Package ‘STBEU’

November 9, 2019

Version 1.0.0

Date 2018-10-23

Title SpaceTime Blockwise Euclidean Likelihood for Gaussian Models in Geostatistics

Author Moreno Bevilacqua [aut, cre],Víctor Morales-Onate [aut]

Maintainer Victor Morales Onate <victor.morales@uv.cl>

Depends R (>= 2.12.0)

Description This package provides a set of procedures modeling Gaussian Random Fields using Blockwise Euclidean Likelihood using OpenCL

Imports methods, MCMCpack, grDevices, graphics, stats

Suggests spam, scatterplot3d, fields, mapproj, gsl, plot3D, shape, sphereplot,GeoModels

License GPL (>= 2)

URL https://github.com/vmoprojs/STBEU

Repository Github

Encoding UTF-8

R topics documented:

`anomalies` *Annual precipitation anomalies in U.S.*

Description

A $(7252x3)$-matrix containing lon/lat and yearly total precipitation anomalies registered at 7.352 location sites in USA. For more details see http://www.image.ucar.edu/Data/precip_tapering/.

Usage

```
data(anomalies)
```

Format

A numerical matrix of dimension $7252x3$.

Source

Kaufman, C.G., Schervish, M.J., Nychka, D.W. (2008) Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association, Theory & Methods*, **103**, 1545–1555.

`checkpar` *Checking Correlation Model*

Description

The procedure controls if the correlation model inserted is correct.

Usage

```
checkpar(fix,theta,cc)
```

Arguments

`fix`	Numeric; a vector which has fixed parameters info.
`theta`	Numeric; a vector which has to be estimated parameters info.
`cc`	Numeric; it indicates the correlation model

Value

Return the `theta` parameters with correct names.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

CreateBinary — *Create binary file*

Description

Matrix decomposition.

Usage

```
CreateBinary(dev,fname)
```

Arguments

dev	numeric; indicates de dev ID
fname	string; the name of the function

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

DevOpenCL — *Prints Device Information*

Description

Prints the device details available in your computer. Device name, Max compute units, whether it supports double precision, among others.

Usage

```
DevOpenCL()
```

Details

The user can take this information into account so that the `local` parameter is set up in `GeoFit` when `GPU` computation is chosen.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

Examples

```
library(STBEU)
DevOpenCL()
```

eucl_st_ocl *Checking Correlation Model*

Description

The procedure controls if the correlation model inserted is correct.

Usage

```
eucl_st_ocl(theta,fixed,coordx,coordy,ncoords,times,ntime,cc,datos,type_dist,maxdist,maxtime,
                      winc_s,winstp_s,winc_t,winstp_t,
                      weighted,type_sub,local,GPU,kernel)
```

Arguments

theta	Numeric; a vector which has fixed parameters info.
fixed	Numeric; a vector which has to be estimated parameters info.
coordx	Numeric; it indicates the correlation model
coordy	Numeric; it indicates the correlation model
ncoords	Numeric; it indicates the correlation model
times	Numeric; it indicates the correlation model
ntime	Numeric; it indicates the correlation model
cc	Numeric; it indicates the correlation model
datos	Numeric; it indicates the correlation model
type_dist	Numeric; it indicates the correlation model
maxdist	Numeric; it indicates the correlation model
maxtime	Numeric; it indicates the correlation model
winc_s	Numeric; it indicates the correlation model
winstp_s	Numeric; it indicates the correlation model
winc_t	Numeric; it indicates the correlation model
winstp_t	Numeric; it indicates the correlation model
weighted	Numeric; it indicates the correlation model
type_sub	Numeric; it indicates the correlation model
local	Numeric; it indicates the correlation model
GPU	Numeric; it indicates the correlation model
kernel	Character; It has the function name for binary convertion

Value

Return the `theta` parameters with correct names.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

MatDecomp *Matrix decomposition*

Description

Matrix decomposition.

Usage

```
MatDecomp(mtx, method)
```

Arguments

`mtx`	numeric; a square positive or semipositive definite matrix.
`method`	string; the type of matrix decomposition. Two possible choices: `cholesky` and `svd`.

Author(s)

Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, `<victor.morales@uv.cl>`, https://sites.google.com/site/moralesonatevictor/

MatInv *Matrix decomposition*

Description

Matrix decomposition.

Usage

```
MatInv(mat.decomp,method)
```

Arguments

`mat.decomp`	numeric; a square positive or semipositive definite matrix.
`method`	string; the type of matrix decomposition. Two possible choices: `cholesky` and `svd`.

Author(s)

Moreno Bevilacqua, `<moreno.bevilacqua@uv.cl>`,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, `<victor.morales@uv.cl>`, https://sites.google.com/site/moralesonatevictor/

OntarioSummer2018Hour *Ontario Hourly temperature*

Description

Ontario Hourly temperature on 2018.

***** (Not necesasary to run) Final Dataset used in STBEU was generated using the following:

library(weathercan) # library(dplyr) # library(GeoModels) # rm(list = ls()) # graphics.off() # gc() # cat("\014") # # # mb <- filter(stations, # prov == "ON", # interval == "day", # end >= 2018) # select(-prov) # # # mb_weather_all <- weather_dl(station_ids = mb$station_id, # start = "2018-03-20", # interval = "hour", quiet = TRUE) # # # Summer - June 21 to September 21 (YES) # st <- "2018-06-21" # ed <- "2018-09-21" # # # fil <- mb_weather_all$date >= st & mb_weather_all$date <= ed # aux <- mb_weather_all[fil,] # save(aux,file = "OntarioSummer2018Hour.RData")

Usage

```
data(OntarioSummer2018Hour)
```

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.

print.STBEUFit *Matrix decomposition*

Description

Matrix decomposition.

Usage

```
print.STBEUFit(x,names,GPU,varest, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	A
names	B
GPU	C
varest	D
digits	D
...	G

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

`setting_param` *Checking Correlation Model*

Description

The procedure controls if the correlation model inserted is correct.

Usage

```
setting_param(cc,theta,fix)
```

Arguments

`fix`	Numeric; a vector which has fixed parameters info.
`theta`	Numeric; a vector which has to be estimated parameters info.
`cc`	Numeric; it indicates the correlation model

Value

Return the `theta` parameters with correct names.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

`STBEUFit` *Blowise Eucliden Likelihood Fitting of Gaussian Random Fields.*

Description

Blowise Eucliden Likelihood Fitting of Gaussian Random Fields. The function returns the model parameters' estimates and the estimates' variances. Moreover the function allows to fix any of the parameters in the optimization.

Usage

```
STBEUFit(theta,fixed=NULL,coords,times,cc,datos,type_dist=1,maxdist=NULL,maxtime=NULL, winc_s=NU
                winc_t=NULL,winstp_t=NULL,subs=NULL,weighted=FALSE,local=c(1,1),GPU=NULL,vares
```

Arguments

`theta`	A vector of starting parametes for estimation
`fixed`	An optional named vector giving the values of the parameters that will be considered as known values. The default is `NULL`
`coords`	A numeric ($d \times 2$)-matrix (where `d` is the number of spatial sites).
`times`	A numeric vector assigning 1-dimension of temporal coordinates.
`cc`	Numerical, the name of a correlation model. Values $(1, 2)$ are allowed for Double Exponential and Gneting respectively. See the Section **Details**.
`datos`	A t-matrix (a single spatial realisation) where `d` is the number of spatial sites and `t` is the number of temporal coordinates. For the description see the Section **Details**.
`type_dist`	Numerical; `1` is Euclinean and `2` is Geodesic. The default is `1`, the euclidean distance. See the Section **Details**.
`maxdist`	Numeric; an optional positive value indicating the maximum spatial distance considered in the composite likelihood computation. See the Section **Details** for more information.
`maxtime`	Numeric; an optional positive value indicating the maximum temporal separation considered in the euclidean likelihood computation (see **Details**).
`winc_s`	Numeric; a positive value for computing the spatial sub-window in the subsampling procedure. See **Details** for more information.
`winstp_s`	Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the spatial sub-sampling procedure. The case 1 correspond to no overlapping. See **Details** for more information.
`winc_t`	Numeric; a positive value for computing the temporal sub-window in the subsampling procedure. See **Details** for more information.
`winstp_t`	Numeric; a value in $(0, 1]$ for defining the the proportion of overlapping in the temporal sub-sampling procedure. The case 1 correspond to no overlapping. See **Details** for more information.
`subs`	Numeric; a value in $(1, 2, 3)$ for defining the type of sub-sampling procedure. Cases are space, time and spacetime respectively. See **Details** for more information.
`weighted`	Logical; if `TRUE` the likelihood objects are weighted, see the Section **Details**. If `FALSE` (the default) the euclidean likelihood is not weighted.
`local`	Numeric; number of local work-items of the OpenCL setup. Default is $c(1, 1)$
`GPU`	Numeric; if `NULL` (the default) no GPU computation is performed.
`varest`	Logical; if `FALSE` (the default) no varcov matrix is returned.
`optimizer`	string; "method" in optim ("Nelder-Mead" the default).

Details

The optimization method is `Nelder-mead`.

The `cc` corrmodel parameter allows to select a specific correlation function for the RF. Options are: 1 for Double Exponential and 2 for Gneting.

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

1. `Eucl`, the euclidean distance (default value);

2. `Geod`, the geodesic distance;

The `subs` parameter represents the subsampling configurations. The settings alternatives are:

1. `Spatial`, preferred option when the number of space sites is greater than temporal coordinates;
2. `Time`, preferred option when the number of temporal coordinates is greater than space sites;
3. `Spacetime`, preferred option when the number of space and temporal are balaced.

All the nuisance and covariance parameters must be specified by the user using the `start` and the `fixed` parameter. Specifically:

The `start` parameter allows to specify (as starting values for the optimization) the parameters to be estimated. The `fixed` parameter allows to fix some of the parameters.

The `maxdist` parameter set the maximum spatial distance below which pairs of sites with inferior distances are considered in the euclidean-likelihood. This can be lower of the 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 RFs. The default is `NULL`, in this case the effective maximum temporal lag between pairs of observations is considered.

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 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.

Value

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

`param`	The vector of parameters' estimates;

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

References

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 RFs:

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 Gaussian RFs:

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 RFs *Techical Report, Department of Statistics, de Valparaiso University*.

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

Examples

```
#####################################********  SPACE  ***#######################################

#======================Double Exponential======================#

rm(list=ls())
#######################R package######################
require(GeoModels)
require(MCMCpack)
require(STBEU)
####location sites ######################################
lambda=8
xx=seq(-lambda,lambda);
coords=as.matrix(expand.grid(xx,xx))   ###regular
# plot(coords)
#set.seed(15)                                                        ### not regular
#pp<-runifpoint(4*(lambda)^2, win=owin(c(-lambda,lambda),c(-lambda,lambda)))      ### not regular
#coords<-cbind(pp$x,pp$y)
### not regular
####temporal instants ######################################
times=seq(1,5,1)
##################################################################


type_dist=1               ### type of distance     1=euclidean 2=chordal  3=geodesic
maxdist=2           ## compact support in weights function for pairwise liklihood
maxtime=2

model=1  #   1=double exponential       2 =gneiting
```

```
if (model == 1) {
  # exponential model
  cov.model <-"exp_exp"
  cc=1
  #####
  mean=0
  nugget=0
  scale_s<-1.5/3
  scale_t<-1.5/3
  sill=1
  param=list(nugget=nugget,mean=mean,scale_t=scale_t,scale_s=scale_s,sill=sill)
  fixed=list(nugget=0)
  fix=c(nugget=nugget)
}
##################################################################
set.seed(276)
####################################################################################################
# Simulation of the spatial Gaussian random field:
data <- GeoSim(coordx=coords,coordt=times,corrmodel=cov.model, param=param)$data
mm=mean(c(data))
vv=var(c(data))
###### Composite likelihood based on pairs estimation ##############################################

start=list(mean=mm,scale_s=scale_s,scale_t=scale_t,sill=vv)
# Maximum composite-likelihood fitting of the random field:
fit <- GeoFit(data=data,coordx=coords,coordt=times,
                    corrmodel=cov.model,maxtime=maxtime,maxdist=maxdist,
                    likelihood="Marginal",type="Pairwise",
                    start=start,fixed=fixed,weighted=TRUE)
# Results:
print(fit$param)
unlist(start)
################################################
#   parameters for the subsampling ####
###############################################
coordx=coords[,1]
coordy=coords[,2]
LX=abs(range(coordx)[1]-range(coordx)[2])
LY=abs(range(coordy)[1]-range(coordy)[2])

lato_fin=3  #changing window size
lx=lato_fin          #lunghezza lato x quadrato subfinestra
ly=lato_fin          #lunghezza lato y quadrato subfinestra
winc=c(lx/sqrt(LX),ly/sqrt(LY))
winstp= 1  ###   1/lato_fin complete overlapping   1 "no" overlapping
###############################################

####################################################################################################
theta=start                #starting value

weighted=0
### eucliden likelihood ################
type_subs=1    ### type of subsampling  1=in space    2= in time
tCPU = proc.time()
# names(fix) = c("nugget");names(theta) = c("mean","sill","scale_s","scale_t")
res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,0,0,type_subs,we
```

```
tCPU = proc.time()-tCPU;tCPU
res$par


### OpenCL eucliden likelihood ################
local <- c(1,1)
GPU <- 0
tGPU = proc.time()
res1 = STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,0,0,type_subs
tGPU = proc.time()-tGPU;tGPU
res1$par
tCPU;tGPU




res1 = STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,0,0,type_subs




#======================Gneiting======================#

rm(list=ls())


#######################R package######################
require(GeoModels)
require(MCMCpack)
require(STBEU)
####location sites ########################################
lambda=8
xx=seq(-lambda,lambda);
coords=as.matrix(expand.grid(xx,xx))   ###regular
nrow(coords)
# plot(coords)
#set.seed(15)                                                          ### not regular
#pp<-runifpoint(4*(lambda)^2, win=owin(c(-lambda,lambda),c(-lambda,lambda)))     ### not regular
#coords<-cbind(pp$x,pp$y)
### not regular
####temporal instants ########################################
times=seq(1,5,1)
##############################################################


type_dist=1                ### type of distance     1=euclidean 2=chordal  3=geodesic

maxdist=2                  ## compact support in weights function for pairwise liklihood
maxtime=2

model=2  #   1=double exponential       2 =gneiting

if (model == 2) {
  # gneiting model
  cov.model <-"gneiting"
  cc=2
  #####
```

```
  mean=0
  nugget=0
  scale_s<-1.5/3
  scale_t<-1.5/20
  sill=4
  power_s=1;power_t=1;sep=0.5
  param=list(nugget=nugget,mean=mean,scale_t=scale_t,scale_s=scale_s,sill=sill,
             power_s=power_s,power_t=power_t,sep=sep)
  fixed=list(nugget=0,power_s=power_s,power_t=power_t,sep=sep)
  fix=c(nugget = nugget, power_s = power_s, power_t = power_t,sep = sep)
}
##################################################################

set.seed(276)
##################################################################
# Simulation of the spatial Gaussian random field:
data <- GeoSim(coordx=coords,coordt=times,corrmodel=cov.model, param=param)$data
mm=mean(c(data))
vv=var(c(data))
###################################################################
###################################################################

###### Composite likelihood based on pairs estimation #############

start=list(mean=mm,scale_s=scale_s,scale_t=scale_t,sill=vv)
# Maximum composite-likelihood fitting of the random field:
fit <- GeoFit(data=data,coordx=coords,coordt=times,
                    corrmodel=cov.model,maxtime=maxtime,maxdist=maxdist,
                    likelihood="Marginal",type="Pairwise",
                    start=start,fixed=fixed,weighted=TRUE)
# Results:
print(fit$param)
unlist(start)
################################################
#   parameters for the subsampling ####
###############################################
coordx=coords[,1]
coordy=coords[,2]
LX=abs(range(coordx)[1]-range(coordx)[2])
LY=abs(range(coordy)[1]-range(coordy)[2])


lato_fin=3  #changing window size
lx=lato_fin          #lunghezza lato x quadrato subfinestra
ly=lato_fin          #lunghezza lato y quadrato subfinestra
winc=c(lx/sqrt(LX),ly/sqrt(LY))
winstp= 1  ###   1/lato_fin complete overlapping   1 "no" overlapping
###############################################
###############################################
theta=start                #starting value

weighted=0
### eucliden likelihood ################
type_subs=1    ### type of subsampling  1=in space    2= in time
tCPU = proc.time()
res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,0,0,type_subs,we
tCPU = proc.time()-tCPU;tCPU
```

```
res$par


res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,0,0,type_subs,we

### OpenCL eucliden likelihood ################
local <- c(1,1)
GPU <- 0
tGPU = proc.time()
res1=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,0,0,type_subs,w
tGPU = proc.time()-tGPU;tGPU
res1$par
tCPU;tGPU



res1=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,0,0,type_subs,w





#################################********  TIME  ***#######################################
#======================Double Exponential======================#
rm(list=ls())
########################R package######################
require(GeoModels)
require(MCMCpack)
require(STBEU)
####location sites ########################################
lambda=2
xx=seq(-lambda,lambda);
coords=as.matrix(expand.grid(xx,xx))   ###regular
nrow(coords)

#set.seed(15)                                                           ### not regular
#pp<-runifpoint(4*(lambda)^2, win=owin(c(-lambda,lambda),c(-lambda,lambda)))      ### not regular
#coords<-cbind(pp$x,pp$y)
### not regular
####temporal instants ########################################
times=seq(1,100,1)


type_dist=1                ### type of distance     1=euclidean 2=chordal  3=geodesic
maxdist=2
maxtime=2
#################################################################
#################################################################
model=1  #   1=double exponential       2 =gneiting

if (model == 1) {
  # exponential model
  cov.model <-"exp_exp"
  cc=1
  #####
  mean=0
  nugget=0
```

```
  scale_s<-1.5/3
  scale_t<-1.5/3
  sill=4
  param=list(nugget=nugget,mean=mean,scale_t=scale_t,scale_s=scale_s,sill=sill)
  fixed=list(nugget=0)
  fix=c(nugget = nugget)
}
###################################################################
###################################################################
# Simulation of the spatial Gaussian random field:
data <- GeoSim(coordx=coords,coordt=times,corrmodel=cov.model, param=param)$data
mm=mean(c(data))
vv=var(c(data))
####################################################################

###### Composite likelihood based on pairs estimation #############
start=list(mean=mm,scale_s=scale_s,scale_t=scale_t,sill=vv)
# Maximum composite-likelihood fitting of the random field:
fit <- GeoFit(data=data,coordx=coords,coordt=times,
                    corrmodel=cov.model,maxtime=maxtime,maxdist=maxdist,
                    likelihood="Marginal",type="Pairwise",
                    start=start,fixed=fixed,weighted=TRUE)


# Results:
print(fit$param)
unlist(start)
#################################################
#   parameters for the temporal subsampling ####
################################################

winc=5    ###  length of temporal window
winstp=1 ###   0.5 half overlapping  1 "no" overlapping
################################################

##################################################################
theta=start                #starting value
weighted=0
### eucliden likelihood ################
type_subs=2    ### type of subsampling  1=in space    2= in time
tCPU = proc.time()
res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,0,0,winc,winstp,type_subs,we
tCPU = proc.time()-tCPU;tCPU
eu_par=c(res$par[1],res$par[3],res$par[4],res$par[2])
names(eu_par)=names(fit$param)
print(eu_par)

res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,0,0,winc,winstp,type_subs,we

### OpenCL eucliden likelihood ################
local <- c(1,1)
GPU <- 0
tGPU = proc.time()
res1=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,0,0,winc,winstp,type_subs,w
tGPU = proc.time()-tGPU;tGPU
eu_par1=c(res1$par[1],res1$par[3],res1$par[4],res1$par[2])
names(eu_par1)=names(fit$param)
```

```
print(eu_par1)
tCPU;tGPU



res1=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,0,0,winc,winstp,type_subs,w


#======================Gneiting======================#

rm(list=ls())
########################R package######################
require(GeoModels)
require(MCMCpack)
require(STBEU)
####location sites ######################################
lambda=2
xx=seq(-lambda,lambda);
coords=as.matrix(expand.grid(xx,xx))   ###regular
nrow(coords)

#set.seed(15)                                                            ### not regular
#pp<-runifpoint(4*(lambda)^2, win=owin(c(-lambda,lambda),c(-lambda,lambda)))      ### not regular
#coords<-cbind(pp$x,pp$y)
### not regular
####temporal instants ######################################
times=seq(1,100,1)


type_dist=1                ### type of distance     1=euclidean 2=chordal  3=geodesic
maxdist=2
maxtime=2
##################################################################
model=2  #   1=double exponential       2 =gneiting

if (model == 2) {
  # gneiting model
  cov.model <-"gneiting"
  cc=2
  #####
  mean=0
  nugget=0
  scale_s<-1.5/3
  scale_t<-1.5/20
  sill=4
  power_s=1;power_t=1;sep=0.5
  param=list(nugget=nugget,mean=mean,scale_t=scale_t,scale_s=scale_s,sill=sill,
             power_s=power_s,power_t=power_t,sep=sep)
  fixed=list(nugget=0,power_s=power_s,power_t=power_t,sep=sep)
  fix=c(nugget = nugget, power_s = power_s, power_t = power_t,sep = sep)
}
##################################################################
##################################################################
# Simulation of the spatial Gaussian random field:
data <- GeoSim(coordx=coords,coordt=times,corrmodel=cov.model, param=param)$data
mm=mean(c(data))
```

```
vv=var(c(data))
####################################################################
####################################################################

###### Composite likelihood based on pairs estimation #############
start=list(mean=mm,scale_s=scale_s,scale_t=scale_t,sill=vv)
# Maximum composite-likelihood fitting of the random field:
fit <- GeoFit(data=data,coordx=coords,coordt=times,
                    corrmodel=cov.model,maxtime=maxtime,maxdist=maxdist,
                    likelihood="Marginal",type="Pairwise",
                    start=start,fixed=fixed,weighted=TRUE)


# Results:
print(fit$param)
unlist(start)
#################################################
#   parameters for the temporal subsampling ####
################################################

winc=4    ###  length of temporal window
winstp=1 ###   0.5 half overlapping  1 "no" overlapping
################################################
theta=start               #starting value
weighted=0
### eucliden likelihood ################
type_subs=2    ### type of subsampling  1=in space    2= in time
tCPU = proc.time()
res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,0,0,winc,winstp,type_subs,we
tCPU = proc.time()-tCPU;tCPU
eu_par=c(res$par[1],res$par[3],res$par[4],res$par[2])
names(eu_par)=names(fit$param)
print(eu_par)


res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,0,0,winc,winstp,type_subs,we

### OpenCL eucliden likelihood ################
local <- c(1,1)
GPU <- 0
tGPU = proc.time()
res1=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,0,0,winc,winstp,type_subs,w
tGPU = proc.time()-tGPU;tGPU
eu_par1=c(res1$par[1],res1$par[3],res1$par[4],res1$par[2])
names(eu_par1)=names(fit$param)
print(eu_par1)
tCPU;tGPU


res1=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,0,0,winc,winstp,type_subs,w
```

```
##################################********  SPACETIME  ***#######################################
#======================Double Exponential======================#
rm(list=ls())
########################R package######################
require(GeoModels)
require(MCMCpack)
require(STBEU)
####location sites ########################################
lambda=8
xx=seq(-lambda,lambda);
coords=as.matrix(expand.grid(xx,xx))   ###regular
nrow(coords)
# plot(coords)
#set.seed(15)                                                            ### not regular
#pp<-runifpoint(4*(lambda)^2, win=owin(c(-lambda,lambda),c(-lambda,lambda)))     ### not regular
#coords<-cbind(pp$x,pp$y)
### not regular
####temporal instants ########################################
times=seq(1,5,1)

type_dist=1                ### type of distance     1=euclidean 2=chordal  3=geodesic
maxdist=2
maxtime=2

##################################################################
model=1  #   1=double exponential       2 =gneiting

if (model == 1) {
  # exponential model
  cov.model <-"exp_exp"
  cc=1
  #####
  mean=0
  nugget=0
  scale_s<-1.5/3
  scale_t<-1.5/3
  sill=4
  param=list(nugget=nugget,mean=mean,scale_t=scale_t,scale_s=scale_s,sill=sill)
  fixed=list(nugget=0)
  fix=c(nugget = nugget)
}

##################################################################
set.seed(276)
##################################################################
# Simulation of the  Gaussian random field:
data <- GeoSim(coordx=coords,coordt=times,corrmodel=cov.model, param=param)$data
##################################################################
mm=mean(c(data))
vv=var(c(data))
###### Composite likelihood based on pairs estimation ############
start=list(mean=mm,scale_s=scale_s,scale_t=scale_t,sill=vv)
```

```
# Maximum composite-likelihood fitting of the random field:
fit <- GeoFit(data=data,coordx=coords,coordt=times,
                    corrmodel=cov.model,maxtime=maxtime,maxdist=maxdist,
                    likelihood="Marginal",type="Pairwise",
                    start=start,fixed=fixed,weighted=TRUE)

# Results:
print(fit$param)
unlist(start)
################################################
#   parameters for the subsampling ####
###############################################
coordx=coords[,1]
coordy=coords[,2]
LX=abs(range(coordx)[1]-range(coordx)[2])
LY=abs(range(coordy)[1]-range(coordy)[2])
lato_fin=3  #changing window size
lx=lato_fin          #lunghezza lato x quadrato subfinestra
ly=lato_fin          #lunghezza lato y quadrato subfinestra
winc=c(lx/sqrt(LX),ly/sqrt(LY))
winstp= 1###   1/lato_fin complete overlapping in space  1 "no" overlapping in space
###############################################
winc_t=4   ###  length of temporal window
winstp_t=1 ###   0.5 half overlapping  1 "no" overlapping
###############################################

theta=start                #starting value
weighted=0                                          #weigthed version  1=yes 0=no
### eucliden likelihood ################
type_subs=3   ### type of subsampling  1=in space    2= in time  3 =spacetime
tCPU = proc.time()
res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,winc_t,winstp_t,
tCPU = proc.time()-tCPU;tCPU
eu_par=c(res$par[1],res$par[3],res$par[4],res$par[2])
names(eu_par)=names(fit$param)
print(eu_par)

res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,winc_t,winstp_t,

### OpenCL eucliden likelihood ################
local <- c(1,1)
GPU <- 0
tGPU = proc.time()
res1=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,winc_t,winstp_t
tGPU = proc.time()-tGPU;tGPU
eu_par1=c(res1$par[1],res1$par[3],res1$par[4],res1$par[2])
names(eu_par1)=names(fit$param)
print(eu_par1)
tCPU;tGPU


res1=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,winc_t,winstp_t
```

```
#=======================Gneting=======================#

rm(list=ls())
########################R package######################
require(GeoModels)
require(MCMCpack)
require(STBEU)
####location sites ########################################
lambda=8
xx=seq(-lambda,lambda);
coords=as.matrix(expand.grid(xx,xx))   ###regular
nrow(coords)
# plot(coords)
#set.seed(15)                                                            ### not regular
#pp<-runifpoint(4*(lambda)^2, win=owin(c(-lambda,lambda),c(-lambda,lambda)))     ### not regular
#coords<-cbind(pp$x,pp$y)
### not regular
####temporal instants #######################################
times=seq(1,5,1)

type_dist=1                ### type of distance     1=euclidean 2=chordal  3=geodesic
maxdist=2
maxtime=2
##################################################################
model=2  #   1=double exponential       2 =gneiting


if (model == 2) {
  # gneiting model
  cov.model <-"gneiting"
  cc=2
  #####
  mean=0
  nugget=0
  scale_s<-1.5/3
  scale_t<-1.5/20
  sill=4
  power_s=1;power_t=1;sep=0.5
  param=list(nugget=nugget,mean=mean,scale_t=scale_t,scale_s=scale_s,sill=sill,
             power_s=power_s,power_t=power_t,sep=sep)
  fixed=list(nugget=0,power_s=power_s,power_t=power_t,sep=sep)
  fix=c(nugget = nugget, power_s = power_s, power_t = power_t,sep=sep)
}
##################################################################
set.seed(276)
##################################################################
# Simulation of the  Gaussian random field:
data <- GeoSim(coordx=coords,coordt=times,corrmodel=cov.model, param=param)$data
##################################################################
##################################################################
mm=mean(c(data))
vv=var(c(data))
###### Composite likelihood based on pairs estimation ############
start=list(mean=mm,scale_s=scale_s,scale_t=scale_t,sill=vv)

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

```
                    corrmodel=cov.model,maxtime=maxtime,maxdist=maxdist,
                    likelihood="Marginal",type="Pairwise",
                    start=start,fixed=fixed,weighted=TRUE)

# Results:
print(fit$param)
unlist(start)
################################################
#   parameters for the subsampling ####
###############################################
coordx=coords[,1]
coordy=coords[,2]
LX=abs(range(coordx)[1]-range(coordx)[2])
LY=abs(range(coordy)[1]-range(coordy)[2])
lato_fin=3  #changing window size
lx=lato_fin          #lunghezza lato x quadrato subfinestra
ly=lato_fin          #lunghezza lato y quadrato subfinestra
winc=c(lx/sqrt(LX),ly/sqrt(LY))
winstp= 1###   1/lato_fin complete overlapping in space  1 "no" overlapping in space
###############################################
winc_t=4   ###  length of temporal window
winstp_t=1 ###   0.5 half overlapping  1 "no" overlapping
###############################################

theta=start                #starting value
weighted=0                                          #weigthed version  1=yes 0=no
### eucliden likelihood ################
type_subs=3   ### type of subsampling  1=in space    2= in time  3 =spacetime
tCPU = proc.time()
res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,winc_t,winstp_t,
tCPU = proc.time()-tCPU;tCPU
eu_par=c(res$par[1],res$par[3],res$par[4],res$par[2])
names(eu_par)=names(fit$param)
print(eu_par)

res=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,winc_t,winstp_t,


### OpenCL eucliden likelihood ################
local <- c(1,1)
GPU <- 0
tGPU = proc.time()
res1=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,winc_t,winstp_t
tGPU = proc.time()-tGPU;tGPU
eu_par1=c(res1$par[1],res1$par[3],res1$par[4],res1$par[2])
names(eu_par1)=names(fit$param)
print(eu_par1)
tCPU;tGPU

res1=STBEUFit(theta,fixed = fix,coords,times,cc,data,type_dist,maxdist ,maxtime,winc,winstp,winc_t,winstp_t




#=======================Wen_time=======================#
###########################################################
require(GeoModels)
```

```
library(spam)
require(fields)
require(STBEU)
require(MCMCpack)

corrmodel="Wen_time";


####################################################
# Define the spatial-coordinates of the points:
set.seed(8)
dp <- 3
ns <- 7
x <- runif(ns)
y <- runif(ns)
coords=cbind(x,y)


n=nrow(coords)
# Define the temporal-coordinates:
times <- seq(0, 14.75, .3)
# times <- seq(0, 14.75, 4)
tt=length(times)
####################################################


####################################################
smooth_t=2 # k or kappa
scale_t=time_comp_supp=0.75     # compact supportt scale_t
scale_s=0.1
power2_t=3.5+smooth_t #nu
power_s=2
power2_s=2.5+2*smooth_t# tau

sep=1  ## 0 0.5  1
sill=1
mean=0
nugget=0
####################################################

####################################################
param <- list(nugget=nugget,mean=mean,sill = sill,
              power2_s=power2_s,power_s=power_s,power2_t=power2_t,scale_s=scale_s,
              scale_t=scale_t,sep=sep,smooth_t=smooth_t)
fixed=list(nugget=nugget,power2_s=power2_s,power2_t=power2_t,sep=sep,mean = mean,power_s=power_s)
start=list(scale_s=scale_s,scale_t=time_comp_supp,sill=sill,smooth_t=smooth_t)
####################################################
# Simulation of a spatial Gaussian RF:

set.seed(8519)

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


# Maximum composite-likelihood fitting of the random field:
tCPU0 <- proc.time()
```

```
fit <- GeoFit(data=data,coordx=coords,coordt=times,corrmodel=corrmodel,
              likelihood="Full",type="Standard",sparse=TRUE,
              start=start,fixed=fixed)
tCPU0 <- proc.time()-tCPU0
print(tCPU0)


model=3  #   1=double exponential       2 =gneiting 3 = WenTime

# gneiting model
cov.model <-"Wen_time"
cc=3
#####

################################################
#   parameters for the temporal subsampling ####
###############################################

winc=2    ###  length of temporal window
winstp=1 ###   0.5 half overlapping  1 "no" overlapping
###############################################

#################################################################
theta=start                #starting value
weighted=0
type_dist=1                ### type of distance     1=euclidean 2=chordal  3=geodesic
maxdist=.4
maxtime=2
### eucliden likelihood ################
type_subs=2    ### type of subsampling  1=in space    2= in time
tCPU = proc.time()
res=STBEUFit(theta,fixed = unlist(fixed),coords,times,cc,data,type_dist,
             maxdist ,maxtime,0,0,winc,winstp,type_subs,weighted)
tCPU = proc.time()-tCPU
print(tCPU)


tGPU = proc.time()
res1=STBEUFit(theta,unlist(fixed),coords,times,cc,data,type_dist,
              maxdist ,maxtime,0,0,winc,
              winstp,type_subs,weighted,
              GPU = 0, local = c(1,1))
tGPU = proc.time()-tGPU
print(tGPU)
```

---

varestfun *Checking Correlation Model*

Description

The procedure controls if the correlation model inserted is correct.

Usage

```
varestfun(theta,fixed=NULL,coordx,coordy,ncoords,times,ntime,cc,datos,type_dist=1,maxdist=NULL,m
                      winc_s=NULL,winstp_s=NULL,winc_t=NULL,winstp_t=NULL,
                      weighted=FALSE,type_sub=NULL,local=c(1,1),GPU=NULL)
```

Arguments

theta	Numeric; a vector which has fixed parameters info.
fixed	Numeric; a vector which has to be estimated parameters info.
coordx	Numeric; it indicates the correlation model
coordy	Numeric; it indicates the correlation model
ncoords	Numeric; it indicates the correlation model
times	Numeric; it indicates the correlation model
ntime	Numeric; it indicates the correlation model
cc	Numeric; it indicates the correlation model
datos	Numeric; it indicates the correlation model
type_dist	Numeric; it indicates the correlation model
maxdist	Numeric; it indicates the correlation model
maxtime	Numeric; it indicates the correlation model
winc_s	Numeric; it indicates the correlation model
winstp_s	Numeric; it indicates the correlation model
winc_t	Numeric; it indicates the correlation model
winstp_t	Numeric; it indicates the correlation model
weighted	Numeric; it indicates the correlation model
type_sub	Numeric; it indicates the correlation model
local	Numeric; it indicates the correlation model
GPU	Numeric; it indicates the correlation model

Value

Return the theta parameters with correct names.

Author(s)

Moreno Bevilacqua, <moreno.bevilacqua@uv.cl>,https://sites.google.com/a/uv.cl/moreno-bevilacqua/home, Víctor Morales Oñate, <victor.morales@uv.cl>, https://sites.google.com/site/moralesonatevictor/

`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` *Irish Daily Wind Speeds Coords*

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.

Index