Package 'ABCExtremes'

August 11, 2014

Type Package

Title ABC Extremes

Version 1.0

Date 2013-05-09

Author Rob Erhardt

Maintainer <erhardrj@wfu.edu>

Description This package contains code to 1.) construct the summary statistic needed to facilitate approximate Bayesian computing for spatial extremes, and then 2.) run the ABC-Rejection method to produce a set of draws from the ABC posterior. The intention is for the user to directly modify the command >abc.rej() to suit his or her purposes. In general, ABC fitting of max-stable processes is very computationally expensive, and requires advanced programming skills with research computing.

Depends SpatialExtremes, combinat

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2013-05-15 10:45:53

R topics documented:

ABCExtremes-package *ABC Extremes*

Description

This package contains code to 1.) construct the summary statistic needed to facilitate approximate Bayesian computing for spatial extremes, and then 2.) run the ABC-Rejection method to produce a set of draws from the ABC posterior. The intention is for the user to directly modify the command >abc.rej() to suit his or her purposes. In general ABC fitting of max-stable processes is very computationally expensive, and requires advanced programming skills with research computing.

Details

Package:	ABCExtremes
Type:	Package
Version:	1.0
Date:	2013-05-14
License:	What license is it under?

Author(s)

Rob Erhardt Maintainer: <erhardrj@wfu.edu>

References

Erhardt, R., Smith, R. (2012). Approximate Bayesian Computing for Spatial Extremes. Computational Statistics and Data Analysis Vol. 56:6, 1468-1481.

See Also

See also: R packages SpatialExtremes, RandomFields

abc.rej *abc.rej*

Description

This function facilitates the ABC rejection sampling. The intention is for the user to directly modify this code to suit his or her purposes. As presently written, this function will use a uniform(0,10) prior for the range parameter, the weighted summary statistic from the Erhardt and Smith 2012 CSDA paper, and all draws will be independent.

Usage

```
abc.rej(sum.stat, assign, sims, coord, yr, cov=c("whitmat", "powexp","cauchy"))
```

Arguments

sum.stat	summary statistic as a vector, taken from >summary.stat()
assign	The group assignments of each individual triplet. Output from the >cluster() function
sims	The number of abc rejection draws.
coord	The 2 dimensional coordinates of the observed data. Rows are individual locations, columns are coordinates.
yr	The number of independent replicates of the observed data (blocks, years, etc.)
cov	The covriance model underlying the max-stable process. Can be "gauss", "whitmat", "powexp", etc. (see SpatialExtremes package for more details)

Examples

```
## Simulate 100 years of a max-stable process at 25 randomly placed locations on a 10 by 10 square.
## Notice that the true parameter is (range,smooth) = (1,3)
library(SpatialExtremes)
n=10
a=runif(n,0,10)
b=runif(n,0,10)
coord=cbind(a,b)
yr=100
data=rmaxstab(yr, coord, cov.mod = "whitmat", nugget=1, range=3, smooth=1, grid=FALSE)
plot(coord)

## cluster the triplets into k groups.  The command will write the number of clusters that change
## assignment each iteration, and the method stops when there are 0 changes.

k=5
assign=cluster(coord, k, "kmeans")

## Form an summary statistic for the observed data

sum.stat=summarize.stat(data, assign, frechet="F")

## Run ABC rejection sampling

out=abc.rej(sum.stat, assign, sims=10, coord, yr=100, cov="whitmat")
out

## To run more simulations, consider running this command across many nodes of a research computing
## cluster.  To run ABC for more than one parameter, with different priors, different summary
## statistics, or any other changes, directly modify the function >abc.rej(), shown below.


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

```
## Here is the function >abc.rej() which should be modified directly
#######################################################################

abc.rej=function(sum.stat, assign, sims, coord, yr, cov=c("whitmat", "powexp","cauchy"))  {
  library(SpatialExtremes)
  library(combinat)
  if(missing(sum.stat) || missing(assign) || missing(sims) || missing(coord) || missing(cov)) {
    stop("Need to specify summary, assign, sims, coord, and cov")} else {}
  n=dim(coord)[1]
  library(combinat)
  C=combn(seq(1:n), 3)
  phi=sum.stat[[1]]
  len=sum.stat[[2]]
  wt=sqrt(-1*len*(len<0)+(len*(len>0)))

  running=array()
  for (i in 1:sims) {
    if(floor(i/100)-floor((i-1)/100)>0) {print(i)} else{}
    rastar=runif(1,0,10)
##  smstar=runif(1,0,10)
    smstar=1
   datastar=rmaxstab(yr, coord, cov.mod = cov, nugget=1, range=rastar, smooth=smstar, grid=FALSE)

    datastart=array()
    for (t in 1:n) {
     hold=gev2frech(datastar[,t], loc=gevmle(datastar[,t])[1], scale=gevmle(datastar[,t])[2],
        shape=gevmle(datastar[,t])[3])
      datastart=cbind(datastart, hold)
    }
    datastart=datastart[,2:(n+1)]

    phistart=array()
    for (i in 1:dim(C)[2]) {
      phistart[i]=yr/sum(pmin(1/datastart[,C[,i][1]], 1/datastart[,C[,i][2]],
        1/datastart[,C[,i][3]]))
    }

    phistar=array()
    for (q in 1:length(sum.stat[[1]])) {
      phistar[q]=mean(phistart[which(assign==q)])
    }
  dist=sum(wt*abs(phistar-phi))
  running=rbind(running,cbind(rastar, smstar, dist  ))
  } ## closes loop on number of simulations)
  out=running[-1,]
}
```

---

cluster *cluster*

Description

This function geometrically clusters the (n choose 3) triplets into k groups using either Ward's method, or the k-means++ algorithm

Usage

```
cluster(coord, k, method = c("Ward", "kmeans"))
```

Arguments

coord	The two dimensional coordinates of the observed data. Rows are locations, columns are coordinates.
k	The desired number of clusters k (same as the dimension of the summary statistic)
method	Choose either "Ward" or "kmeans" for Ward's and kmeans clustering, respectively. K-means++ may run faster for datasets with a large number of locations (n).

Value

Returns a vector of dimention (n choose 3), where n is the number of locations of observed data. Each element is the cluster assignment from 1 to k.

Examples

```
library(SpatialExtremes)
n=10
a=runif(n,0,10)
b=runif(n,0,10)
coord=cbind(a,b)
k=5
yr=30
data=rmaxstab(yr, coord, cov.mod = "whitmat", nugget=1, range=3, smooth=1, grid=FALSE)

plot(coord)

out=cluster(coord, k, "kmeans")
out
```

---

is.wholenumber *is.wholenumber*

Description

An internal function to check if a variable is a positive integer

Usage

```
is.wholenumber(x, tol = .Machine$double.eps^0.5)
```

Arguments

x	a number
tol	tolerance used to check if whole.

Details

(There should be no reason to run or modify this function)

Examples

```
is.wholenumber = function (x, tol = .Machine$double.eps^0.5)
abs(x - round(x)) < tol
```

---

summarize *summarize*

Description

This function produces the summary statistic from unit Frechet data and assigned groups. It is called by >summary.stat(), which allows for transforming the margins. Use >summary.stat() to produce a summary statistic.

Usage

```
summarize(datat, assign)
```

Arguments

datat	A matrix of spatial extremes data transformed to unit Frechet margins. Rows are replicates, columns are locations. See the SpatialExtremes package for more information on transforming all margins to unit Frechet.
assign	A vector of assignments of each of the (n choose 3) triplets into one of the k groups.

Value

A vector of dimension k, where each element is the average extremal tripletwise coefficient in the kth cluster.

summarize.stat *summarize.stat*

Description

This function produces the summary statistic used in ABC rejection sampling. It allows for spatial extremes data without unit Frechet margins. It calls the >summary() function.

Usage

```
summarize.stat(data, assign, frechet)
```

Arguments

data	Spatial extremes data in matrix form. Each row is an independent replicate, and each column is one location. The order of the columns must match the order of the rows in coord. Need not have unit Frechet margins.
assign	The vector assignments of the triplets into one of the k groups. This comes from the >cluster() function.
frechet	A logical indicator if the data already has unit Frechet margins or not. If not, data are first transformed to unit Frechet.

Value

Produces a k dimensional vector whose elements are the average extremal tripletwise coefficients in each group.

Examples

```
## Simulate 100 years of a max-stable process at 25 randomly placed locations on a 10 by 10 square.
## Notice that the true parameter is (range,smooth) = (1,3)
library(SpatialExtremes)
n=10
a=runif(n,0,10)
b=runif(n,0,10)
coord=cbind(a,b)
yr=100
data=rmaxstab(yr, coord, cov.mod = "whitmat", nugget=1, range=3, smooth=1, grid=FALSE)
plot(coord)

## cluster the triplets into k groups.  The command will write the number of clusters that change
## assignment each iteration, and the method stops when there are 0 changes.
k=5
assign=cluster(coord, k, "kmeans")

## Form an summary statistic for the observed data
sumstat=summarize.stat(data, assign, frechet="F")
sumstat
```

Index