Package ‘CompRandFld’

June 21, 2010

Version 0.1

Date 2010-04-06

Title Composite-likelihood based Analysis of Random Fields

Author Simone Padoan <simone.padoan@epfl.ch>, Moreno Bevilacqua <bevilacqua@stat.unipd.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, rootSolve

URL http://eflum.epfl.ch/people/padoan.en.php

R topics documented:

Covariogram *Computation of 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, ...)
```

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.
`...`	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://eflum.epfl.ch/people/padoan.en.php.

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)
```

FitComposite — *Maximum Composite-likelihood Fitting of Random Fields*

Description

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

Usage

```
FitComposite(coordx, coordy=NULL, corrmodel, data, fixed=NULL,
             grid=FALSE, hessian=FALSE, likelihood='Marginal',
             lonlat=FALSE, model='Gaussian', optimizer='Nelder-Mead',
             start=NULL, varest=FALSE, time=FALSE, type='Pairwise',
             weighted=FALSE, weights=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.
`hessian`	Logical; if `FALSE` (the default) the hessian matrix is not computed from the optimizer routine.
`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 density 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.
`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**).
`varest`	Logical; if `TRUE` the estimate' variances and standard errors are returned. `FALSE` is the default.
`time`	Logical; if `FALSE` (the default) a spatial random field is considered (one temporal realisation), if `TRUE` a spatial-temporal random field is considered, 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**).
`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**).

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 a numeric vector, the data are interpreted as one spatial realisation;
- If a numeric $(n \times d)$-matrix, the columns represent the data observed at different points and the rows represent the data for different time steps.
- If a numeric $(d \times d \times n)$-matrix the data are observed at $(d \times d)$ points for `n` time steps.

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

1. `Conditional`, the composite-likelihood is formed by conditionals likelihoods (not implemented yet);
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 density function underlying the definition of the likelihoods which form the composite-likelihood. The settings alternatives are:

- `Gaussian`, the Gaussian density.

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 `time` parameter allows to specify the type of random field. If `FALSE` a spatial random field is considered, if `TRUE` a spatial-temporal random field is used. For the moment the case of i.i.d. time replications is implemented. Soon will be possible to specify also dependence structure for the temporal component.

The `type` parameter represents the type of likelihood used in the composite-likelihood definition. The settings alternatives are:

1. If the composite is formed by `Marginal` likelihoods:
   - If each likelihood is obtained by the `Gaussian` density then with:
     - `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.
2. If the `Full` likelihood is considered:
   - If the likelihood is obtained by the `Gaussian` density then with:
     - `Standard`, the likelihood used is the standard version;
     - `Restricted`, the likelihood used is the restricted version.

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.

Value

The returned object is a list with:

- `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://eflum.epfl.ch/people/padoan.en.php; Moreno Bevilacqua, <moreno.bevilacqua@unive.it>.

References

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

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.

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.

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)



################################################################
###
### Example 4. Maximum composite-likelihood fitting of n i.i.d.
### spatial realisations of a Gaussian random field.
### Composite-likelihood setting: difference likelihoods.
###
###############################################################

numdata <- 30

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

sim <- t(sim)

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

```
fit <- FitComposite(x, y, corrmodel, sim, time=TRUE, type='Difference')


# Results:
print(fit)


#################################################################
###
### Example 5. Maximum composite-likelihood fitting of one
### spatial realisations of a Gaussian random field on a
### regular grid.
### Composite-likelihood setting: difference likelihoods.
###
################################################################

step <- 1
x <- seq(0, 20, step)
y <- seq(0, 20, step)

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

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

# Results:
print(fit)


#################################################################
###
### Example 6. Maximum composite-likelihood fitting of n i.i.d.
### spatial realisations of a Gaussian random field on a
### regular grid.
### Composite-likelihood setting: difference likelihoods.
###
################################################################

numdata <- 5

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

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

# Results:
print(fit)
```

`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 square estimator.

Usage

```
WLeastSquare(coordx, coordy, corrmodel, data, fixed=NULL,
             grid=FALSE, lonlat=FALSE, maxdist=NULL,
             optimizer='Nelder-Mead', numbins=NULL, start=NULL,
             time=FALSE, 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**.
`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**
`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`).
`time`	Logical; if `FALSE` (the default) a spatial random field is considered (one temporal realisation), if `TRUE` a spatial-temporal random field is considered (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

Insert description of `maxdist` and `numbins`.

Value

The returned object is a list with:

- `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://eflum.epfl.ch/people/padoan.en.php; 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.

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)
```

Index