Package 'HiDimMaxStable'

October 8, 2014

Version 0.1

Date 2014-10-07

Encoding UTF-8

Title Inference on High Dimensional Max-Stable Distributions

Depends R (>= 2.2.0), methods, copula

Imports mnormt, partitions, maxLik, mnormpow, VGAM

Suggests rgl, snowfall, MASS, SpatialExtremes

Description Inference of high dimensional max-stable distributions, from the paper ``Likelihood based inference for high-dimensional extreme value distributions", by A. Bienvenüe and C. Robert, arXiv:1403.0065 [stat.AP].

License GPL (>= 2)

URL http://arxiv.org/abs/1403.0065

Author Alexis Bienvenüe [aut, cre],Christian Robert [aut]

Maintainer Alexis Bienvenüe <alexis.bienvenue@univ-lyon1.fr>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-10-08 21:57:23

R topics documented:

`build.clusters.spatial` *Builds clusters with a given maximum size using a k-means clustering.*

Description

Builds clusters from the spatial locations of sites using a k-means clustering, to get a partition (of the sites) whose block sizes are at most 5 so that the partition-composite likelihood for observations of a max-stable process at the sites can be computed in a moderate time.

Usage

```
build.clusters.spatial(xy,max.size=5,plot=FALSE)
```

Arguments

`xy`	a matrix giving the coordinates of each location. Each row corresponds to one location.
`max.size`	an integer giving the maximum size of the blocks of the returned partition.
`plot`	Whether you also whant a plot or not.

Value

the return value is a vector of integers giving the index of the set for each location.

Examples

```
n.site<-10
xy<-matrix(runif(2 * n.site, 0, 0.5), ncol = 2)

param<-c(0.5,1.5)
n.obs<-20
library(SpatialExtremes)
data<-t(rmaxstab(n.obs, xy, "whitmat",
                 nugget = 0, range = param[1], smooth = param[2]))
cl<-build.clusters.spatial(xy)

d<-maxstable.l.clusters(data,clusters=cl,
                        params=param,
                        category="normal",
                        spatial=list(sites=xy,family=spatialWhittleMatern))
```

`dens.grid` *Computes the likelihood function on a grid of parameters*

Description

The `dens.grid.*` function family is used to compute the likelihood at several points on a grid. `*` must be one of the following: "`maxstable`", "`excess`" or "`simultoccur`".

Usage

```
dens.grid.maxstable(...)
dens.grid.excess(...)
dens.grid.simultoccur(...)
```

Arguments

`...`	see details.

Details

`dens.grid` specific arguments are the following:

`data` a matrix representing the data.

`params` a vector giving the arguments to be passed to the likelihood function; `NA` indicates that the coordinate has to be replaced by multiple values on a grid (there must be two `NA`'s in the vector).

`seqx,seqy` vectors giving the values to be used to compute the likelihood for the two coordinates that are set to `NA` in the `params` vector.

`ln` logical. If `TRUE` log-density is computed.

`parallel` logical. If `TRUE` a parallel computation of the log-likelihood function is performed, using the `snowfall` package (`sfInit` must be called before).

Other arguments are passed to the likelihood function.

Value

returns a list `l` including parameters used in the call (`l$params`, `l$seqx`, `l$seqy`), and a matrix for the computed values of the log-likelihood: `l$dens`.

See Also

`maxstable.l.clusters`, `excess.l`, `simultoccur.l` for the likelihood functions; `plot3d.densgrid` for a 3D visualisation of the computed values of the log-likelihood.

Examples

```
# Log density of the Max-stable distribution of the Schlather process
# with the Whittle Matern correlation function
n.site<-3
n.obs<-2
xy<-matrix(runif(2*n.site,0,2),ncol=2)
param<-c(0.5,1.5)
library(SpatialExtremes)
data<-t(rmaxstab(n.obs, xy, "whitmat",
    nugget = 0, range = param[1], smooth = param[2]))
cl<-build.clusters.spatial(xy)
d<-dens.grid.maxstable(data,c(NA,NA),seq(0.1,1,length=5),seq(1,2,length=5),
    category="normal",
    spatial=list(sites=xy,family=spatialWhittleMatern),
    parallel=FALSE,
    clusters=cl)
plot3d.densgrid(d)


#  Log density of the Max-stable distribution of the Brown Resnick process
n.site<-100
xy<-matrix(runif(2 * n.site, 0, 2), ncol = 2)
param<-c(0.5,1)
n.obs<-20
library(SpatialExtremes)
data<-t(rmaxstab(n.obs, xy, cov.mod = "brown", range = param[1], smooth = param[2]))
cl<-build.clusters.spatial(xy)
library(snowfall)
sfInit(parallel=TRUE,cpus=4)
sfLibrary(HiDimMaxStable)
d<-dens.grid.maxstable(data,c(NA,NA),seq(0.1,1,length=5),seq(0.1,1.8,length=5),
    category="lnormal",
    spatial=list(sites=xy,family=spatialPower),
    parallel=TRUE,
    clusters=cl)
sfStop()
plot3d.densgrid(d)

#   Log density of the distribution of the vector of excesses for an
# homogeneous clustered max-stable distribution
#
raw.data<-rCMS(copulas=c(copClayton,copGumbel),
```

```
    margins=c(marginLnorm,marginFrechet),
    classes=c(rep(1,50),rep(2,50)),
    params=c(0.5,1,1.5,1.7),n=1000)
data<-excess.censor(raw.data)
library(snowfall)
sfInit(parallel=TRUE,cpus=48)
sfLibrary(HiDimMaxStable)
d<-dens.grid.excess(data,c(NA,1,NA,1.7),
    seq(0.1,1,length=5),seq(1,2,length=5),
    category="copula",
    copulas=c(copClayton,copGumbel),
    margins=c(marginLnorm,marginFrechet),
            parallel=TRUE,
    classes=c(rep(1,50),rep(2,50)))
sfStop()
plot3d.densgrid(d)

# Log density of the distribution of the componentwise maxima
# with occurences for an homogeneous clustered max-stable distribution
raw.data<-rCMS(copulas=c(copClayton,copGumbel),
    margins=c(marginLnorm,marginFrechet),
    classes=c(rep(1,50),rep(2,50)),
    params=c(0.5,1,1.5,1.7),n=1000)
data<-maxblocks(raw.data,n.blocks=20)
library(snowfall)
sfInit(parallel=TRUE,cpus=4)
sfLibrary(HiDimMaxStable)
d<-dens.grid.simultoccur(data$normalized.max,occur=data$classes.max,
    c(NA,1,NA,1.7),
    seq(0.1,1,length=5),seq(1,2,length=5),
    category="copula",
    copulas=c(copClayton,copGumbel),
    margins=c(marginLnorm,marginFrechet),
    classes=c(rep(1,50),rep(2,50)),
    parallel=TRUE)
sfStop()
plot3d.densgrid(d)
```

`excess.censor` *Transforms data to normalized exceedances with censoring*

Description

First transforms empirical marginal distributions to unit Pareto by using order statistics, second scales to 1/t, third censor values smaller than one, and then drops all vectors with no value greater than one.

Usage

```
excess.censor(z,t=10)
```

Arguments

z	a matrix representing the data.
t	a threshold value.

Examples

```
raw.data<-rCMS(copulas=c(copClayton,copGumbel),
               margins=c(marginLnorm,marginFrechet),
               classes=c(rep(1,10),rep(2,10)),
               params=c(0.5,1,1.5,1.7),n=1000)
data<-excess.censor(raw.data)
```

---

excess.l *Likelihood for vectors of exceedance with censored components*

Description

Computes the likelihood for observations of vectors of exceedances that belong to the maximum domain of attraction of a multivariate max-stable distribution whose spectral random vector is Gaussian, Log-normal or has a clustered copula distribution.

Usage

```
excess.l(data,ln=FALSE,...)
```

Arguments

data	a matrix representing the data. Each column corresponds to one observation of a vector of exceedance with censored components. Note that all components must be larger or equal to one.
ln	logical. If TRUE log-density is computed.
...	further arguments to be passed to mubz.* function (where * stands for the category of the model). In particular, category is a character string indicating the model to be used: "normal", "lnormal" or "copula", and params gives the values of the parameters for which the likelihood is computed.

See Also

mubz.normal,mubz.lnormal, mubz.copula.

Examples

```
raw.data<-rCMS(copulas=c(copClayton,copGumbel),
               margins=c(marginLnorm,marginFrechet),
               classes=c(rep(1,4),rep(2,4)),
               params=c(0.5,1,1.5,1.7),n=50)
data<-excess.censor(raw.data)

d<-excess.l(data,params=c(0.5,1,1.5,1.7),
            category="copula",
            copulas=c(copClayton,copGumbel),
            margins=c(marginLnorm,marginFrechet),
            classes=c(rep(1,4),rep(2,4)))
```

---

margin *Margin distributions*

Description

Margin distributions

Usage

```
marginExp
marginFrechet
marginGPD
marginGamma
marginLnorm
marginUnif
marginWeibull
```

Value

A "margin" object.

See Also

mubz.copula for a description of the distributions and their parameters.

margin-class *margin class*

Description

Class for margins types

Objects

marginExp exponential distribution

marginFrechet Frechet distribution

marginGPD GPD distribution

marginGamma Gamma distribution

marginLnorm log-normal distribution

marginUnif uniform distribution

marginWeibull Weibull distribution

See Also

`mubz.copula` for a description of the distributions and their parameters.

maxblocks *Computes the normalized componentwise maxima with their occurrences for several blocks*

Description

Splits up the data into blocks and computes the normalized componentwise maxima with their occurrences for each block

Usage

```
maxblocks(y,n.blocks=50)
```

Arguments

`y`	a matrix representing the data.
`n.blocks`	the numbers of blocks.

Value

returns a list `l` including:

`l$normalized.max`	the normalized maxima (divided by the number of observation in a block
`l$classes.max`	the partitions that gives which componentwise maxima occur simultaneously

Examples

```
raw.data<-rCMS(copulas=c(copClayton,copGumbel),
               margins=c(marginLnorm,marginFrechet),
               classes=c(rep(1,10),rep(2,10)),
               params=c(0.5,1,1.5,1.7),n=1000)
data<-maxblocks(raw.data,n.blocks=20)
```

`maxgrid` *Identifies the coordinates of the maximum on a grid*

Description

Finds the maximum likelihood point from a maxgrid object.

Usage

```
maxgrid(dg)
```

Arguments

`dg`	an object returned by a `dens.grid` call.

Value

returns a list `l` including parameters used in the `dens.grid` call of `l$params`, the coordinates of the maximum on the grid `l$xy`, and the value of the maximum `l$value`.

See Also

`plot3d.densgrid` for a 3D visualisation of the computed values on the grid.

Examples

```
n.site<-3
xy<-matrix(runif(2 * n.site, 0, 0.5), ncol = 2)
param<-c(0.5,1.5)
n.obs<-2
library(SpatialExtremes)
data<-t(rmaxstab(n.obs, xy, "whitmat",
                 nugget = 0, range = param[1], smooth = param[2]))
d<-dens.grid.maxstable(data,c(NA,NA),seq(0.1,1,length=5),seq(1,2,length=5),
                       category="normal",
                       spatial=list(sites=xy,family=spatialWhittleMatern))
maxgrid(d)$xy
```

maxlik *Maximum likelihood estimation*

Description

The maxlik.* function family computes the maximum likelihood from data of a max-stable distribution or in the maximum domain of attraction of a max-stable distribution. * must be one of the following: "maxstable", "excess" or "simultoccur".

Usage

```
maxlik.maxstable(...)
maxlik.excess(...)
maxlik.simultoccur(...)
```

Arguments

...	all the arguments to be passed to the likelihood function including:
	data a matrix representing the data
	params a vector giving the arguments to be passed to the likelihood function. NA indicates the coordinates that will be used for the optimization of the likelihood function.
	start a vector giving the initial values of the parameters

Value

returns a list l including the following components: l$estimate giving the estimated parameter values, l$message giving a short message describing if the convergence is successfull, l$iterations giving the number of iterations...

See Also

maxLik, maxstable.l.clusters, excess.l, simultoccur.l.

Examples

```
# ML estimation for a sample from a Schlather max-stable distribution
# (Use larger values for n.site and n.obs to get good results!)
n.site<-3
n.obs<-2
xy<-matrix(runif(2 * n.site, 0, 0.5), ncol = 2)
param<-c(0.5,1.5)
library(SpatialExtremes)
data<-t(rmaxstab(n.obs, xy, "whitmat",
                 nugget = 0, range = param[1], smooth = param[2]))
ml<-maxlik.maxstable(data,params=c(NA,NA),start=c(1,1),
                     category="normal",
                     spatial=list(sites=xy,family=spatialWhittleMatern),
```

```
                      iterlim=4)


# ML estimation for a sample in the max domain of attraction
# of from an homogeneous clustered max-stable distribution

# A) using excess

raw.data<-rCMS(copulas=c(copClayton,copGumbel),
               margins=c(marginLnorm,marginFrechet),
               classes=c(rep(1,10),rep(2,10)),
               params=c(0.5,1,1.5,1.7),n=1000)
data<-excess.censor(raw.data)
d<-dens.grid.excess(data,c(NA,1,NA,1.7),
                    seq(0.1,1,length=10),seq(1,2,length=10),
                    category="copula",
                    copulas=c(copClayton,copGumbel),
                    margins=c(marginLnorm,marginFrechet),
                    classes=c(rep(1,10),rep(2,10)))
plot3d.densgrid(d)
ml<-maxlik.excess(data,
                  params=c(NA,NA,NA,NA),start=c(0.5,1,1.5,1.7),
                  category="copula",
                  copulas=c(copClayton,copGumbel),
                  margins=c(marginLnorm,marginFrechet),
                  classes=c(rep(1,10),rep(2,10)))

# B) using block maxima with occurences

raw.data<-rCMS(copulas=c(copClayton,copGumbel),
               margins=c(marginLnorm,marginFrechet),
               classes=c(rep(1,10),rep(2,10)),
               params=c(0.5,1,1.5,1.7),n=1000)
data<-maxblocks(raw.data,n.blocks=20)
d<-dens.grid.simultoccur(data$normalized.max,occur=data$classes.max,
                         c(NA,1,NA,1.7),
                         seq(0.1,1,length=10),seq(1,2,length=10),
                         category="copula",
                         copulas=c(copClayton,copGumbel),
                         margins=c(marginLnorm,marginFrechet),
                         classes=c(rep(1,10),rep(2,10)))
plot3d.densgrid(d)
ml<-maxlik.simultoccur(data$normalized.max,
                       params=c(NA,NA,NA,NA),start=c(0.5,1,1.5,1.7),
                       occur=data$classes.max,
                       category="copula",
                       copulas=c(copClayton,copGumbel),
                       margins=c(marginLnorm,marginFrechet),
                       classes=c(rep(1,10),rep(2,10)))
```

`maxstable.l.clusters` *Partition-composite likelihood for multivariate max-stable distributions*

Description

Computes the partition-composite likelihood for observations sampled from a multivariate max-stable distribution whose spectral random vector is Gaussian, Log-normal or has a clustered copula distribution.

Usage

```
maxstable.l.clusters(data,clusters=rep(1,dim(data)[1]),ln=FALSE,spatial=NULL,...)
```

Arguments

`data`	a matrix representing the data. Each column corresponds to one observation.
`clusters`	a vector of integers that gives the partition that is used to compute the partition-composite likelihood. Blocks of the partition should be of size smaller or equal to 7 to avoid a too long computing time. `clusters=rep(1,dim(data)[1])` must be used to get the full likelihood.
`ln`	logical. If `TRUE` log-density is computed.
`spatial`	argument passed to the `mubz.*` function (where `*` stands for the `category` of the model).
`...`	further arguments to be passed to `mubz.*` function (where `*` stands for the `category` of the model). In particular, `category` is a character string indicating the model to be used: `"normal"`, `"lnormal"` or `"copula"`, and `params` gives the values of the parameters for which the likelihood is computed.

See Also

`mubz.normal`,`mubz.lnormal`, `mubz.copula`.

Examples

```
n.site<-5
xy<-matrix(runif(2 * n.site, 0, 0.5), ncol = 2)

param<-c(0.5,1.5)
n.obs<-2
library(SpatialExtremes)
data<-t(rmaxstab(n.obs, xy, "whitmat",
                 nugget = 0, range = param[1], smooth = param[2]))

d<-maxstable.l.clusters(data,clusters=c(1,1,1,2,2),
                        params=param,
                        category="normal",
                        spatial=list(sites=xy,family=spatialWhittleMatern))
```

`mubz.copula` $\mu(B, z)$ *for the copula model*

Description

Computes $\mu(B, z)$ for the copula model.

Usage

```
mubz.copula(details=FALSE,...)
```

Arguments

`details`	get more details in the return value?
`...`	See details section.

Details

`mubz.copula` uses `integrate` to compute the value of $\mu(B, z)$. If `details` is `TRUE`, `mubz.copula` returns the `integrate` return value. If `details` is `FALSE`, `mubz.copula` returns the value only.

The types of distributions (with scalar parameter p) in the 'margin' class are the following: `marginUnif` is for the Uniform distribution with support $[1 - p, 1 + p]$; `marginLnorm` is for the Lognormal distribution whose the standard deviation of the normal distribution is equal to p; `marginWeibull` is for the Weibull distribution with shape parameter equal to p; `marginFrechet` is for the Frechet distribution with shape parameter equal to p; `marginGamma` is for the Gamma distribution with shape parameter equal to p; `marginGPD` is for the GPD distribution with shape parameter equal to p.

The types of Archimedean copulas are those implemented in the package copula: `copAMH`, `copClayton`, `copFrank`, `copGumbel`, `copJoe`.

Parameters of `mubz.copula`:

b a vector of TRUE or FALSE, of length `d` where `d=length(z)`, TRUE indicating the coordinates of B

z a vector of positive constants

params a vector of length `2*max(classes)`, giving successively the parameters of the archimedean copula and of the marginal distribution for each class

cop a vector of 'acopula' objects from package `copula` of length `max(classes)` giving the archimedean copulas for each class

margins a vector of objects of 'margin' class of length `max(classes)` giving the marginal distributions for each class

classes a vector of integers indicating for each coordinate of `z` the number of its class (from 1 to `max(classes)`)

See Also

`mubz.normal`, `mubz.lnormal`

Examples

```
# In this example, we compute mu(B,z) for the independent copula
# and Frechet margins.
z<-c(2,3)
kappa<-2
mu<-numeric(2)
mu[1]<-mubz.copula(b=c(TRUE,FALSE),z=z,params=c(1,kappa),
    cop=c(copGumbel),margins=c(marginFrechet),classes=c(1,1))
mu[2]<-mubz.copula(b=c(FALSE,TRUE),z=z,params=c(1,kappa),
    cop=c(copGumbel),margins=c(marginFrechet),classes=c(1,1))
# Compares mu({1},z)+mu({2},z) with the exact value:
t(mu) %*% z
(sum(1/z^kappa))^(1/kappa)

# For independent components with different distributions,
# one can use any "one-dimensional" copula:
mubz.copula(b=c(TRUE,FALSE),z=z,
    params=c(1,2,1,3),cop=c(copGumbel,copGumbel),
    margins=c(marginFrechet,marginGamma),classes=c(1,2))
```

---

mubz.lnormal — $\mu(B,z)$ *for the Log-normal model*

Description

Computes $\mu(B,z)$ for the Log-normal model.

Usage

```
mubz.lnormal(b,z,params=NULL,spatial=NULL,
    cov.matrix=spatial.cor.matrix(params,spatial),
    details=FALSE)
```

Arguments

b	a vector of TRUE or FALSE, of length d where d=length(z), TRUE indicating the coordinates of B
z	a vector of positive constants
params	a vector of length 2 if spatial=spatialPower (Brown-Resnick max-stable process): the first component is for the range parameter, the second component is for the smoothness parameter. a vector of length 3 for the other spatial models (Geometric Gaussian max-stable process): the first component is for the range parameter, the second component is for the smoothness parameter and the third component is for the scale parameter

spatial	the correlation model given as a list: spatial$sites is a matrix that gives the coordinates of each location. Each row corresponds to one location. spatial$family is a object from the spatial class that gives the spatial model. This must be one of the following family: - Brown-Resnick max-stable process: spatialPower - Geometric Gaussian max-stable process: spatialWhittleMatern for the Whittle Matern correlation model, spatialCauchy for the Cauchy correlation model, spatialPowerExp for the Power exponential model, spatialBessel for the Bessel correlation model
cov.matrix	a covariance matrix if spatial=NULL
details	get more details in the return value?

Details

mubz.lnormal uses pmnorm to compute the value of $\mu(B, z)$. If the dimension of z is too large (cannot exceed 20), the computation may fail.

See Also

mubz.normal, mubz.copula

Examples

```
# In this example, we compute mu(B,z) for Brown Resnick spatial model
# from 10 sites uniformly distributed on the square [0,2]x[0,2]
mubz.lnormal(b=c(TRUE,TRUE,FALSE,FALSE,TRUE,FALSE,FALSE,FALSE,FALSE,TRUE),
    z=rep(1,10),params=c(1,1),
    spatial=list(sites=matrix(2*runif(20),ncol=2),family=spatialPower))
```

---

mubz.normal $\mu(B, z)$ *for the Gaussian model*

Description

Computes $\mu(B, z)$ for the Gaussian model.

Usage

```
mubz.normal(b,z,params=NULL,spatial=NULL,
    cor.matrix=spatial.cor.matrix(c(params,1),spatial),
    details=FALSE)
```

Arguments

b	a vector of TRUE or FALSE, of length d where d=length(z), TRUE indicating the coordinates of B
z	a vector of positive constants
params	a vector of length 2 that must be informed if spatial is given; the first component is for the range parameter and the second component is for the smoothness parameter
spatial	the correlation model given as a list: spatial$sites is a matrix that gives the coordinates of each location. Each row corresponds to one location. spatial$family is a object from the spatial class that gives the spatial model. This must be one of the following family: spatialWhittleMatern for the Whittle Matern correlation model, spatialCauchy for the Cauchy correlation model, spatialPowerExp for the Power exponential model, spatialBessel for the Bessel correlation model
cor.matrix	a correlation matrix if spatial=NULL
details	get more details in the return value?

Details

mubz.normal uses mnormpow to compute the value of $\mu(B, z)$. If the dimension of z is too large (cannot exceed 20), the computation may fail.

See Also

mubz.lnormal, mubz.copula

Examples

```
# In this example, we compute mu(B,z) for Whittle Matern spatial model
# from 10 sites uniformly distributed on the square [0,2]x[0,2]
mubz.normal(b=c(TRUE,TRUE,FALSE,FALSE,TRUE,FALSE,FALSE,FALSE,FALSE,TRUE),
    z=rep(1,10),params=c(1,2),
    spatial=list(sites=matrix(2*runif(20),ncol=2),family=spatialWhittleMatern))
```

---

plot3d.densgrid *3D visualisation of the computed values of the likelihood function on a grid.*

Description

3D vizualisation for maximum likelihood computed by densgrid.

Usage

```
plot3d.densgrid(dg,...)
```

Arguments

dg	an object returned by a dens.grid call.
...	further arguments passed to persp3d.

See Also

maxstable.l.clusters, excess.l, simultoccur.l for the likelihood functions; dens.grid.maxstable, dens.grid.excess and dens.grid.simultoccur for some examples.

---

rCMS — *Simulation of vectors in the maximum domain of attraction of an homogeneous clustered max-stable distribution*

Description

Generates realisations of vectors in the maximum domain of attraction of an homogeneous clustered max-stable distribution.

Usage

```
rCMS(copulas,margins,classes,params,n=100)
```

Arguments

copulas	a vector of 'acopula' objects from package copula of length max(classes) giving the archimedean copulas for each class
margins	a vector of objects of 'margin' class of length max(classes) giving the marginal distributions for each class
classes	a vector of integers indicating for each component the number of its class (from 1 to max(classes))
params	a vector of length 2*max(classes), giving successively the parameters of the archimedean copula and of the marginal distribution for each class
n	an integer giving the number of observations

Details

a vector is generated as the product of two independent random variables: a unit Pareto random variable and a random vector whose components are independent sub-vectors with distributions (copula,margin).

Examples

```
raw.data<-rCMS(copulas=c(copClayton,copGumbel),
               margins=c(marginLnorm,marginFrechet),
               classes=c(rep(1,10),rep(2,10)),
               params=c(0.5,1,1.5,1.7),n=1000)
data<-excess.censor(raw.data)
```

`rSchlatherExcess` *Simulation of vectors in the maximum domain of attraction of a spatial Schlather max-stable distribution*

Description

Generates realisations of vectors in the maximum domain of attraction of a spatial Schlather max-stable distribution.

Usage

```
rSchlatherExcess(n=500,spatial,params)
```

Arguments

`n`	an integer giving the number of observations
`spatial`	the correlation model given as a list: `spatial$sites` is a matrix that gives the coordinates of each location. Each row corresponds to one location. `spatial$family` is an object from the `spatial` class that gives the spatial model. This must be one of the following family: `spatialWhittleMatern` for the Whittle Matern correlation model, `spatialCauchy` for the Cauchy correlation model, `spatialPowerExp` for the Power exponential model, `spatialBessel` for the Bessel correlation model
`params`	A vector of length 2 that must be informed if `spatial` is given; the first component is for the range parameter and the second component is for the smoothness parameter.

Details

a vector is generated as the product of two independent random variables: a unit Pareto random variable and a Gaussian random vector whose covariance matrix is derived from the `spatial` model. The package `MASS` must be loaded to generate Gaussian random vectors

Examples

```
library(MASS)
rSchlatherExcess(n=500,
  spatial=list(sites=matrix(2*runif(20),ncol=2),family=spatialWhittleMatern),
  params=c(1,2))
```

`select.mean` *Selects vectors for which the mean of the components is larger than a threshold*

Description

For a matrix, first computes the mean of each column, second drop all column whose mean is less than a threshold, third returns the radial-angular decompositions

Usage

```
select.mean(data,t)
```

Arguments

`data`	a matrix representing the data.
`t`	a threshold value.

Value

returns a list `l` including:

`l$selected`	the normalized selected vectors (divided by the threshold)
`l$u`	their angular part
`l$gamma`	their radial part

See Also

`excess.censor`

Examples

```
raw.data<-rCMS(copulas=c(copClayton,copGumbel),
               margins=c(marginLnorm,marginFrechet),
               classes=c(rep(1,10),rep(2,10)),
               params=c(0.5,1,1.5,1.7),n=1000)
data<-select.mean(raw.data,t=10)
```

simultoccur.l *Likelihood for vectors of componentwise maxima with additional information on maxima occurences*

Description

Computes the likelihood for observations of vectors of componentwise maxima with additional information on maxima occurences. The data that are used to compute componentwise maxima must belong to the maximum domain of attraction of a multivariate max-stable distribution whose spectral random vector is Gaussian, Log-normal or has a clustered copula distribution.

Usage

```
simultoccur.l(data,occur,ln=FALSE,...)
```

Arguments

data	a matrix representing the data. Each column corresponds to one observation of a vector of componentwise maxima.
occur	a matrix representing the data. Each column corresponds to one observation of a vector that gives which componentwise maxima occur simultaneously.
ln	logical. If TRUE log-density is computed.
...	further arguments to be passed to mubz.* function (where * stands for the category of the model). In particular, category is a character string indicating the model to be used: "normal", "lnormal" or "copula", and params gives the values of the parameters for which the likelihood is computed.

See Also

mubz.normal,mubz.lnormal, mubz.copula.

Examples

```
raw.data<-rCMS(copulas=c(copClayton,copGumbel),
               margins=c(marginLnorm,marginFrechet),
               classes=c(rep(1,3),rep(2,3)),
               params=c(0.5,1,1.5,1.7),n=10)

data<-maxblocks(raw.data,n.blocks=3)

d<-simultoccur.l(data$normalized.max,occur=data$classes.max,
            params=c(0.5,1,1.5,1.7),
            category="copula",
            copulas=c(copClayton,copGumbel),
            margins=c(marginLnorm,marginFrechet),
            classes=c(rep(1,3),rep(2,3)))
```

spatial *Spatial models*

Description

Spatial models

Usage

```
spatialWhittleMatern
spatialCauchy
spatialPower
spatialPowerexp
spatialBessel
```

Value

A "spatial" object.

spatial-class *spatial class*

Description

Class for spatial models

Objects

spatialWhittleMatern for the Whittle Matern correlation model

spatialCauchy for the Cauchy correlation model

spatialPower for the power model

spatialPowerexp for the power exponential model

spatialBessel for the Bessel correlation model

Index