Package 'copent'

March 21, 2021

Version 0.2

Date 2021-03-20

Title Estimating Copula Entropy and Transfer Entropy

Author MA Jian [aut, cre]

Maintainer MA Jian <majian03@gmail.com>

Depends R (>= 2.7.0)

Imports stats

Suggests mnormt

Description The nonparametric methods for estimating copula entropy and transfer entropy are implemented. The method for estimating copula entropy composes of two simple steps: estimating empirical copula by rank statistic and estimating copula entropy with k-Nearest-Neighbour method. The method for estimating transfer entropy composes of two steps: estimating three copula entropy terms and then calculate transfer entropy from the estimated copula entropy terms. Copula Entropy is a mathematical concept for multivariate statistical independence measuring and testing, and proved to be equivalent to mutual information. Estimating copula entropy can be applied to many cases, including but not limited to variable selection and causal discovery (by estimating transfer entropy). Please refer to Ma and Sun (2011) <doi:10.1016/S1007-0214(11)70008-6> and Ma (2019) <arXiv:1910.04375> for more information.

License GPL (>= 2)

URL https://github.com/majianthu/copent

NeedsCompilation no

Repository CRAN

Date/Publication 2021-03-21 05:20:02 UTC

R topics documented:

`ci` *Conditional independence test with copula entropy*

Description

Testing conditional independence between (x,y) conditioned on z with copula entropy.

Usage

```
ci(x,y,z,k=3,dt=2)
```

Arguments

`x`	the data with 1 row.
`y`	the data with 1 row.
`z`	the data with 1 row.
`k`	kth nearest neighbour, default = 3.
`dt`	the type of distance between samples, 1 for Eclidean distance; 2 for Maximum distance.

Details

This program involves testing conditional independence between (**x,y**) conditioned on **z** with copula entropy nonparametrically. It was proposed in Ma (2019).

The algorithm composes of two simple steps: estimating three copula entropy terms with `copent` and then calculate the test statistic.

The argument **x,y,z** are for the data with 1 row and same length as samples from random variables. The argument **k** and **dt** is used in the kNN method for estimating entropy. **k** is for the kth nearest neighbour (default = 3) and **dt** is for the type of distance between samples which has currently two value options (1 for Eclidean distance, and 2(default) for Maximum distance).

Value

The function returns the value of the test statistic of conditional independence.

References

Ma, Jian. Estimating Transfer Entropy via Copula Entropy. arXiv preprint arXiv:1910.04375 (2019).

Examples

```
library(copent)
library(mnormt)
rho1 <- 0.5
rho2 <- 0.6
rho3 <- 0.5
sigma <- matrix(c(1,rho1,rho2,rho1,1,rho3,rho2,rho3,1),3,3)
x <- rmnorm(500,c(0,0,0),sigma)
ci1 <- ci(x[,1],x[,2],x[,3])
```

`construct_empirical_copula` *Construct empirical copula by rank statistic*

Description

Construct empirical copula by rank statistic.

Usage

```
construct_empirical_copula(x)
```

Arguments

x	the data with each row as a sample.

Details

This program involves estimating empirical copula from data by rank statistic nonparametrically. It was proposed in Ma and Sun (2008, 2011). The algorithm is the first step of estimating copula entropy `copent`.

The argument **x** is for the data with each row as a sample from random variables.

Value

The function returns the estimated empirical copula of data **x**.

References

Ma, J., & Sun, Z. (2011). Mutual information is copula entropy. *Tsinghua Science & Technology*, **16**(1): 51-54. See also *ArXiv preprint*, arXiv: 0808.0845, 2008.

Examples

```
library(mnormt)
rho <- 0.5
sigma <- matrix(c(1,rho,rho,1),2,2)
x <- rmnorm(500,c(0,0),sigma)
xc1 <- construct_empirical_copula(x)
```

`copent` *Estimating Copula Entropy*

Description

Estimating copula entropy nonparametrically.

Usage

```
copent(x,k=3,dt=2)
```

Arguments

`x`	the data with each row as a sample.
`k`	kth nearest neighbour, default = 3.
`dt`	the type of distance between samples, 1 for Eclidean distance; 2 for Maximum distance.

Details

This program involves estimating copula entropy from data nonparametrically. It was proposed in Ma and Sun (2008, 2011).

The algorithm composes of two simple steps: estimating empirical copula by rank statistic using `construct_empirical_copula` and then estimating copula entropy with kNN method using `entknn` proposed in Kraskov et al (2004).

The argument **x** is for the data with each row as a sample from random variables. The argument **k** and **dt** is used in the kNN method for estimating entropy. **k** is for the kth nearest neighbour (default = 3) and **dt** is for the type of distance between samples which has currently two value options (1 for Eclidean distance, and 2(default) for Maximum distance).

Copula Entropy is proved to be equivalent to negative mutual information so this program can also be used to estimate multivariate mutual information.

Value

The function returns *negative* value of copula entropy of data **x**.

References

Ma, J., & Sun, Z. (2011). Mutual information is copula entropy. *Tsinghua Science & Technology*, **16**(1): 51-54. See also *ArXiv preprint*, arXiv: 0808.0845, 2008.

Kraskov, A., St\"ogbauer, H., & Grassberger, P. (2004). Estimating Mutual Information. *Physical Review E*, **69**(6), 66138.

Examples

```
library(mnormt)
rho <- 0.5
sigma <- matrix(c(1,rho,rho,1),2,2)
x <- rmnorm(500,c(0,0),sigma)
ce1 <- copent(x,3,2)
```

`entknn` *Estimating entropy from data with kNN method*

Description

Estimating entropy from data with kNN method.

Usage

```
entknn(x,k=3,dt=2)
```

Arguments

`x`	the data with each row as a sample.
`k`	kth nearest neighbour, default = 3.
`dt`	the type of distance between samples, = 1 for Eclidean distance; other for Maximum distance.

Details

This program involves estimating entropy from data by kNN method. It was proposed in Kraskov et al (2004). The algorithm is the second step of estimating copula entropy `copent`.

The argument **x** is for the data with each row as a sample from random variables. The argument **k** and **dt** is used in the kNN method for estimating entropy. **k** is for the kth nearest neighbour (default = 3) and **dt** is for the type of distance between samples which has currently two value options (1 for Eclidean distance, and 2(default) for Maximum distance).

Value

The function returns the estimated entropy value of data **x**.

References

Kraskov, A., St\"ogbauer, H., & Grassberger, P. (2004). Estimating Mutual Information. *Physical Review E*, **69**(6), 66138.

Examples

```
library(mnormt)
rho <- 0.5
sigma <- matrix(c(1,rho,rho,1),2,2)
x <- rmnorm(500,c(0,0),sigma)
xent1 <- entknn(x)
```

`transent` *Estimating transfer entropy via copula entropy*

Description

Estimating transfer entropy via copula entropy nonparametrically.

Usage

```
transent(x,y,lag=1,k=3,dt=2)
```

Arguments

`x`	the data with 1 row.
`y`	the data with 1 row.
`lag`	time lag, >0
`k`	kth nearest neighbour, default = 3.
`dt`	the type of distance between samples, 1 for Eclidean distance; 2 for Maximum distance.

Details

This program involves estimating transfer entropy from **y** to **x** with time lag **lag** via copula entropy nonparametrically. It was proposed in Ma (2019).

The algorithm first prepare the data according to **lag**, and then call `ci` for conditional independence testing.

The argument **x,y** are for the data with 1 row as samples from random variables. The argument **lag** is for time lag. The argument **k** and **dt** is used in the kNN method for estimating entropy. **k** is for the kth nearest neighbour (default = 3) and **dt** is for the type of distance between samples which has currently two value options (1 for Eclidean distance, and 2(default) for Maximum distance).

Value

The function returns the value of transfer entropy from **y** to **x** with time lag **lag**.

References

Ma, Jian. Estimating Transfer Entropy via Copula Entropy. arXiv preprint arXiv:1910.04375 (2019).

Examples

```
library(copent)
num = 300
x = rnorm(num)
y = rnorm(num)
transent(y,x,2)
```

Index