copent: Estimating Copula Entropy and Transfer Entropy in R

MA Jian, PhD

Tsinghua University
majian03@gmail.com

useR! 2021

Contents

Introduction

- Statistical independence and conditional independence are two fundamental conceptes in statistics and machine learning with many applications in different areas.
- Copula Entropy (CE) is a mathematical concept for multivariate statistical independence testing with several good properties, and can be estimated nonparametrically.
- Transfer Entropy (TE), a tool for measuring causality, can be representation with only CE, and therefore can also be estimated nonparametrically via CE.
- The `copent` package in R implements the above methods for estimating CE and TE.
- This talk introduces the implementation of the package and compares it with the other related methods implemented in R on (conditional) independence testing with two real-world data examples.

Copula Theory

Theorem (Sklar's Theorem)

[a] *Given a random vector* $\mathbf{X} = (X_1, \ldots, X_N)$, *its PDF* $p(\mathbf{x})$ *can be represented as*

$$p(\mathbf{x}) = c(\mathbf{u}) \prod_{i=1}^{N} p_i(x_i), \tag{1}$$

where $\mathbf{u} = \{u_i\}$ *are marginal distribution functions of* $\mathbf{X}$, $\{p_i, i = 1, \ldots, N\}$ *are marginal density functions of* $\mathbf{X}$, *and* c *is copula density.*

[a] M. Sklar. "Fonctions de repartition an dimensions et leurs marges". In: *Publ. Inst. Statist. Univ. Paris* 8 (1959), pp. 229–231.

- the core of copula theory
- seperating dependence representation from properties of individual variables

Definition and Theorem

Definition (Copula Entropy)

Let $\mathbf{X}$ be random variables with marginals $\mathbf{u}$ and copula density $c(\mathbf{u})$. CE of $\mathbf{X}$ is defined as

$$H_c(\mathbf{x}) = -\int_{\mathbf{u}} c(\mathbf{u}) \log c(\mathbf{u}) d\mathbf{u}. \tag{2}$$

Theorem

Mutual Information of $\mathbf{X}$ *is equivalent to its negative CE.*

$$I(\mathbf{x}) = -H_c(\mathbf{x}). \tag{3}$$

- the theory of statistical independence measure
- the bridge between copula theory and information theory[1]

[1] Jian Ma and Zengqi Sun. "Mutual information is copula entropy". In: *Tsinghua Science & Technology* 16.1 (2011). See also arXiv preprint arXiv:0808.0845 (2008), pp. 51–54.

Properties and Comparison

- Axiomatic properties of CE
  - multivariate
  - symmetric
  - non-negative, 0 iff independence
  - invariant to monotonic transformation
  - equivalent to correlation coefficient in Gaussian cases
- An ideal measure compared with others

Table: Comparison with other independence measures.

	CE	Distance Correlation	HSIC
Definition	copula based	generalised corr	corr in RKHS
Multivariate	Yes	distance multivariance	dHSIC
Invariance	monotonic trans	No	No
Gaussanity	equivalent to cc	unclear	unclear
Computation	low	high	high

Estimating CE

- **Non-Parametric** Estimation Method[2]
  1. estimating empirical copula density with rank statistic
  2. estimating CE with the KSG entropy estimation method

- Advantages
  - distribution-free, non-parametric
  - tuning-free, insensitive to parameters
  - good convergence
  - easy to implement
  - low computation burden

[2] Jian Ma and Zengqi Sun. "Mutual information is copula entropy". In: *Tsinghua Science & Technology* 16.1 (2011). See also arXiv preprint arXiv:0808.0845 (2008), pp. 51–54.

Estimating Transfer Entropy via CE

Definition (Transfer Entropy)

[a] Let x_t, y_t be two time series observations at time $t = 1, \ldots, N$ of the processes X_t, Y_t. TE $T_{Y\to X}$ from Y to X is defined as

$$T_{Y\to X} = \sum p(x_{t+1}, x_t, y_t) \log \frac{p(x_{t+1}|x_t, y_t)}{p(x_{t+1}|x_t))}. \tag{4}$$

[a] Thomas Schreiber. "Measuring information transfer". In: *Physical Review Letters* 85.2 (2000), p. 461.

1. CE representation of TE
   Ma[3] proved that TE can be represented with only CE as follows:

$$T_{Y\to X} = -H_c(x_{t+1}, x_t, y_t) + H_c(x_{t+1}, x_t) + H_c(y_t, x_t). \tag{5}$$

2. Nonparametric estimation of TE via CE
   1. Estimating three CE terms in (5);
   2. Calculating TE with the estimated CE terms.

[3] Jian Ma. "Estimating Transfer Entropy via Copula Entropy". In: *arXiv preprint arXiv:1910.04375* (2019).

The copent Package: Overview

The `copent` package for estimating CE was developed during the author's PhD study at Tsinghua University, and first released on the CRAN on April 16, 2020. Currently, it implements the methods for estimating CE and TE.

- latest version: 0.2
- including 5 functions

Table: The functions in the package.

Function	Description
`construct_empirical_copula(x)`	constructing empirical copula function from data `x` based on rank statistic
`entknn(x,k,dt)`	estimating entropy from data `x` with the KSG method
`copent(x,k,dt)`	main function for estimating CE by calling the above two functions
`ci(x,y,z,k,dt)`	testing conditional independence between `(x,y)` conditioned on `z`
`transent(x,y,lag,k,dt)`	estimating TE from `y` to `x` with time lag `lag`

Note: `k,dt` are the arguments for k^{th} nearest neighbour and distance type of the KSG algorithm respectively.

Functions for Estimating CE

1. `construct_empirical_copula`
   This function estimates copula density from data with rank statistic.

```
construct_empirical_copula(x)
```

2. `entknn`
   This function implements the KSG method for estimating entropy.

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

3. `copent`
   main function which implements the nonparametric method for estimating CE. It returns *negative* CE for convenience.

```
copent<-function(x,k=3,dt=2){
  xc = construct_empirical_copula(x)
  -entknn(xc,k,dt)
}
```

Function for Conditional Independence Testing

1. ci

   This function implements the method for testing conditional independence between (`x,y`) conditioned on `z` by calling the function `copent` three times according to (5).

```
ci<-function(x,y,z,k=3,dt=2){
  xyz = cbind(x,y,z)
  xz  = cbind(x,z)
  yz  = cbind(y,z)
  copent(xyz,k,dt) - copent(xz,k,dt) - copent(yz,k,dt)
}
```

Function for Estimating TE

1. transent
   This function implements the method for estimating TE from `y` to `x` with time lag `lag` by simply calling the function `ci` after preparing the data according to `lag`.

```
transent<-function(x,y,lag=1,k=3,dt=2){
  l = length(x)
  x1 = x[1:(l-lag)]
  x2 = x[(lag+1):l]
  y1 = y[1:(l-lag)]
  ci(x2,y1,x1,k,dt)
}
```

Example I

Variable Selection[4][5]

[4] Jian Ma. "Variable Selection with Copula Entropy". In: *Chinese Journal of Applied Probability and Statistics* (accepted). See also arXiv preprint arXiv:1910.12389 (2019).

[5] The code for this example is available at `https://github.com/majianthu/aps2020`.

Variable Selection with CE

- CE based method
  To select variables based on ranks of their negative CE values with target
- Other related measures in R
  - Hilbert-Schmidt Independence Criterion (HSIC): `dHSIC`
  - Distance Correlation: `energy`
  - Heller-Heller-Gorfine Tests of Independence: `HHG`
  - Hoeffing's D Test: `independence`
  - Bergsma-Dassios T* sign covariance: `independence`
  - Ball Correlation: `Ball`

```r
library(copent) # Copula Entropy
library(energy) # Distance Correlation
library(dHSIC) # Hilbert-Schmidt Independence Criterion
library(HHG) # Heller-Heller-Gorfine Tests of Independence
library(independence) # Hoeffding's D test or Bergsma-
    Dassios T* sign covariance
library(Ball) # Ball correlation
```

UCI Heart Disease Data

The data set contains 4 databases collected from four different locations worldwide, including 899 samples without missing values. Each sample has 76 attributes concerning heart disease diagnosis (#58 for diagnosis), 13 attributes of which were recommended by professionals as clinical relevant.

```r
scan_data <-function(filename1, nl = 0){
  url1 = paste("http://archive.ics.uci.edu/ml/machine-
      learning-databases/heart-disease/",filename1,sep="")
  data1 = scan(url1, nlines = nl, what = c(as.list(rep
      (0,75)),list(""))))
  l = length(data1[[1]])
  data1m = matrix(unlist(data1), l, 76)
  matrix(as.numeric(data1m[,1:75]), l, 75)
}
h1 = scan_data("cleveland.data", 282*10)
h2 = scan_data("hungarian.data")
h3 = scan_data("switzerland.data")
h4 = scan_data("long-beach-va.data")
heart1 = as.matrix( rbind(h1,h2,h3,h4) )
```

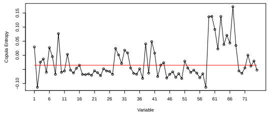
Code of the Example

The dependences between #58 attribute (diagnosis) and the other attributes are estimated with the 6 measures as follows:

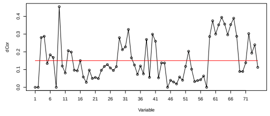
```
for (i in 1:76){
  ce58[i] = copent(heart1[,c(i,58)])
  dcor58[i] = dcor(heart1[,i],heart1[,58])
  dhsic58[i] = dhsic(heart1[,i],heart1[,58])$dHSIC
  Dx = as.matrix(dist((heart1[,i]),diag=TRUE,upper=TRUE))
  Dy = as.matrix(dist((heart1[,58]),diag=TRUE,upper=TRUE))
  hhg58[i] = hhg.test(Dx,Dy, nr.perm = 500)
  ind58[i] = hoeffding.D.test(heart1[,i],heart1[,58])$Dn
  ball58[i] = bcor(heart1[,i],heart1[,58])
}
```

Selection Results

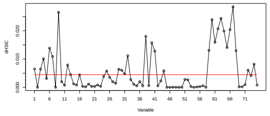
The figures below show the dependence between #58(diagnosis) and the other attributes. The red lines are all the dependence between #58 and #16, which are taken as the selection threshold for each measure.



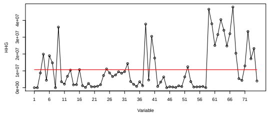
(a) CE



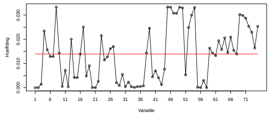
(b) dCor



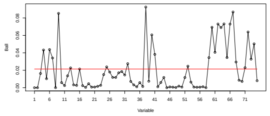
(c) dHSIC



(d) HHG



(e) Hoeffding's D



(f) Ball Correlation

Figure: Variables selected with the 6 measures.

Interpretability of the Selections

- Number of the selected recommended variables
  CE selects more recommended variables with biomedical meanings than the other measures do.

Table: Selected variables with the 6 measures.

Measure	**Selected Variables' ID**	✓
CE	3,4,6,7,9,12,16,28-32,38,40,41,44,51,59-68	**11**
dCor	3,4,6,7,9,12,13,16,28-33,38,40,41,52,59-68	9
dHSIC	3,4,6,7,9,12,13,16,25,29-32,38,40,41,44,59-68	10
HHG	4,6,7,9,16,25,32,38,40,41,52	7
Hoeffding's D	4,5,8,9,13,16,17,23,26,27,38,39,45-50,52-54	4
Ball	4,6,7,9,13,16,25,32,38,40,41,52	7
Recommendations	3,4,9,10,12,16,19,32,38,40,41,44,51	13

Example II

Causal Discovery[6][7]

[6] Jian Ma. "Estimating Transfer Entropy via Copula Entropy". In: *arXiv preprint arXiv:1910.04375* (2019).

[7] The code for this example is available at `https://github.com/majianthu/transferentropy`.

Causal Discovery

- Goal
  To infer causality from time series data by *estimating TE*
- Other Related Methods in R
  - Kernel-based Conditional Independence (KCI): `CondIndTests`
  - Conditional Distance Correlation (CDC): `cdcsis`
  - COnditional DEpendence Coefficient (CODEC): `FOCI`

```r
library(copent)
library(CondIndTests)
library(cdcsis)
library(FOCI)
```

UCI Beijing PM2.5 Data

- Overview
  - Time & Location
    hourly data from 2010-01-01 to 2014-12-31, including PM2.5 data of US Embassy in Beijing and meteorological data from Beijing Capital International Airport
  - Meteorological factors
    dew point, temperature, pressure, cumulated wind speed, combined wind direction, cumulated hours of snow, cumulated hours of rain.
- Experimental data
  - the 'pressure' factor used in the example;
  - 501 samples without missing values (2010-04-02∼2010-04-23).

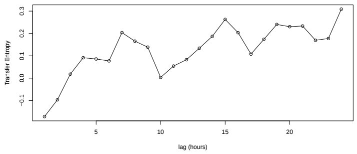
```r
ucidata = read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/00381/PRSA_data_2010.1.1-2014.12.31.csv")
data = ucidata[2200:2700, c(6,9)] # 6(PM2.5),9(Pressure)
```

Code of the Example

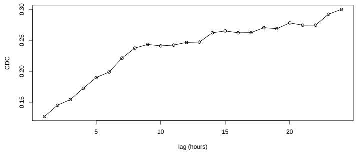
The causality from pressure to PM2.5 with time lag from 1h to 24h is estimated with the 4 measures as follows:

```
for (lag in 1:24){
  pm25a = data[1:(501-lag),1]
  pm25b = data[(lag+1):501,1]
  v1 = data[1:(501-lag),2]

  te1[lag] = transent(data[,1],data[,2],lag)
  # te1[lag] = ci(pm25b,v1,pm25a)

  kci1[lag] = KCI(pm25b,v1,pm25a)$testStatistic
  cdc1[lag] = cdcor(pm25b,v1,pm25a)
  codec1[lag] = codec(pm25b,v1,pm25a)
}
```

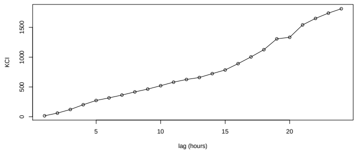
Results



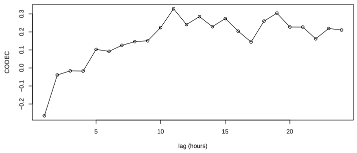
(a) TE via CE



(b) CDC



(c) KCI



(d) CODEC

Figure: Estimated causality from pressure to PM2.5 with lags from 1h to 24h.

Summary

- The theory of CE and the estimation methods of CE and TE are introduced.
- **copent**, the R package for estimating TE and CE, is introduced with implementation details.
- The examples on variable selection and causal discovery demonstrate the usage of the **copent** package and compare it with the related R packages.

References

1. Jian Ma and Zengqi Sun. "Mutual information is copula entropy". In: *Tsinghua Science & Technology* 16.1 (2011). See also arXiv preprint arXiv:0808.0845 (2008), pp. 51–54
2. Jian Ma. "Variable Selection with Copula Entropy". In: *Chinese Journal of Applied Probability and Statistics* (accepted). See also arXiv preprint arXiv:1910.12389 (2019)
3. Jian Ma. "Estimating Transfer Entropy via Copula Entropy". In: *arXiv preprint arXiv:1910.04375* (2019)
4. Jian Ma. "copent: Estimating Copula Entropy and Transfer Entropy in R". In: *arXiv preprint arXiv:2005.14025* (2020)

`http://arxiv.org/a/ma_j_3`

Softwares

https://cran.r-project.org/package=copent

https://pypi.org/project/copent

https://github.com/majianthu

The package **copent**[8] in R and Python for estimating copula entropy and transfer entropy are available on CRAN and PyPI respectively. The source codes are provided on GitHub.

[8] Jian Ma. "copent: Estimating Copula Entropy and Transfer Entropy in R". In: *arXiv preprint arXiv:2005.14025* (2020).

Enjoy the Power of Copula Entropy!