Package ‘Kcop’

January 20, 2025

Title Smooth Test for Equality of Copulas and Clustering Multivariate

Version 1.0.0

Description Implements approaches of non-parametric smooth test to compare simultaneously K(K>1) copulas and non-parametric clustering of multivariate populations with arbitrary sizes.
See Yves I. Ngounou Bakam and Denys Pommeret (2022) <arXiv:2112.05623> and Yves I. Ngounou Bakam and Denys Pommeret (2022) <arXiv:2211.06338>.

License GPL (>= 3)

Encoding UTF-8

LazyData false

RoxygenNote 7.1.1

Depends R (>= 2.10)

Imports base, copula,dplyr, gtools, orthopolynom, stats

NeedsCompilation no

Author Yves Ismael Ngounou Bakam [aut, cre],
Denys Pommeret [aut]

Maintainer Yves Ismael Ngounou Bakam <yvesngounou20@gmail.com>

Repository CRAN

Date/Publication 2022-11-16 15:00:20 UTC

Contents

KcopClust — *Nonparametric clustering of multivariate populations with arbitrary sizes*

Description

This function performs the data driven clustering procedure to cluster K multivariate populations of arbitrary sizes into N subgroups characterized by a common dependence structure where the number N of clusters is unknow and will be automatically chosen by our approach. The method is adapted to paired population and can be used with panel data. See the paper at the following arXiv weblink: https://arxiv.org/abs/2211.06338 for further information.

Usage

```
KcopClust(Kdata, dn = 3, paired = FALSE, alpha = 0.05)
```

Arguments

Kdata	A list of the K dataframe or matrix
dn	Number of copulas coefficients considered
paired	A logical indicating whether to consider the datas as paired
alpha	The significance level used in our decision rule.

Value

A list with three elements: the number of identified clusters; 2) the cluster affiliation; 3) the discrepancy matrix. the numbers in the clusters refer to the population indexes of the data list

Author(s)

Yves I. Ngounou Bakam and Denys Pommeret

Examples

```
## simulation of 5 three-dimensional populations of different sizes
Packages <- c("copula","gtools","dplyr", "orthopolynom", "stats")
lapply(Packages, library, character.only = TRUE) # if necessary
set.seed(2022)
dat1<-rCopula(50, copula = gumbelCopula(param=6,dim = 2))
dat2<-rCopula(60, copula = claytonCopula(param=0.4,dim = 2))
dat3<-rCopula(55, copula = claytonCopula(param=0.4,dim = 2))
## Form a list of data
Kdata<-list(data1=dat1,data2=dat2,data3=dat3)
## Applying the clustering
KcopClust(Kdata = Kdata)
```

`KcopTest` *Nonparametric smooth test for equality of copulas*

Description

This functions performs the nonparametric smooth test to compare simultaneously K(K>1) copulas. See 'Details' below for further information.

Usage

```
KcopTest(Kdata, dn = 3, paired = FALSE)
```

Arguments

`Kdata`	A list of the K dataframe or matrix
`dn`	Number of copulas coefficients considered
`paired`	A logical indicating whether to consider the datas as paired

Details

Recall that we have K multivariate populations of arbitrary sizes, possibly paired with unknow associated copulas `C_1,...,C_K` respectively. `KcopTest` performs the following hypothesis H0: C_1=C_2=...=C_K against H1: C_l differs from C_m (l different from m and l,m in 1:K). The test is based on copulas cross-moments founded on Legendre polynomials that he called copulas coefficients. See the paper at the following HAL weblink: https://hal.archives-ouvertes.fr/hal-03475324v2

Value

A list with three elements: the p-value of the test, the value of the test statistic and the selected rank of copulas coefficients (number of terms involved in the test statistic)

Author(s)

Yves Ismael Ngounou Bakam

Examples

```
## simulation of 5 three-dimensional populations of different sizes
Packages <- c("copula","gtools","dplyr", "orthopolynom", "stats")
lapply(Packages, library, character.only = TRUE) # if necessary
set.seed(2022)
dat1<-rCopula(50, copula = gumbelCopula(param=6,dim = 2))
dat2<-rCopula(60, copula = claytonCopula(param=0.4,dim = 2))
dat3<-rCopula(55, copula = claytonCopula(param=0.4,dim = 2))
## Form a list of data
Kdata<-list(data1=dat1,data2=dat2,data3=dat3)
## Applying the test
KcopTest(Kdata = Kdata)
```

Index