Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

copent: Estimating Copula Entropy in R

Jian MA

Abstract

Statistical independence and conditional independence are the fundemental concepts in statistics and machine learning. Copula Entropy is a mathematical concept for multivariate statistical independence measuring and testing, and also closely related to conditional independence or transfer entropy. It has been applied to solve several statistical or machine learning problems, including association discovery, structure learning, variable selection, and causal discovery. Copula entropy was proposed to be estimated nonparametrically with rank statistic and the kNN method for estimating entropy. **copent**, is a R package which implements this proposed method for estimating copula entropy. The implementation detail of the package is presented in this paper. Two illustration examples with simulated data and real-world data on causal discovery are also presented. The **copent** package is available on the Comprehensive R Archive Network (CRAN) and also on GitHub at `https://github.com/majianthu/copent`.

Keywords: copula entropy, independence, transfer entropy, nonparametrical method, R.

1. Introduction

Statistical independence and conditional independence are the fundemental concepts in statistics and machine learning. The research on mathematical tool for their measurement date back to the early days of the statistics disipline. The most widely used tool is correlation coefficients proposed by Pearson (Pearson 1896). However, it is only applicable to linear cases with Gaussian assumptions. The other tool for statistical independence is Mutual Information (MI) in information theory (Cover and Thomas 2012), which is defined for bivariate cases.

Copula is the theory on representation of dependence relationships (Nelsen 2007; Joe 2014). According to Sklar theorem (Sklar 1959), any probabilistic distribution can be represented as a copula function with marginal functions as inputs. Based on this representation, Ma and Sun (Ma and Sun 2011) proposed a mathematical concept for statistical dependence measurement, named *Copula Entropy (CE)*. They also proved the equivalence between MI and CE. CE enjoys several properties which an ideal statistical independence measure should have,

such as multivariate, symmetric, non-negative (0 iff independence), invariant to monotonic transformation, and equivalent to correlation coefficent in Gaussian cases. The nonparametric method for estimating CE was also proposed in Ma and Sun (2011), which is composed of two simple steps: estimating empirical copula function with rank statistic and estimating CE with the kNN method proposed in Kraskov, Stögbauer, and Grassberger (2004). CE has been applied to solve several problems, including association discovery (Ma 2019), structure learning (Ma and Sun 2008), variable selection (Ma 2019), and causal discovery (Ma 2019).

This paper introduces **copent** (Ma 2020), the R (R Core Team 2020) package which implements the nonparametric method for estimating CE proposed in Ma and Sun (2011), and now is available from CRAN at `https://CRAN.R-project.org/package=copent`. The latest release of the package is available on GitHub at `https://github.com/majianthu/copent`. The **copent** package in Python (Van Rossum and Drake Jr 1995) is also provided on the Python Package Index (PyPI) at `https://pypi.org/project/copent`.

This paper is orgnized as follows: the theory, estimation, and applications of CE are introduced in Section 2, Section 3 presents the implementation details of the **copent** package with an open dataset, two more examples are presented to further demonstrate the usage of the package in Section 4, and Section 5 summarize the paper.

2. Copula Entropy

2.1. Theory

Copula theory unifies representation of multivariate dependence with copula function (Nelsen 2007; Joe 2014). According to Sklar theorem (Sklar 1959), multivariate density function can be represented as a product of its marginals and copula density function which represents dependence structure among random variables. This section is to define an association measure with copula. For clarity, please refer to Ma and Sun (2011) for notations.

With copula density, Copula Entropy is define as follows (Ma and Sun 2011):

Definition 1 (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*

$$H_c(\mathbf{X}) = -\int_{\mathbf{u}} c(\mathbf{u}) \log c(\mathbf{u}) d\mathbf{u}. \tag{1}$$

In information theory, MI and entropy are two different concepts (Cover and Thomas 2012). In Ma and Sun (2011), Ma and Sun proved that MI is actually a kind of entropy, negative CE, stated as follows:

Theorem 1 *MI of random variables is equivalent to negative CE:*

$$I(\mathbf{X}) = -H_c(\mathbf{X}). \tag{2}$$

Theorem 1 has simple proof (Ma and Sun 2011) and an instant corollary (Corollary 1) on the relationship between information containing in joint probability density function, marginals and copula density.

Corollary 1

$$H(\mathbf{X}) = \sum_i H(X_i) + H_c(\mathbf{X}) \tag{3}$$

The above results cast insight into the relationship between entropy, MI, and copula through CE, and therefore build a bridge between information theory and copula theory. CE itself provides a theoretical concept of statistical independence measure.

2.2. Estimation

It is widely considered that estimating MI is notoriously difficult. Under the blessing of Theorem 1, Ma and Sun (Ma and Sun 2011) proposed a non-parametric method for estimating CE (MI) from data which is composed of only two simple steps:

1. Estimating Empirical Copula Density (ECD);
2. Estimating CE.

For Step 1, if given data samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ i.i.d. generated from random variables $\mathbf{X} = \{x_1, \ldots, x_N\}^T$, one can easily estimate ECD as follows:

$$F_i(x_i) = \frac{1}{T} \sum_{t=1}^{T} \chi(\mathbf{x}_t^i \leq x_i), \tag{4}$$

where $i = 1, \ldots, N$ and χ represents for indicator function. Let $\mathbf{u} = [F_1, \ldots, F_N]$, and then one can derives a new samples set $\{\mathbf{u}_1, \ldots, \mathbf{u}_T\}$ as data from ECD $c(\mathbf{u})$.

Once ECD is estimated, Step 2 is essentially a problem of entropy estimation which can be tackled by many existing methods. Among those methods, the kNN method (Kraskov *et al.* 2004) was suggested in Ma and Sun (2011), which leads to a non-parametric way of estimating CE.

2.3. Applications

CE has been applied to solve several typical statistical problems, including:

- Association Measuring (Ma 2019). CE is used as an association measure, which enjoys many advantages over the traditional association measure, such as Pearson correlation coefficient.
- Structure Learning (Ma and Sun 2008). Based on dependence relationship between random variables measured by CE, a graph can be derived with the maximal spanning tree algorithm.
- Variable Selection (Ma 2019). For regression or classification tasks, variables can be selected based on statistical independence strength between variables and target variable measured by CE. Due to the merits of CE, such selection is both model-free and tuning-free.

Function	Description
`construct_empirical_copula(x)`	constructing empirical copula function from data based on rank statistics
`entknn(x,k,dtype)`	estmating entropy from data with the kNN method (Kraskov *et al.* 2004)
`copent(x,k,dtype)`	main function for estimating copula entropy, which is composed of two steps implemented as the above two functions

Table 1: The functions in the package. `x,k,dtype` represent the arguments for data, kth nearest neighbour, and distance type respectively.

- Causal Discovery (Ma 2019). To discover causal relationships from observational data, transfer entropy can be estimated via CE nonparametrically to measure causality. Such estimation makes no assumption on the underlying mechanism and can be applied to any cases provided time series data are available.

3. Implementation

The **copent** package contains three functions as listed in Table 1. The function `copent()` is the main function which implements the method for estimating CE and the other two functions `construct_empirical_copula()` and `entknn()` are called by `copent()` as two steps of the estimation method.

To illustrate the implementation and usage of the functions, we use the "airquality" dataset in R as a working dataset, which contains daily air quality measurements in New York, May to September 1973.

```
R> data("airquality")
R> x1 = airquality[,1:4]
```

The function `construct_empirical_copula()` estimates empirical copula from data with rank statistic. After the four numerical measurements are loaded, the corresponding empirical copula function can be derived by the function `construct_empirical_copula()` as follows:

```
R> xc1 = construct_empirical_copula(x1)
```

The estimated empirical copula of the four measurements is illustrated in Figure 1.

The function `entknn()` implements the kNN method for estimating entropy proposed in Kraskov *et al.* (2004). It is based on the following estimation equation:

$$\hat{H}(X) = -\psi(k) + \psi(N) + \log c_d + \frac{d}{N}\sum_{i=1}^{N} \log \epsilon(i). \tag{5}$$

Here, $\psi()$ is the digamma function; the c_d is the volume of the d-dimensional unit ball, for which two cases are implemented: $c_d = 1$ for the maximum norm and $c_d = \pi^{d/2}/\Gamma(1 + d/2)/2^d$

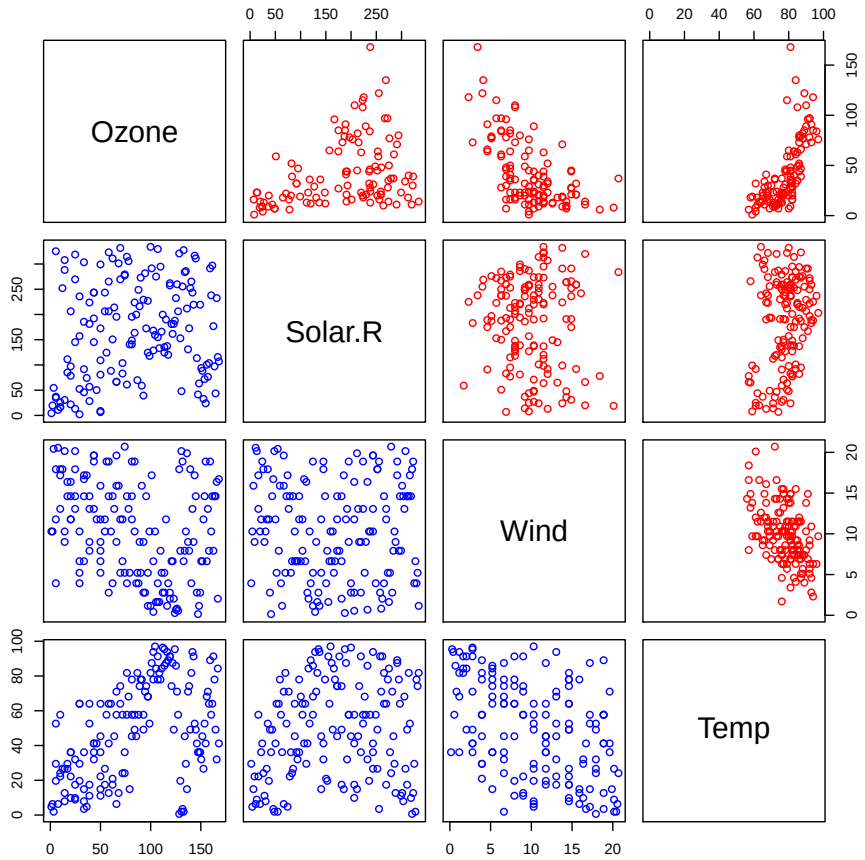

Figure 1: The joint distribution of the four measurements (upper panels) and the estimated empirical copula (lower panels).

for Euclidean norm; and $\epsilon()$ is twice the distance from the sample to its k-th nearest neighbour. In the package, the function `entknn()` has three arguments, two of which are `k` and `dtype`, k-th neighbour and distance type (maximum norm or Euclidean norm) which are used for computing the last two terms in the above estimation equation.

Now we can use the function `entknn()` to estimate the entropy of empirical copula of these four measurements:

```
R> entknn(xc1)

[1] -0.03305222
```

Here we use the default value of `k` and `dtype` because of the good estimation performance of the kNN method for estimating entropy.

The main function `copent()` implements the method in Section 2.2. It simply call the function `construct_empirical_copula()` to derive empirical copula function from data and then use the estimated empirical copula as input of the function `entknn()` to estimate CE. For user's

convenience, the function `copent()` returns *negative* value of CE. Here, the negative CE of the four measurement can be easily estimated with `copent()`:

```
R> copent(x1)

[1] 0.03305222
```

4. Two Examples

To further demonstrate the usage of the **copent** package, two simple examples are presented in this section: the first one based on simulated data and the second one based on real-world data for causal discovery.

4.1. Simulation Example

This demonstration example is based on the simulated data. We generate the simulated data with the **mnormt** (Azzalini and Genz 2020) package.

```
R> library(mnormt)
```

First, 500 data samples are generated from bivariate Gaussian distribution. Without loss of generality, the correlation coefficient ρ is set as `0.75`.

```
R> rho = 0.75
R> sigma = matrix(c(1,rho,rho,1),2,2)
R> x = rmnorm(500,c(0,0),sigma)
```

The negative CE of bivariate Gaussian can be calculated analytically as $-\log(1-\rho^2)/2$:

```
R> truevalue = -0.5 * log(1- rho^2)
R> truevalue

[1] 0.4133393
```

With the function `copent()`, the estimated value is:

```
R> copent(x)

[1] 0.4039309
```

4.2. Example on Causal Discovery

The second example is based on the Beijing PM2.5 dataset on the UCI machine learning repository (Dua and Graff 2017), which is about air pollution at Beijing. This hourly data set contains the PM2.5 data of US Embassy in Beijing. Meanwhile, meteorological data from

Beijing Capital International Airport are also included. The data has been analyzed at month scale (Liang, Zou, Guo, Li, Zhang, Zhang, Huang, and Chen 2015). With this data, we try to discover the causal relationships between meteorological factors and PM2.5 by estimating transfer entropy via CE with the method proposed in Ma (2019).

After the data is loaded, we select only a part of data as the working set. For illustration purpose, the factors on PM2.5 and pressure are chosen. Meanwhile, to avoid the missing values, only a 501 hours data without missing values are used.

```
R> prsa2010data = read.csv('~/workingdir/PRSA2010.csv')
R> data = prsa2010data[2200:2700,c(6,9)]
```

We consider causal relationship from pressure to PM2.5 with time lag from 1 hour to 24 hour. By setting time lag as 1 hour, we prepare the working set as follows:

```
R> lag = 1
R> pm25a = data[1:(501-lag),1]
R> pm25b = data[(lag+1):501,1]
R> pressure1 = data[1:(501-lag),2]
R> data1 = cbind(pm25a, pm25b, pressure1)
```

where `pm25a` and `pm25b` is the PM2.5 time series for 'now' and '1 hour later', and `pressure1` is the pressure time serie for 'now'.

According to Ma (2019), the transfer entropy **TE** from X to Y can be represented with CE as follows:

$$TE(X,Y) = -H_c(Y_{i+1}, Y_i, X_i) + H_c(Y_{i+1}, Y_i) + H_c(Y_i, X_i). \tag{6}$$

So the transfer entropy from pressure to PM2.5 with 1 hour time lag can be easily estimated via three CE terms:

```
R> tslag = copent(data1) - copent(data1[,c(1,2)]) - copent(data1[,c(1,3)])
```

By setting `lag` from `1` to `24`, we get the values of transfer entropy of different time lags as illustrated in Figure 2.

5. Summary

CE provides a fundemental tool for multivariate statistical independence measuring and testing, and can been applied to solve several typical statistical or machine learning problems. CE can be estimating nonparametrically with two simple steps. **copent**, the R package for estimating CE nonparametrically is introduced with implementation details in this paper. Two examples with simulated data and real-world data on causal discovery illustrates the usage of the package. The **copent** package in R is available on the CRAN and also on GitHub at https://github.com/majianthu/copent.

Computational details

The results in this paper were obtained using R 3.6.3 with the **datasets** 3.6.3, **copent** 0.1, and **mnormt** 1.5-7 packages. R itself and all packages used are available from the Comprehensive

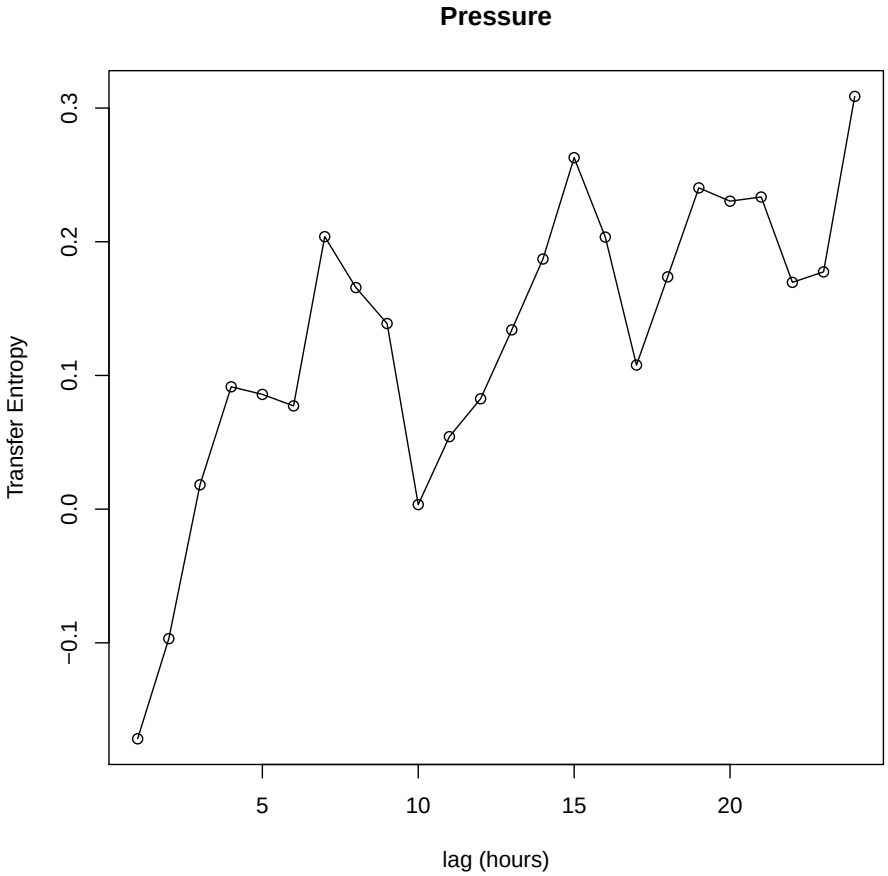

Figure 2: The transfer entropy from pressure to PM2.5 with time lag from 1h to 24h.

R Archive Network (CRAN) at https://CRAN.R-project.org/. The UCI Beijing PM2.5 data dataset was accessed at May 25th, 2020.

Acknowledgments

The code of the **copent** package was developed during the author's PhD study at Tsinghua University.

References

Azzalini A, Genz A (2020). *The R package* ***mnormt****: The multivariate normal and t distributions (version 1.5-7)*. URL http://azzalini.stat.unipd.it/SW/Pkg-mnormt.

Cover TM, Thomas JA (2012). *Elements of information theory*. John Wiley & Sons.

Dua D, Graff C (2017). "UCI Machine Learning Repository." URL http://archive.ics.uci.edu/ml.

Joe H (2014). *Dependence modeling with copulas*. CRC press.

Kraskov A, Stögbauer H, Grassberger P (2004). "Estimating mutual information." *Physical review E*, **69**(6), 066138. doi:10.1103/PHYSREVE.69.066138.

Liang X, Zou T, Guo B, Li S, Zhang H, Zhang S, Huang H, Chen SX (2015). "Assessing Beijing's PM2.5 pollution: severity, weather impact, APEC and winter heating." *Proceedings of The Royal Society A: Mathematical, Physical and Engineering Sciences*, **471**(2182), 20150257. doi:10.1098/RSPA.2015.0257.

Ma J (2019). "Discovering Association with Copula Entropy." *arXiv preprint arXiv:1907.12268*.

Ma J (2019). "Estimating Transfer Entropy via Copula Entropy." *arXiv preprint arXiv:1910.04375*.

Ma J (2019). "Variable Selection with Copula Entropy." *arXiv preprint arXiv:1910.12389*.

Ma J (2020). ***copent**: Estimating Copula Entropy*. R package version 0.1, URL https://CRAN.R-project.org/package=copent.

Ma J, Sun Z (2008). "Dependence Structure Estimation via Copula." *arXiv preprint arXiv:0804.4451*.

Ma J, Sun Z (2011). "Mutual information is copula entropy." *Tsinghua Science & Technology*, **16**(1), 51–54. doi:10.1016/S1007-0214(11)70008-6.

Nelsen RB (2007). *An introduction to copulas*. Springer Science & Business Media.

Pearson K (1896). "Mathematical contributions to the theory of evolution.âĂŤOn a form of spurious correlation which may arise when indices are used in the measurement of organs." *Proceedings of The Royal Society of London*, **60**(1), 489–498.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Sklar M (1959). "Fonctions de repartition an dimensions et leurs marges." *Publications de l'Institut de statistique de l'Université de Paris*, **8**, 229–231.

Van Rossum G, Drake Jr FL (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Affiliation:

Jian MA
E-mail: majian03@gmail.com
URL: https://github.com/majianthu/
ORCID: https://orcid.org/0000-0001-5357-1921

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd