. Available only if `type` is equal to `madogram` or `Fmadogram`.
`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.
`gev`	A numeric vector with the three GEV parameters;
`lonlat`	Logical; if `FALSE` (the default), `coordx` and `coordy` are interpreted as Cartesian coordinates otherwise they are considered as longitude and latitude.
`maxdist`	A numeric value denoting the maximum distance, see the Section **Details**.

`numbins`	A numeric value denoting the numbers of bins, see the Section **Details**.
`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.
`type`	A String denoting the type of variogram. Three options are available, `variogram` which return the standard variogram and for extreme values the `madogram` or `Fmadogram`.

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

`bins`	Adjacent intervals of grouped distances if `cloud=FALSE`. Otherwise the pairwise distances if `cloud=TRUE`;
`cloud`	If the variogram cloud is returned (`TRUE`) or the empirical variogram (`FALSE`);
`centers`	The centers of the bins;
`extremalcoeff`	The extremal coefficient estimates;
`lenbins`	The number of pairs in each bin;
`maxdist`	The maximum distance used for the calculation of the variogram;
`variogram`	The empirical variogram;
`type`	The type of estimated variogram: the standard variogram or the madogram.

Author(s)

Simone Padoan, `<simone.padoan@epfl.ch>`, `http://people.epfl.ch/simone.padoan`.

References

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

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

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

See Also

`FitComposite`

Examples

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

# Set the coordinates of the sites:
x <- runif(100, 0, 20)
y <- runif(100, 0, 20)


################################################################
###
### Example 1. Empirical variogram estimate of one
### spatial realisation of a Gaussian random field.
###
###
###############################################################

# Set the model's parameters:
corrmodel <- "stable"
mean <- 0
variance <- 1
nugget <- 0
scale <- 10
power <- 1.5

# Simulation of a Gaussian random field in the specified points:
sim <- GaussRF(x=x, y=y, model=corrmodel, grid=FALSE,
              param=c(mean, variance, nugget, scale, power))

# Empirical variogram estimation:
fit <- EVariogram(x, y, sim)

# Results:
plot(fit$centers, fit$variogram, xlab='h', ylab=expression(gamma(h)),
     ylim=c(0, max(fit$variogram)), xlim=c(0, fit$maxdist), pch=20)


################################################################
###
### Example 2. Empirical mandogram estimate of one
### spatial realisation of a max-stable random field.
###
###
###############################################################

# Simulation of a max-stable random field in the specified points:
sim <- MaxStableRF(x=x, y=y, model=corrmodel, grid=FALSE, maxstable="extr",
                   param=c(mean, variance, nugget, scale, power), n=100)
sim <- t(sim)
sim <- Dist2Dist(sim, to='sGumbel')

# Empirical madogram estimation:
fit <- EVariogram(x, y, sim, type='madogram', replicates=TRUE,
                  extcoeff=TRUE, cloud=TRUE)

# Results:
```

```
par(mfrow=c(1,2))
plot(fit$centers, fit$variogram, xlab='h', ylab=expression(nu(h)),
     ylim=c(0, max(fit$variogram)), xlim=c(0, fit$maxdist), pch=20)
plot(fit$centers, fit$extremalcoeff, xlab='h', ylab=expression(theta(h)),
     ylim=c(1, 2), xlim=c(0, fit$maxdist), pch=20)
```

FitComposite   *Maximum Weighted Composite-likelihood Fitting of Gaussian Random Fields*

Description

Maximum weighted composite-likelihood fitting for Gaussian and max-stable random fields. The function returns the model parameters' estimates and the estimates' variances by weighted maximisation of the composite-likelihood and allows to fix any of the parameters.

Usage

```
FitComposite(coordx, coordy=NULL, corrmodel, data, fixed=NULL, grid=FALSE,
             likelihood='Marginal', lonlat=FALSE, model='Gaussian',
             optimizer='Nelder-Mead', replicates=FALSE, start=NULL,
             type='Pairwise', varest=FALSE, vartype='SubSamp',
             weighted=FALSE, weights=NULL, winconst=NULL)
```

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`	String; the name of a correlation model, for the description see the Section **Details**.
`data`	A numeric vector or a ($n \times d$)-matrix or ($d \times d \times n$)-matrix of observations (see **Details**).
`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, see the Section **Details**.
`lonlat`	Logical; if `FALSE` (the default), `coordx` and `coordy` are interpreted as Cartesian coordinates otherwise they are considered as longitude and latitude.
`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`	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 (see **Details**).
`type`	String; the type of the likelihood objects. If `Pairwise` (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods (see **Details**).
`varest`	Logical; if `TRUE` the estimate' variances and standard errors are returned. `FALSE` is the default.
`vartype`	String; (`Sub-Samp` 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.
`weights`	A numeric vector of weights (see **Details**).
`winconst`	Numeric; a positive real value – if `vartype=Sub-Samp` – that determines the window size in the sub-sampling estimates of the variances (see **Details**).

Details

The `corrmodel` parameter allows to select a specific correlation function for the random field. The implemented correlation models are:

1. `cauchy`;
2. `exponential`;
3. `gauss` (Gaussian);
4. `gencauchy` (generalised Cauchy);
5. `stable` (or powered exponential);
6. `whittlematern` (Whittle-Matern).

See for more details `CovarianceFct`.

With the `data` parameter:

- If it is a numeric vector, the data are interpreted as a spatial or temporal realisation;
- If it is a numeric $(n \times d)$-matrix, the `d` columns represent the different points where the data observed and the `n` rows represent the available iid replications.
- If it is a numeric $(d \times d \times n)$-matrix then the data are considered as observations observed at $(d \times d)$ points with eventually `n` available replications.

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 `model` paramter represents the type of random field considered, an example is the Gaussian random field. Therefore, this also determines the associated density functions that will define the likelihood objects and then the composite-likelihood. The available options:

- `Gaussian`, for a Gaussian random field.

The `replicates` parameter specifies if iid replications of the random field are available. If `FALSE`, then a realisation of spatial random field is considered. Instead if `TRUE` the `data` are considered as iid replications of a spatial random field. For the moment the option of iid replications is implemented only for a spatial random field. The alternative is to work with a realisation of a spatial-temporal random field.

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 `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 likelihood that will be considered is the standard multivariate version;
        - `Restricted`, the likelihood that will be considered is the restricted version (see **References**).

The `vartype` parameter – if the `varest` is `TRUE` – specifies the method used to compute the estimates' variances. The default `Sub-Samp` uses the Sub-Sampling method to estimate the variability matrix in the Godambe matrix. Other options are: `Theoretical` where for the variability matrix it is used the exact expression (for the moment it has been implemented only for the `Difference` likelihood, see the `type` field) and `Sampling` where the variability matrix in this case is estimated by the sample contro-part (available only for iid. replicates of a random field, see the `replicates` parameter).

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.

The `weights` parameter allows to weight the composite-likelihood by weights insert by the users. These do not imply any gain in efficiency of the maximum composite-likelihood estimator but still be a reasonable setting (not implemented yet!).

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 conicide with the Akaike information criterion;
`coord`	The vector of coordinates;
`convergence`	A string that denotes if convergence is reached;
`corrmodel`	The correlation model;
`data`	The vector or matrix of data;

`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;
`lonlat`	The type of coordinates;
`message`	Extra message passed from the numerical routines;
`model`	The density associated to the likelihood objects;
`param`	The vector of parameters' estimates;
`stderr`	The vector of standard errors;
`sensmat`	The sensitivity matrix;
`varcov`	The matrix of the variance-covariance of the estimates;
`varimat`	The variability matrix;
`type`	The type of the likelihood objects.

Author(s)

Simone Padoan, <simone.padoan@epfl.ch>, http://people.epfl.ch/simone.padoan; Moreno Bevilacqua, <moreno.bevilacqua@unive.it>.

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.

Composite-likelihood:

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

Varin, C. (2008) On Composite Marginal Likelihoods. *Advances in Statistical Analysis*, **92**, 1–28.

Weighted Composite-likelihood for max-stable random fields:

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. (2011) Estimating space and space-time covariance functions for large data sets: a weighted composite likelihood approach. *Journal of the American Statistical Association*, to appear.

Spatial Statistics:

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

Wackernagel, H. (1998) *Multivariate Geostatistics*. Berlin: Springer, 2nd edition.

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

Spatial Extremes:

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

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

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

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.

Kabluchko, Z. (2010) Extremes of Independent Gaussian Processes. *Extremes*, to appear.

Sub-sampling estimation:

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.

See Also

CovarianceFct, WLeastSquare, optim

Examples

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

# Set the coordinates of the points:
x <- runif(100, 0, 20)
y <- runif(100, 0, 20)


################################################################
###
### Example 1. Maximum composite-likelihood fitting of one
### spatial realisation of a Gaussian random field.
### Composite-likelihood setting: pairwise marginal likelihoods.
###
###############################################################

# Set the model's parameters:
corrmodel <- "stable"
mean <- 0
variance <- 1
nugget <- 0
scale <- 10
power <- 1.5

# Simulation of the Gaussian random field in the specified points:
sim <- GaussRF(x=x, y=y, model=corrmodel, grid=FALSE,
               param=c(mean, variance, nugget, scale, power))

# Maximum composite-likelihood fitting of the random field:
fit <- FitComposite(x, y, corrmodel, sim)

# Results:
print(fit)
```

```
################################################################
###
### Example 2. Maximum composite-likelihood fitting of one
### spatial realisation of a Gaussian random field.
### Composite-likelihood setting: difference likelihoods.
###
###############################################################

# Set the model's parameters:
corrmodel <- "stable"
mean <- 0
variance <- 1
nugget <- 0
scale <- 10
power <- 1.5

# Simulation of the Gaussian random field in the specified points:
sim <- GaussRF(x=x, y=y, model=corrmodel, grid=FALSE,
               param=c(mean, variance, nugget, scale, power))

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

# Results:
print(fit)


################################################################
###
### Example 3. Maximum likelihood fitting of one
### spatial realisation of a Gaussian random field.
### Likelihood setting: restricted likelihood.
###
###############################################################

# Set the model's parameters:
corrmodel <- "stable"
mean <- 0
variance <- 1
nugget <- 0
scale <- 10
power <- 1.5

# Simulation of the Gaussian random field in the specified points:
sim <- GaussRF(x=x, y=y, model=corrmodel, grid=FALSE,
               param=c(mean, variance, nugget, scale, power))

# Maximum composite-likelihood fitting of the random field:
fit <- FitComposite(x, y, corrmodel, sim, likelihood='Full',
                    type='Restricted')

# Results:
print(fit)
```

FitGev	*Maximum-likelihood Fitting of the Generalized Extreme Value 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@epfl.ch>`, `http://people.epfl.ch/simone.padoan`

References

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

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

See Also

`GevLogLik`, `optim`

GevLogLik — *Log-Likelihood of the Generalized Extreme Value 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@epfl.ch>`, http://people.epfl.ch/simone.padoan

References

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

InitParam — *Initializes of the Fitting Procedures*

Description

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

Usage

```
InitParam(coordx, coordy, corrmodel, data, fixed, grid,
          likelihood, lonlat, model, parscale, paramrange,
          replicates, start, type, vartype, weighted)
```

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	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.
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.
lonlat	Logical; if FALSE (the default), coordx and coordy are interpreted as Cartesian coordinates otherwise they are considered as longitude and latitude.
model	String; the density associated to the likelihood objects. Gaussian is the default.
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. NULL is the default (see **Details**).
type	String; the type of the likelihood objects. If Pairwise (the default) then the marginal composite likelihood is formed by pairwise marginal likelihoods.
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.

Author(s)

Simone Padoan, <simone.padoan@epfl.ch>, http://people.epfl.ch/simone.padoan.

See Also

FitComposite

Likelihood *Optimizes of the log-likelihood*

Description

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

Usage

```
Likelihood(corrmodel, data, fixed, grid, lower, model, namescorr,
           namesnuis, namesparam, numcoord, numdata, numpairs,
           optimizer, param, varest, 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.
`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.
`numdata`	Numeric; the number of data replications in time.
`numpairs`	Numeric; the number of pairwise distances.
`optimizer`	String; the optimization algorithm (see `optim` for details). 'Nelder-Mead' is the default.
`param`	A numeric vector of parameters' values.
`varest`	Logical; if `TRUE` the estimate' variances and standard errors are returned. `FALSE` is the default.
`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@epfl.ch>`, `http://people.epfl.ch/simone.padoan`.

See Also

`FitComposite`

LogNormDen *Computes the multivariate log-normal density*

Description

Subroutine called by the Likelihood procedure. The procedure compute the multivariate log-normal density for a given set of data and parameters.

Usage

```
LogNormDen(stdata, detvarcov, ivarcov, numcoord, type)
```

Arguments

`stdata`	A numeric vector $(d \times 1)$ of data.
`detvarcov`	Numeric; the determinant of the variance-covariance matrix.
`ivarcov`	The inverse of the variance-covariance matrix $(d \times d)$.
`numcoord`	The number of point's coordinates.
`type`	The numeric id denoting the type of likelihood.

Author(s)

Simone Padoan, <simone.padoan@epfl.ch>, http://people.epfl.ch/simone.padoan.

See Also

FitComposite

MomEst *Moment Estimator of the Extreme Value Distribution's parameters*

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@epfl.ch>`, http://people.epfl.ch/simone.padoan

References

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

See Also

`GevLogLik`, `FitGev`

`SetRangeParam` *Identifies the Parameters' Range*

Description

Subroutine called by InitParam and the fitting procedures. The procedure returns the range of the parameters for a given vector of parameters.

Usage

```
SetRangeParam(namesparam, numparam)
```

Arguments

`namesparam`	String; the names of the parameters.
`numparam`	Numeric; the numer of parameters.

Author(s)

Simone Padoan, `<simone.padoan@epfl.ch>`, http://people.epfl.ch/simone.padoan.

See Also

`FitComposite`

WLeastSquare *Weighted Least Square Estimation of Random Fields*

Description

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

Usage

```
WLeastSquare(coordx, coordy, corrmodel, data, fixed=NULL, grid=FALSE,
             lonlat=FALSE, maxdist=NULL, model='Gaussian',
             optimizer='Nelder-Mead', numbins=NULL,
             replicates=FALSE, start=NULL, weighted=FALSE)
```

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`	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 (see `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.
`lonlat`	Logical; if `FALSE` (the default), `coordx` and `coordy` are interpreted as Cartesian coordinates otherwise they are considered as longitude and latitude.
`maxdist`	A numeric value denoting the maximum distance, see the Section **Details**.
`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`	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 (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;
coord	The vector of coordinates;
convergence	A string that denotes if convergence is reached;
corrmodel	The correlation model;
data	The vector or matrix of data;
fixed	The vector of fixed parameters;
iterations	The number of iteration used by the numerical routine;
message	Extra message passed from the numerical routines;
param	The vector of parameters' estimates;
variogram	The empirical variogram.

Author(s)

Simone Padoan, <simone.padoan@epfl.ch>, http://people.epfl.ch/simone.padoan; Moreno Bevilacqua, <moreno.bevilacqua@unive.it>.

References

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

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

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

See Also

FitComposite, optim

Examples

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

# Set the coordinates of the sites:
x <- runif(100, 0, 20)
y <- runif(100, 0, 20)


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

```
### Example 1. Least square fitting of one
### spatial realisation of a Gaussian random field.
### Non weighted version (all weights equals to 1)
###
###############################################################

# Set the model's parameters:
corrmodel <- "stable"
mean <- 0
variance <- 1
nugget <- 0
scale <- 10
power <- 1.5

# Simulation of the Gaussian random field in the specified points:
sim <- GaussRF(x=x, y=y, model=corrmodel, grid=FALSE,
              param=c(mean, variance, nugget, scale, power))

# Least square fitting of the random field:
fit <- WLeastSquare(x, y, corrmodel, sim)

# Results:
print(fit)


################################################################
###
### Example 1. Weighted least square fitting of one
### spatial realisation of a Gaussian random field.
### Weighted version.
###
###############################################################

# Set the model's parameters:
corrmodel <- "stable"
mean <- 0
variance <- 1
nugget <- 0
scale <- 10
power <- 1.5

# Simulation of the Gaussian random field in the specified points:
sim <- GaussRF(x=x, y=y, model=corrmodel, grid=FALSE,
              param=c(mean, variance, nugget, scale, power))

# Least square fitting of the random field:
fit <- WLeastSquare(x, y, corrmodel, sim, weighted=TRUE)

# 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, corrmodel, data, fixed, grid,
        likelihood, lonlat, model, parscale, paramrange,
        replicates, start, type, vartype, weighted)
```

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`	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.
`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.
`lonlat`	Logical; if `FALSE` (the default), `coordx` and `coordy` are interpreted as Cartesian coordinates otherwise they are considered as longitude and latitude.
`model`	String; the name of the model. Here the default is `NULL`.
`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.
`type`	String; the type of estimation method.
`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`.

Author(s)

Simone Padoan, `<simone.padoan@epfl.ch>`, `http://people.epfl.ch/simone.padoan`.

See Also

`FitComposite`, `WLeastSquare`.

Index