

# Modeling dependence with C- and D-vine copulas: The R-package CDVine

Eike Christian Brechmann  
Technische Universität München

Ulf Schepsmeier  
Technische Universität München

---

## Abstract

Flexible multivariate distributions are needed in many areas. The popular multivariate Gaussian distribution is however very restrictive and cannot account for features like asymmetry and heavy tails. Therefore dependence modeling using copulas is nowadays very common to account for such patterns. The use of copulas is however challenging in higher dimensions, where standard multivariate copulas suffer from rather inflexible structures. Vine copulas overcome such limitations and are able to model complex dependency patterns by benefiting from the rich variety of bivariate copulas as building blocks. This article presents the R-package **CDVine** which provides functions and tools for statistical inference of canonical vine (C-vine) and D-vine copulas. It contains tools for bivariate exploratory data analysis and for bivariate copula selection as well as for selection of pair-copula families in a vine. Models can be estimated either sequentially or by joint maximum likelihood estimation. Sampling algorithms and graphical methods are also included.

*Keywords:* multivariate copula, bivariate copula, canonical vine, D-vine, statistical inference, maximum likelihood estimation, R.

---

## 1. Introduction

In search for flexible multivariate distributions, copula modeling has recently become increasingly popular in many fields of application. Standard references on copula theory include the books by Joe (1997) and Nelsen (2006). The most fundamental theorem, which constitutes the important role of copulas for describing dependence in statistics, is the theorem of Sklar (1959). It establishes the link between multivariate distribution functions and their univariate margins.

Let  $F$  be the  $d$ -dimensional distribution function of the random vector  $\mathbf{X} = (X_1, \dots, X_d)'$  with margins  $F_1, \dots, F_d$ . Then there exists a copula  $C$  such that for all  $\mathbf{x} = (x_1, \dots, x_d)' \in [-\infty, \infty]^d$ ,

$$F(\mathbf{x}) = C(F_1(x_1), \dots, F_d(x_d)). \quad (1)$$

$C$  is unique if  $F_1, \dots, F_d$  are continuous. Conversely, if  $C$  is a copula and  $F_1, \dots, F_d$  are distribution functions, then the function  $F$  defined by (1) is a joint distribution function with margins  $F_1, \dots, F_d$ . In particular  $C$  can be interpreted as the distribution function of a  $d$ -dimensional random variable on  $[0, 1]^d$  with uniform margins. Corresponding densities will be denoted by a small letter  $c$ . Furthermore, the random variables  $X_1, \dots, X_d$  will be assumed to be continuous in the following.

The practical implication of Sklar’s theorem is that the modeling of the marginal distributions can be conveniently separated from the dependence modeling in terms of the copula. The problem in practical applications is how to identify this copula. For the bivariate case, a rich variety of copula families is available and well-investigated (see Joe 1997; Nelsen 2006). However, in arbitrary dimension, the choice of adequate families is rather limited. Standard multivariate copulas such as the multivariate Gaussian or Student-t as well as exchangeable Archimedean copulas lack the flexibility of accurately modeling the dependence among larger numbers of variables. Generalizations of these offer some improvement, but typically become rather intricate in their structure and hence exhibit other limitations such as parameter restrictions.

Vine copulas do not suffer from any of these problems. Initially proposed by Joe (1996) and developed in more detail in Bedford and Cooke (2001, 2002) and in Kurowicka and Cooke (2006), vines are a flexible graphical model for describing multivariate copulas built up using a cascade of bivariate copulas, so-called *pair-copulas*. Such pair-copula constructions (PCCs) decompose a multivariate probability density into bivariate copulas, where each pair-copula can be chosen independently from the others. This allows for a enormous flexibility in dependence modeling. In particular, asymmetries and tail dependence can be taken into account as well as (conditional) independence to build more parsimonious models. Vines thus combine the advantages of multivariate copula modeling, that is separation of marginal and dependence modeling, and the flexibility of bivariate copulas. Their “statistical breakthrough” was due to Aas, Czado, Frigessi, and Bakken (2009) who described statistical inference techniques for the two classes of *canonical (C-)* and *D-vines*.

C- and D-vine copulas have been very successful in many applications, mainly, but not exclusively, in risk management in finance and insurance, see, e.g., Schirmacher and Schirmacher (2008), Chollete, Heinen, and Valdesogo (2009), Heinen and Valdesogo (2009), Mendes, Semeraro, and Leal (2010), Czado, Schepsmeier, and Min (2012), and Nikoloulopoulos, Joe, and Li (2012). Bayesian approaches are followed by Min and Czado (2010), Min and Czado (2011), Smith, Min, Czado, and Almeida (2010), and Hofmann and Czado (2010). In two comparison studies Berg and Aas (2009) and Fischer, Köck, Schlüter, and Weigert (2009) showed the the good performance of vine copulas compared to alternative multivariate copulas. Recent overviews about the vine methodology can be found in Czado (2010) and Kurowicka and Joe (2011), which includes further applications and theory.

So far publicly available and reliable software for C- and D-vine copula inference has been lacking. Only the software tool “Uncertainty analysis with Correlations” (UNICORN, <http://risk2.ewi.tudelft.nl/oursoftware/3-unicorn>) includes some functionality for vines but only to a rather limited extent. We therefore try to fill this gap with the package **CDVine** for the statistical software R (R Development Core Team 2011). It includes functions for statistical inference of C- and D-vine copulas as well as, due to the underlying pair-copula structure, tools for bivariate data analysis. Some other R-packages for copula modeling are available on the Comprehensive R Archive Network (CRAN, <http://cran.r-project.org/>): the comprehensive package **copula** described in Yan (2007) and Kojadinovic and Yan (2010b), the package **fCopulae** (Wuertz *et al.* 2009) and finally the package **nacopula** (Hofert and Mächler 2011) for so-called nested Archimedean copulas, a generalization of Archimedean copulas. Furthermore, our package depends on the packages **igraph** (Csardi and Nepusz 2006) for illustrations of vine trees and **mvtnorm** (Genz, Bretz, Miwa, Mi, Leisch, Scheipl, and Hothorn 2012), which provides efficient implementations of multivariate Gaussian and

Student-t distributions. These will be loaded (if not already loaded) when loading the package **CDVine** by

```
R> library("CDVine")
```

In the following, we assume that this has been done. Additionally, to allow for reproducibility of the results, we preliminarily fix a seed.

```
R> set.seed(10)
```

The remainder of the paper is structured as follows. The required methodological background on vine copulas is provided in Section 2. In Section 3 we then discuss methods for bivariate data analysis, while those for statistical inference of C- and D-vine copulas are treated in Section 4. An illustrative example is presented in Section 5. Section 6 concludes and provides an outlook to further software implementations of the vine copula methodology.

## 2. Methodological background

Vines are a graphical representation to specify so-called pair copula constructions (PCCs) as introduced by [Aas et al. \(2009\)](#). Before we provide more general expressions for C- and D-vines, we motivate the PCC in three dimensions. For this let  $\mathbf{X} = (X_1, X_2, X_3) \sim F$  with marginal distribution functions  $F_1, F_2$  and  $F_3$  and corresponding densities. By recursive conditioning we can write

$$f(x_1, x_2, x_3) = f_1(x_1)f(x_2|x_1)f(x_3|x_1, x_2). \quad (2)$$

By Sklar's theorem (1) we know that

$$f(x_2|x_1) = \frac{f(x_1, x_2)}{f_1(x_1)} = \frac{c_{1,2}(F_1(x_1), F_2(x_2))f_1(x_1)f_2(x_2)}{f_1(x_1)} = c_{1,2}(F_1(x_1), F_2(x_2))f_2(x_2), \quad (3)$$

and

$$\begin{aligned} f(x_3|x_1, x_2) &= \frac{f(x_2, x_3|x_1)}{f(x_2|x_1)} = \frac{c_{2,3|1}(F(x_2|x_1), F(x_3|x_1))f(x_2|x_1)f(x_3|x_1)}{f(x_2|x_1)} \\ &= c_{2,3|1}(F(x_2|x_1), F(x_3|x_1))f(x_3|x_1) \\ &\stackrel{(3)}{=} c_{2,3|1}(F(x_2|x_1), F(x_3|x_1))c_{1,3}(F_1(x_1), F_3(x_3))f_3(x_3). \end{aligned}$$

The three-dimensional joint density (2) can therefore be represented in terms of bivariate copulas  $C_{1,2}, C_{1,3}$  and  $C_{2,3|1}$  with densities  $c_{1,2}, c_{1,3}$  and  $c_{2,3|1}$ , so-called pair-copulas, which may be chosen independently of each other to achieve a wide range of different dependence structures. Typically it is assumed that the conditional copula  $C_{2,3|1}$  is independent of the conditioning variable  $X_1$  to facilitate inference (see [Aas et al. \(2009\)](#) and [Hobæk Haff, Aas, and Frigessi \(2010\)](#)).

Since the decomposition in (2) is not unique, there exist many such iterative PCCs. To classify them [Bedford and Cooke \(2001, 2002\)](#) introduced the graphical model called *vine*, which is also treated in detail in [Kurowicka and Cooke \(2006\)](#) and [Kurowicka and Joe \(2011\)](#).

Vines arrange the  $d(d-1)/2$  pair-copulas of a  $d$ -dimensional PCC in  $d-1$  linked trees (acyclic connected graphs with nodes and edges). In the first C-vine tree, the dependence with respect to one particular variable, the first *root node*, is modeled using bivariate copulas for each pair. Conditioned on this variable, pairwise dependencies with respect to a second variable are modeled, the second root node. In general, a root node is chosen in each tree and all pairwise dependencies with respect to this node are modeled conditioned on all previous root nodes, i.e., C-vine trees have a star structure (see the left panel of Figure 1). This gives the following decomposition of a multivariate density, the *C-vine density* w.l.o.g. with root nodes  $1, \dots, d$  (otherwise nodes can be relabeled),

$$f(\mathbf{x}) = \prod_{k=1}^d f_k(x_k) \times \prod_{i=1}^{d-1} \prod_{j=1}^{d-i} c_{i,i+j|1:(i-1)}(F(x_i|x_1, \dots, x_{i-1}), F(x_{i+j}|x_1, \dots, x_{i-1})|\boldsymbol{\theta}_{i,i+j|1:(i-1)}), \quad (4)$$

where  $f_k$ ,  $k = 1, \dots, d$ , denote the marginal densities and  $c_{i,i+j|1:(i-1)}$  bivariate copula densities with parameter(s)  $\boldsymbol{\theta}_{i,i+j|1:(i-1)}$  (in general  $i_k : i_m$  means  $i_k, \dots, i_m$ ). Here, the outer product runs over the  $d-1$  trees and root nodes  $i$ , while the inner product refers to the  $d-i$  pair-copulas in each tree  $i = 1, \dots, d-1$ . Our three-dimensional example can be interpreted as a C-vine with  $X_1$  as first root node. A more detailed discussion of the C-vine construction and its likelihood can be found in Aas *et al.* (2009) and in Czado *et al.* (2012).

Similarly, D-vines are also constructed by choosing a specific order of the variables. Then in the first tree, the dependence of the first and second variable, of the second and third, of the third and fourth, and so on, is modeled using pair-copulas, i.e., if we assume the order  $1, \dots, d$ , we model the pairs  $(1, 2)$ ,  $(2, 3)$ ,  $(3, 4)$ , etc. In the second tree, conditional dependence of the first and third given the second variable (the pair  $(1, 3|2)$ ), the second and fourth given the third (the pair  $(2, 4|3)$ ), and so on, is modeled. In the same way, pairwise dependencies of variables  $a$  and  $b$  are modeled in subsequent trees conditioned on those variables which lie between the variables  $a$  and  $b$  in the first tree, e.g., the pair  $(1, 5|2, 3, 4)$ . That is each D-vine tree has a path structure (see the right panel of Figure 1). This then leads to the *D-vine density* which also conveniently decomposes a  $d$ -dimensional density (as above the order is w.l.o.g. chosen as  $1, \dots, d$ ; otherwise nodes can be relabeled):

$$f(\mathbf{x}) = \prod_{k=1}^d f_k(x_k) \times \prod_{i=1}^{d-1} \prod_{j=1}^{d-i} c_{j,j+i|(j+1):(j+i-1)}(F(x_j|x_{j+1}, \dots, x_{j+i-1}), F(x_{j+i}|x_{j+1}, \dots, x_{j+i-1})|\boldsymbol{\theta}_{j,j+i|(j+1):(j+i-1)}). \quad (5)$$

Again the outer product runs over the  $d-1$  trees, while the pairs in each tree are designated by the inner product. By ordering the variables appropriately, the above three-dimensional example corresponds to a D-vine with order  $2, 1, 3$ .

The crucial question for inference is how to obtain the conditional distribution functions  $F(x|\mathbf{v})$  for an  $m$ -dimensional vector  $\mathbf{v}$ . For a pair-copula term in tree  $m+1$ , this can easily be established using the pair-copulas of the previous trees  $1, \dots, m$  and by sequentially applying the relationship

$$h(x|\mathbf{v}, \boldsymbol{\theta}) := F(x|\mathbf{v}) = \frac{\partial C_{xv_j|\mathbf{v}_{-j}}(F(x|\mathbf{v}_{-j}), F(v_j|\mathbf{v}_{-j})|\boldsymbol{\theta})}{\partial F(v_j|\mathbf{v}_{-j})}, \quad (6)$$

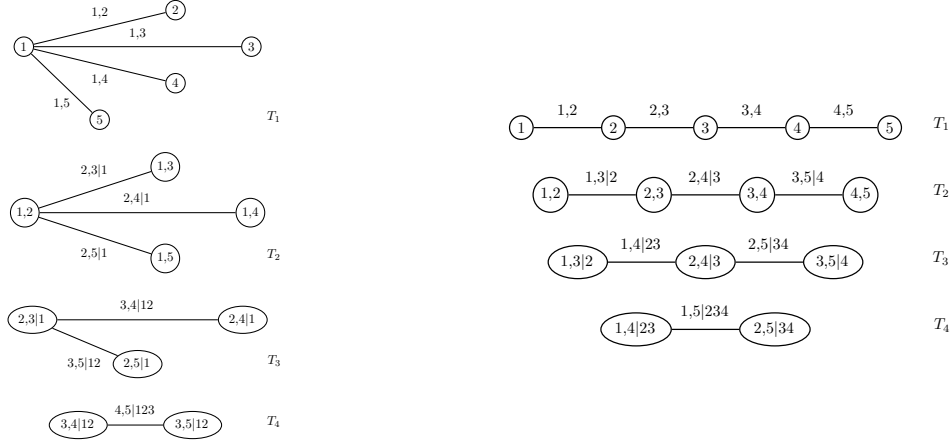


Figure 1: Examples of five-dimensional C- (left panel) and D-vine trees (right panel) with edge indices.

where  $v_j$  is an arbitrary component of  $\mathbf{v}$  and  $\mathbf{v}_{-j}$  denotes the  $(m - 1)$ -dimensional vector  $\mathbf{v}$  excluding  $v_j$  (Joe 1996). Further  $C_{xv_j|\mathbf{v}_{-j}}$  is a bivariate copula distribution function with parameter(s)  $\boldsymbol{\theta}$  specified in tree  $m$ . The notion of the  $h$ -function is introduced for convenience (see Aas *et al.* 2009).

By allowing arbitrary bivariate copulas for each pair-copula term in the decompositions (4) and (5), the multivariate copulas obtained from C- and D-vine structures, so-called *C-vine copulas* and *D-vine copulas*, constitute very flexible models, since bivariate copulas can easily accommodate complex dependence structures such as asymmetric dependence or strong joint tail behavior (see Joe, Li, and Nikoloulopoulos 2010). For this purpose a wide range of bivariate copula families is implemented in the package **CDVine**. Examples of five-dimensional C- and D-vine trees are shown in Figure 1. Here, the order of root nodes in the C-vine is  $1, \dots, 5$ , which also is the order of the first D-vine tree. Edge labels show the indices of the corresponding pair-copula terms.

Fitting a vine copula model involves different steps: First an appropriate vine tree structure has to be identified. Such a structure may either be given by the data itself or has to be selected manually or through expert knowledge. For a given vine structure, adequate copulas have to be selected and, in the next step, estimated. Finally, models need to be evaluated and compared to alternatives. The workflow shown in Figure 2 allocates functions in **CDVine** to these different steps of data analysis and model building.

### 3. Bivariate data analysis methods

Since C- and D-vine copulas as pair-copula constructions are based on bivariate copulas as building blocks, **CDVine** includes a range of tools for bivariate data analysis and inference of bivariate copula families. We hence discuss these methods before turning to functions for statistical inference of C- and D-vine copulas in Section 4.

In the following we further assume that the data we are working with has approximately uniform margins in  $[0, 1]$ , so-called *copula data*. For general data sets this is typically established either by non-parametrically transforming the data with the empirical marginal distribution

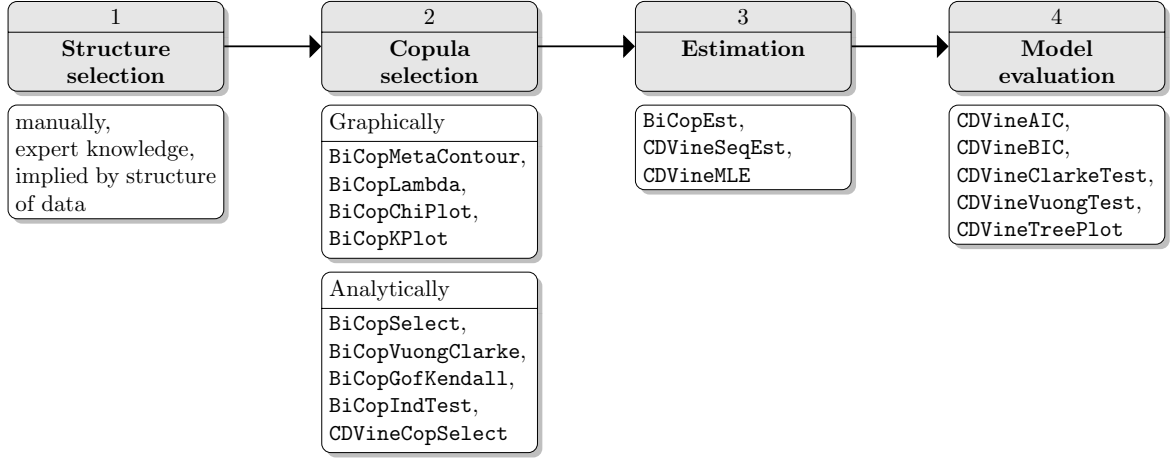


Figure 2: Proposed data analysis and model building workflow and provided functionality in the package **CDVine**.

functions or by choosing (and fitting) appropriate marginal distributions and then applying the parametric distribution functions to the data (see Sklar’s Theorem (1)).

### 3.1. Bivariate copula families

The package **CDVine** provides a wide range of bivariate copula families from the two major classes of elliptical and Archimedean copulas (see Joe 1997; Nelsen 2006). Elliptical copulas are directly obtained by inverting Sklar’s Theorem (1). Given a bivariate distribution function  $F$  with invertible margins  $F_1$  and  $F_2$ , then

$$C(u_1, u_2) = F(F_1^{-1}(u_1), F_2^{-1}(u_2)),$$

is a bivariate copula for  $u_1, u_2 \in [0, 1]$ .  $C$  is called *elliptical* if  $F$  is elliptical. The most famous examples, which are also implemented in **CDVine**, are the bivariate Gaussian copula

$$C(u_1, u_2) = \Phi_\rho(\Phi^{-1}(u_1), \Phi^{-1}(u_2)),$$

and the bivariate Student-t copula

$$C(u_1, u_2) = t_{\rho, \nu}(t_\nu^{-1}(u_1), t_\nu^{-1}(u_2)),$$

with dependence parameter  $\rho \in (-1, 1)$  and degrees of freedom parameter  $\nu > 2$  for the Student-t copula.  $\Phi_\rho$  denotes the bivariate standard normal distribution function with correlation parameter  $\rho$  and  $\Phi^{-1}$  the inverse of the univariate standard normal distribution function. Similarly,  $t_{\rho, \nu}$  is the bivariate Student-t distribution function with correlation parameter  $\rho$  and  $\nu$  degrees of freedom, while  $t_\nu^{-1}$  denotes the inverse univariate Student-t distribution function with  $\nu$  degrees of freedom. Both copulas are obviously symmetric and hence lower and upper tail dependence coefficients are the same.

Bivariate *Archimedean* copulas, on the other hand, are defined as

$$C(u_1, u_2) = \varphi^{[-1]}(\varphi(u_1) + \varphi(u_2)),$$

#	Elliptical distribution	Parameter range	Kendall's $\tau$	Tail dependence
1	Gaussian	$\rho \in (-1, 1)$	$\frac{2}{\pi} \arcsin(\rho)$	0
2	Student-t	$\rho \in (-1, 1), \nu > 2$	$\frac{2}{\pi} \arcsin(\rho)$	$2t_{\nu+1} \left( -\sqrt{\nu+1} \sqrt{\frac{1-\rho}{1+\rho}} \right)$

Table 1: Denotation and properties of bivariate elliptical copula families included in **CDVine**.

where  $\varphi : [0, 1] \rightarrow [0, \infty]$  is a continuous strictly decreasing convex function such that  $\varphi(1) = 0$  and  $\varphi^{[-1]}$  is the pseudo-inverse

$$\varphi^{[-1]}(t) = \begin{cases} \varphi^{-1}(t), & 0 \leq t \leq \varphi(0), \\ 0, & \varphi(0) \leq t \leq \infty. \end{cases}$$

$\varphi$  is called the *generator function* of the copula  $C$  (see [Nelsen 2006](#), for further details).

In **CDVine** we implemented the most common single parameter Archimedean families such as the Clayton, Gumbel, Frank and Joe. Furthermore, the packages provides functionality for four Archimedean copula families with two parameters, namely the Clayton-Gumbel, the Joe-Gumbel, the Joe-Clayton and the Joe-Frank. Following [Joe \(1997\)](#) we simply refer to them as BB1, BB6, BB7 and BB8, respectively. Their more flexible structure allows for different non-zero lower and upper tail dependence coefficients. As boundary cases they include the Clayton and Gumbel, the Joe and Gumbel, the Joe and Clayton as well as the Joe and Frank copulas, respectively.

To each family we assigned a number which is called by the argument **family** in many functions (see the respective first columns of Tables 1 and 2). Corresponding parameters are called by the arguments **par** and **par2**, where **par2** is needed for the degrees of freedom parameter of the Student-t copula as well as for the  $\delta$ -parameter of the BB1, BB6, BB7 and BB8 copulas. By default **par2** is set to zero. The used notation and properties (relationship of parameter(s) to Kendall's  $\tau$  as well as to lower and upper tail dependence coefficients; see [Joe 1996](#); [Nelsen 2006](#), for further details) are shown in Table 1 for bivariate elliptical and in Table 2 for bivariate Archimedean copulas, respectively.

In addition to these families, we also implemented rotated versions of the Clayton (3), Gumbel (4), Joe (6) and the BB families (7,8,9,10). When rotating them by 180 degrees, one obtains the corresponding survival copulas, while rotation by 90 and 270 degrees allows for the modeling of negative dependence which is not possible with the standard non-rotated versions. In particular, the distribution functions  $C_{90}$ ,  $C_{180}$  and  $C_{270}$  of a copula  $C$  rotated by 90, 180 and 270 degrees, respectively, are given as follows:

$$\begin{aligned} C_{90}(u_1, u_2) &= u_2 - C(1 - u_1, u_2), \\ C_{180}(u_1, u_2) &= u_1 + u_2 - 1 + C(1 - u_1, 1 - u_2), \\ C_{270}(u_1, u_2) &= u_1 - C(u_1, 1 - u_2). \end{aligned}$$

To the survival copulas of the Clayton, Gumbel, Joe and the BB copulas the numbers 13, 14, 16, 17, 18, 19 and 20 are assigned, while rotation by 90 degrees is indicated by the numbers 23, 24, 26, 27, 28, 29 and 30 and families 33, 34, 36, 37, 38, 39 and 40 correspond to rotation by 270 degrees. For example, family 24 is a Gumbel copula rotated by 90 degrees, while 16 denotes the Joe survival copula. Note that the parameter ranges of copulas rotated by 90 and



#	Name	Generator function	Parameter range	Kendall's $\tau$	Tail dep. (lower, upper)
3	Clayton	$\frac{1}{\theta}(t^{-\theta} - 1)$	$\theta > 0$	$\frac{\theta}{\theta+2}$	$(2^{-\frac{1}{\theta}}, 0)$
4	Gumbel	$(-\log t)^\theta$	$\theta \geq 1$	$1 - \frac{1}{\theta}$	$(0, 2 - 2^{\frac{1}{\theta}})$
5	Frank	$-\log[\frac{e^{-\theta t} - 1}{e^{-\theta} - 1}]$	$\theta \in \mathbb{R} \setminus \{0\}$	$1 - \frac{4}{\theta} + 4\frac{D_1(\theta)}{\theta}$	$(0, 0)$
6	Joe	$-\log[1 - (1 - t)^\theta]$	$\theta > 1$	$1 + \frac{4}{\theta^2} \int_0^1 t \log(t)(1 - t)^{2(1-\theta)/\theta} dt$	$(0, 2 - 2^{\frac{1}{\theta}})$
7	BB1	$(t^{-\theta} - 1)^\delta$	$\theta > 0, \delta \geq 1$	$1 - \frac{2}{\delta(\theta+2)}$	$(2^{-\frac{1}{\delta\theta}}, 2 - 2^{\frac{1}{\delta\theta}})$
8	BB6	$(-\log[1 - (1 - t)^\theta])^\delta$	$\theta \geq 1, \delta \geq 1$	$1 + \frac{4}{\delta\theta} \int_0^1 (-\log(1 - (1 - t)^\theta)) \times (1 - t)(1 - (1 - t)^{-\theta}) dt$	$(0, 2 - 2^{\frac{1}{\delta\theta}})$
9	BB7	$(1 - (1 - t)^\theta)^{-\delta} - 1$	$\theta \geq 1, \delta > 0$	$1 + \frac{4}{\delta\theta} \int_0^1 (-\log(1 - (1 - t)^\theta)^{\delta+1} \times \frac{(1 - (1 - t)^\theta)^{-\delta} - 1}{(1 - t)^{\theta-1}}) dt$	$(2^{-\frac{1}{\delta}}, 2 - 2^{\frac{1}{\delta}})$
10	BB8	$-\log \left[ \frac{1 - (1 - \delta t)^\theta}{1 - (1 - \delta)^\theta} \right]$	$\theta \geq 1, \delta \in (0, 1]$	$1 + \frac{4}{\delta\theta} \int_0^1 (-\log \left( \frac{(1 - t\delta)^\theta - 1}{(1 - \delta)^\theta - 1} \right) \times (1 - t\delta)(1 - (1 - t\delta)^{-\theta})) dt$	$(0, 0)$

Table 2: Denotation and properties of bivariate Archimedean copula families included in **CDVine**.  $D_1(\theta) = \int_0^\theta \frac{c/\theta}{\exp(x)-1} dx$  is the Debye function. For  $\delta = 1$  the upper tail dependence coefficient of the BB8 copula is  $2 - 2^{1/\theta}$ .

270 degrees are on the negative scale (see Table 2), e.g., the parameter of a rotated Gumbel copula (90/270 degrees) has to be smaller than  $-1$ .

By 0 we denote the independence copula, which is a boundary case of the implemented bivariate copulas, e.g., for the elliptical copulas with  $\rho = 0$  and the Frank copula with  $\theta \rightarrow 0$ . As a reminder of the coding of the copula families the function **BiCopName** transforms a copula family name to its number analogue and vice versa.

The cumulative distribution functions (CDFs) and probability density functions (PDFs) of the bivariate copula families can be found in the books of Joe (1997) and Nelsen (2006) and are implemented in **CDVine** as the functions **BiCopCDF** and **BiCopPDF**, respectively. The example code illustrates the CDF and the PDF of a Student-t copula (**family** = 2) with dependence parameter  $\rho = 0.7$  (**par** = 0.7) and 4 degrees of freedom (**par2** = 4). Perspective plots are shown in Figure 3. Corresponding code can be found in the appendix.

Conditional bivariate distribution functions, the so-called  $h$ -functions defined in (6), can be evaluated using the function **BiCopHfunc**. For bivariate copula data **u1** and **u2** and given bivariate copula family (**family**) and parameter(s) (**par** and **par2**) it returns the  $h$ -functions of **u2** given **u1**,  $h(u_2|u_1, \theta)$ , in the first (**hfunc1**) and of **u1** given **u2**,  $h(u_1|u_2, \theta)$ , in the second argument (**hfunc2**).

To account for the relationship between bivariate copula parameter(s) and Kendall's  $\tau$  and vice versa, the package **CDVine** contains the functions **BiCopPar2Tau** and **BiCopTau2Par**. However note that the inverse relationship (Kendall's  $\tau$  to copula parameter(s)) is only well-defined for one parameter bivariate copulas, i.e., the families 1,3,4,5,6 and the rotated versions of the one parameter Archimedean copulas.

Similarly, the relationship between the copula parameter(s) and the tail dependence coefficients as tabulated in Tables 1 and 2 is implemented in the function **BiCopPar2TailDep**.



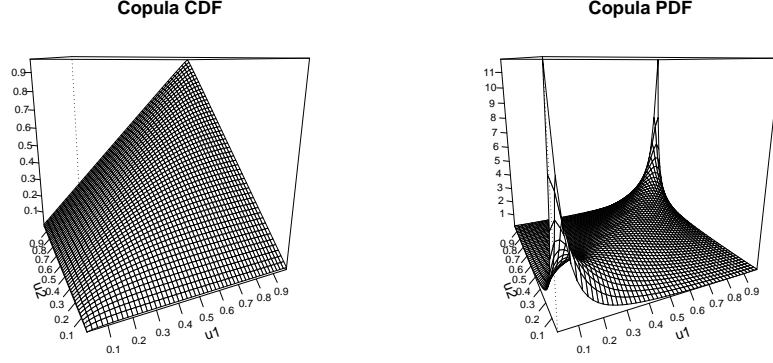


Figure 3: CDF and PDF of a bivariate Student-t copula with dependence parameter  $\rho = 0.7$  and 4 degrees of freedom.

Simulation of general bivariate copula families can easily be established using the probability integral transform. For this, let  $C$  be the bivariate copula under consideration with parameter(s)  $\boldsymbol{\theta}$ . Further, let  $v_1$  and  $v_2$  be two independent uniform samples. Then  $\mathbf{u} = (u_1, u_2)'$  given by

$$\begin{aligned} u_1 &= v_1, \\ u_2 &= h^{-1}(v_2 | u_1, \boldsymbol{\theta}), \end{aligned}$$

with the  $h$ -function as defined in (6), is a sample from the bivariate copula  $C$  with uniform margins.

This is implemented in the function `BiCopSim` which returns a sample of size  $N$  for given bivariate copula family and parameter(s). To illustrate rotated bivariate Archimedean copulas, we simulate samples of size  $N = 500$  from Clayton copulas rotated by 0, 90, 180 and 270 degrees, respectively. Parameters are chosen according to Kendall's  $\tau$  values of 0.5 for positive dependence (`family = 3` and `13`) and  $-0.5$  for negative dependence (`family = 23` and `33`). Corresponding scatter plots are shown in Figure 4. The code can again be found in the appendix.

### 3.2. Tools for bivariate exploratory data analysis

When analyzing (bivariate) data, the true copula describing the dependence is however always unknown. Hence, we require tools to determine an appropriate bivariate copula family to describe the observed dependence pattern (see Step 2 of the proposed workflow in Figure 2). **CDVine** provides graphical as well as analytical tools.

#### *Graphical tools*

One of the most common graphical tools beside the standard scatter plot is the contour plot. `BiCopMetaContour` either plots a bivariate contour plot corresponding to a bivariate meta

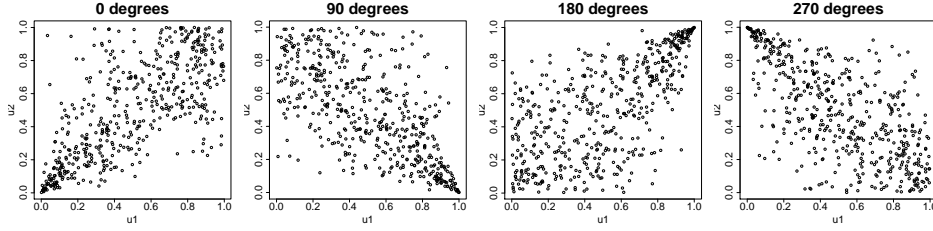


Figure 4: Samples from Clayton copulas rotated by 0, 90, 180 and 270 degrees with parameters corresponding to Kendall's  $\tau$  values of 0.5 for positive dependence and  $-0.5$  for negative dependence.

Distribution	margins	margins.par
Uniform	"unif"	-
Standard normal	"norm" (default)	-
Student-t	"t"	degrees of freedom
Exponential	"exp"	rate
Gamma	"gamma"	(shape, scale)

Table 3: Possible margins for `BiCopMetaContour`.

distribution with specified margins (out of a set of possible margins; one common distribution for both margins) and specified copula family and parameter(s) or creates an empirical contour plot based on bivariate copula data. The choice of margins for `BiCopMetaContour` is summarized in Table 3, where additional parameters for the margins can be set by the argument `margins.par`. Standard normal margins are chosen as default, since they allow for direct comparisons to multivariate normal shapes and bring out characteristic features such as sharpe corners which indicate tail dependence.

Figure 5 shows an empirical contour plot (contours based on an estimated bivariate density) as well as theoretical contour plots (contours based on the theoretical bivariate density) with standard normal and Gamma margins for a Gumbel copula with parameter  $\theta = 2$ . Code is given in the appendix.

While contour plots are rather general tools, there also exist specialized graphical tools to investigate bivariate copula dependence directly. Kendall's plot (K-plot) and the  $\chi$ -plot (or chi-plot) for detecting dependence are well-described in [Genest and Favre \(2007\)](#). The corresponding functions in `CDVine` are `BiCopKPlot` and `BiCopChiPlot`, respectively. Examples of both can be found in Figure 8 of Section 5.

[Genest and Rivest \(1993\)](#) introduced a further method—the  $\lambda$ -function. The  $\lambda$ -function is characteristic for each copula family and defined as

$$\lambda(v, \boldsymbol{\theta}) := v - K(v, \boldsymbol{\theta}),$$

where  $K(v, \boldsymbol{\theta}) := P(C(U_1, U_2 | \boldsymbol{\theta}) \leq v)$  is Kendall's distribution function for a copula  $C$  with parameter(s)  $\boldsymbol{\theta}$ ,  $v \in [0, 1]$  and  $(U_1, U_2)$  distributed according to  $C$ . Note that for Archimedean copulas the  $\lambda$ -function is explicitly given in terms of the generator function  $\varphi$  and its derivative  $\varphi'$  as  $\lambda(v, \boldsymbol{\theta}) = \varphi(v)/\varphi'(v)$  (see [Genest and Rivest 1993](#), for more details).

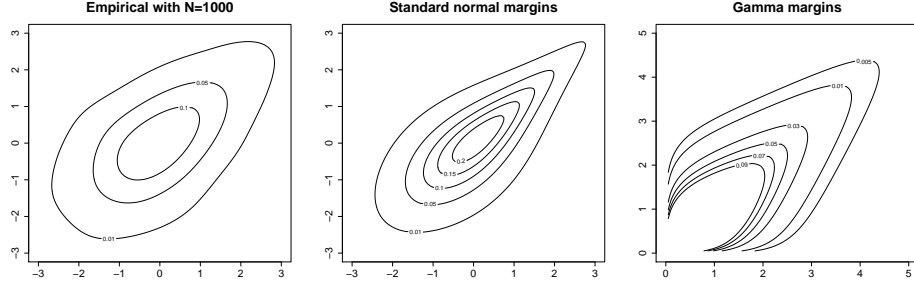


Figure 5: Left panel: empirical contour plot with standard normal margins for simulated data ( $N = 1000$ ) of a Gumbel copula with parameter  $\theta = 2$ . Middle and right panel: meta Gumbel copula distribution with standard normal and Gamma margins with shape parameter 1.5 and scale parameter 0.75. The Gumbel copula parameter is  $\theta = 2$ .

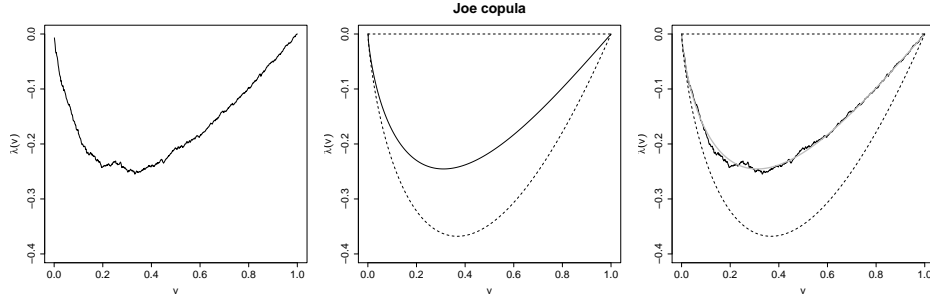


Figure 6: Left panel: empirical  $\lambda$ -function for simulated data ( $N = 1000$ ) of a Joe copula with parameter  $\theta = 2$ . Middle panel: theoretical  $\lambda$ -function of a Joe copula with parameter  $\theta = 2$ . Right panel: both plots combined. The dashed lines in the two rightmost panels are bounds corresponding to independence and comonotonicity ( $\lambda = 0$ ), respectively.

In `BiCopLambda` we implemented the  $\lambda$ -function for the copula families 1 - 10. However note that for the bivariate Gaussian and Student-t copulas no closed form expressions of the theoretical  $\lambda$ -functions exist. Therefore they are simulated based on samples of size 1000. The plot of the theoretical  $\lambda$ -function also shows bounds of the  $\lambda$ -function corresponding to independence and comonotonicity ( $\lambda = 0$ ). For rotated bivariate copulas one can transform the input arguments `u1` and/or `u2` in order to use the  $\lambda$ -function. For copulas rotated by 90 degrees `u1` has to be set to `1-u1`, for 270 degrees `u2` to `1-u2` and for 180 degrees `u1` and `u2` to `1-u1` and `1-u2`, respectively. Then  $\lambda$ -functions of the corresponding non-rotated copula families can be considered.

Comparing empirical to theoretical  $\lambda$ -functions gives an indication which copula family might be appropriate to describe the observed dependence (see Section 5). An illustrative example for the Joe copula with parameter  $\theta = 2$  is shown here: we first produce a plot of the empirical  $\lambda$ -function, then of the theoretical one, and finally a plot showing both (see Figure 6; code can be found in the appendix).

### Analytical tools

In addition to the graphical tools we implemented a range of analytical tools, too, where the numerical output of the plotting functions (set `PLOT = FALSE`) can of course also be considered as analytical. Typically a good start of a bivariate data analysis is an independence test, in particular if the strength of dependence appears to be rather small. In this regard [Genest and Favre \(2007\)](#) propose the use of a simple bivariate independence test based on Kendall's  $\tau$ . The test exploits the asymptotic normality of the test statistic

$$T := \sqrt{\frac{9N(N-1)}{2(2N+5)}} |\hat{\tau}|,$$

where  $N$  is the number of observations and  $\hat{\tau}$  the empirical Kendall's  $\tau$  of the data. The approximate  $p$ -value of the null hypothesis of bivariate independence hence is

$$p\text{-value} = 2 \times (1 - \Phi(T)).$$

Test statistic and  $p$ -value are computed by the function `BiCopIndTest`.

A copula goodness-of-fit test based on Kendall's process for bivariate data, as investigated by [Genest and Rivest \(1993\)](#), is implemented in the function `BiCopGofKendall`. It computes the Cramér-von Mises and Kolmogorov-Smirnov test statistics as well as the corresponding estimated  $p$ -values by bootstrapping (the default are `B = 100` bootstrap samples; note that, if `B` is chosen rather large, computations may take very long). For rotated copulas the input arguments are transformed and the goodness-of-fit procedure for the corresponding non-rotated copula is used (see the discussion of the  $\lambda$ -function above).

A second goodness-of-fit test implemented in **CDVine** is based on a scoring approach. Given a set of bivariate copula families, the function `BiCopVuongClarke` performs for each possible pair of families the asymptotic tests by [Vuong \(1989\)](#) and by [Clarke \(2007\)](#). The Vuong as well as the Clarke test compare two non-nested models against each other and based on their null hypothesis, allow for a statistically significant decision among the two models (see below). In the goodness-of-fit test proposed by [Belgorodski \(2010\)](#) this is used for bivariate copula selection. It compares a bivariate copula model  $C_0$  to all other possible bivariate copula models under consideration in order to determine which family fits the data better than the other families. If copula model  $C_0$  is favored over another copula model, a score of "+1" is assigned and similarly a score of "-1" if the other copula model is determined to be superior. No score is assigned, if the respective test cannot discriminate between two copula models. The total score is the sum of the scores from all pairwise comparisons.

The Vuong and the Clarke tests are suitable to compare two non-nested models. Both are likelihood ratio based and related to the common Kullback-Leibler information criterion, which measures the distance between two statistical models. In the following let  $c_1$  and  $c_2$  be two competing bivariate copula densities with estimated parameters  $\hat{\theta}_1$  and  $\hat{\theta}_2$ , respectively. For the Vuong test we then compute the standardized sum,  $\nu$ , of the log differences of their pointwise likelihoods  $m_i := \log(c_1(u_{i,1}, u_{i,2}|\hat{\theta}_1)) - \log(c_2(u_{i,1}, u_{i,2}|\hat{\theta}_2))$  for observations  $u_{i,j}$ ,  $i = 1, \dots, N$ ,  $j = 1, 2$ , i.e.,

$$\nu = \frac{\frac{1}{n} \sum_{i=1}^N m_i}{\sqrt{\sum_{i=1}^N (m_i - \bar{m})^2}}. \quad (7)$$

Vuong (1989) showed that  $\nu$  is asymptotically standard normal. We hence prefer copula model 1 to copula model 2 at level  $\alpha$  if

$$\nu > \Phi^{-1} \left( 1 - \frac{\alpha}{2} \right).$$

Similarly, if  $\nu < -\Phi^{-1} \left( 1 - \frac{\alpha}{2} \right)$ , we choose model 2. If, however,  $|\nu| \leq \Phi^{-1} \left( 1 - \frac{\alpha}{2} \right)$ , no decision among the models is possible, that is the null hypothesis that both models are statistically equivalent cannot be rejected ( $H_0 : E(m_i) = 0 \forall i = 1, \dots, N$ ).

The null hypothesis of statistical indistinguishability in the Clarke test, on the other hand, is

$$H_0 : P(m_i > 0) = 0.5 \forall i = 1, \dots, N.$$

The intuition behind this null hypothesis is, that under statistical equivalence of the two models the log-likelihood ratios of the single observations are uniformly distributed around zero and in expectation 50% of the log-likelihood ratios are greater than zero. The test statistic

$$B = \sum_{i=1}^N \mathbf{1}_{(0,\infty)}(m_i), \quad (8)$$

where  $\mathbf{1}$  is the indicator function, was proposed by Clarke (2007) and is asymptotically distributed Binomial with parameters  $N$  and  $p = 0.5$ . Based on this, critical values can easily be obtained (see Clarke 2007). Model 1 is interpreted as statistically equivalent to model 2 if  $B$  is not significantly different from the expected value  $Np = \frac{N}{2}$ .

Both test statistics (7) and (8) can be corrected for the number of parameters used in the models, either using the Akaike or the parsimonious Schwarz correction, which correspond to the penalty terms of the AIC (Akaike 1973) and the BIC (Schwarz 1978), respectively. These can be specified using the argument `correction`, while the significance level of the tests is set by `level`.

An example of this scoring goodness-of-fit test will be given in Section 5. The set of copula families to compare is specified by the argument `familyset`.

Commonly used alternative criteria to discriminate among models are the above-mentioned AIC and BIC. They are however less reliable when non-nested models are compared. By correcting the log-likelihood for the number of parameters used in a model, they allow for an efficient comparison based on single numbers, namely among a class of models the model with smallest AIC/BIC is chosen. We implemented this selection procedure in the function `BiCopSelect` which estimates copula parameters for a given set of families to choose from (`familyset`) using maximum likelihood estimation (see the discussion of `BiCopEst` in Section 3.3) and then selects the family based on the AIC (default) or the BIC. Furthermore, a preliminary independence test (see the description of `BiCopIndTest` above) can be performed to accommodate that an independence copula might be appropriate for the given bivariate data anyway. The function returns the selected bivariate copula family and the estimated parameter(s).

### 3.3. Estimation of bivariate copula families

Having selected an appropriate bivariate copula family for given observations, e.g., using the graphical and analytical tools discussed above, the corresponding copula parameter(s)

has/have to be estimated (see Step 3 of the suggested workflow in Figure 2). This can be established using the function `BiCopEst` which performs either a method of moments (inversion of Kendall's  $\tau$  (`method = "itau"`); see Tables 1 and 2 and the function `BiCopTau2Par`) or maximum likelihood estimation (MLE; `method = "mle"`). Note again that the inversion of Kendall's  $\tau$  is however not available for all bivariate copula families but only for the one parameter ones. If possible, starting values for the MLE are obtained by inversion of Kendall's  $\tau$ , while optimization is performed using the L-BFGS-B algorithm for constraint optimization to account for the parameter ranges (see Tables 1 and 2). Furthermore, standard errors for both estimation methods are provided, too (if `se = TRUE`). For MLE standard errors are based on inversion of the Hessian matrix, while for inversion of Kendall's  $\tau$  they are obtained as described in Kojadinovic and Yan (2010a).

As noted above, **CDVine** always assumes that marginally uniform data is given. The MLE used here therefore corresponds to the *inference functions from margins* (IFM; Joe 1997) or *maximum pseudo likelihood* method (MPL; Genest, Ghoudi, and Rivest 1995) depending on whether the transformation to  $[0, 1]$  was parametric or rank based.

To stabilize numerical computations, upper bounds for the degrees of freedom parameter of the Student-t copula as well as for the parameters of the BB copulas (in absolute values) can be specified using the arguments `max.df` for the Student-t copula and `max.BB` for the BB copulas. The default values are based on experience and work quite well in most cases. In certain circumstances, lower or higher values might however be sensible to improve results. In particular, if the degrees of freedom parameter of the Student-t copula is estimated to be quite large (as a rule of thumb 20-30 degrees of freedom can already be regarded as "large"), the Student-t is very similar to the Gaussian copula and therefore it is preferable to work with the Gaussian because it has only one parameter and is thus more efficient when doing inference. A corresponding warning message is returned if this is the case.

## 4. Statistical inference of C- and D-vine copulas

Having discussed techniques for bivariate data analysis, we now turn to the main part of **CDVine**: methods for statistical inference of C- and D-vine copulas. Before discussing estimation and model selection, the coding of C- and D-vines is introduced. Finally, some numerical issues are discussed.

### 4.1. Specification of C- and D-vine copula models and data simulation

As discussed in Section 2, one has to select an order of the variables when specifying C- and D-vine copulas. For the D-vine, the order of the variables in the first tree has to be chosen and for the C-vine, the root nodes for each tree need to be determined. Functions for inference of C- and D-vine copulas in the package **CDVine** assume that the order of the variables in the data set under investigation exactly corresponds to this C- or D-vine order. E.g., in a C-vine the first column of a data set is the first root node, the second column the second root node, etc.. According to this order arguments have to be provided to functions for C- and D-vine copula inference. After choosing which vine type we are working with (`type = 1` or `"CVine"` denotes a C-vine, while `type = 2` or `"DVine"` corresponds to a D-vine), the copula families (`family`) and parameters (`par` and `par2`) have to be specified as vectors of length  $d(d-1)/2$ , where  $d$  is the number of variables. In a C-vine, the entries of this vector correspond to the

following pairs and associated pair-copula terms

$$\begin{aligned}
 (1, 2), (1, 3), (1, 4), \dots, (1, d), & \quad (\text{Tree 1}) \\
 (2, 3|1), (2, 4|1), \dots, (2, d|1), & \quad (\text{Tree 2}) \\
 (3, 4|1, 2), (3, 5|1, 2), \dots, (3, d|1, 2), & \quad (\text{Tree 3}) \\
 \dots, & \\
 (d-1, d|1, \dots, d-2). & \quad (\text{Tree } d-1)
 \end{aligned}$$

Similarly, the pairs of a D-vine are specified in the following order:

$$\begin{aligned}
 (1, 2), (2, 3), (3, 4), \dots, (d-1, d), & \quad (\text{Tree 1}) \\
 (1, 3|2), (2, 4|3), \dots, (d-2, d|d-1), & \quad (\text{Tree 2}) \\
 (1, 4|2, 3), (2, 5|3, 4), \dots, (d-3, d|d-2, d-1), & \quad (\text{Tree 3}) \\
 \dots, & \\
 (1, d|2, \dots, d-1). & \quad (\text{Tree } d-1)
 \end{aligned}$$

As an example consider the following four-dimensional C-vine copula model involving the pair-copula terms  $c_{12}, c_{13}, c_{14}, c_{23|1}, c_{24|1}$  and  $c_{34|12}$ :

```

R> type = 1
R> family = c(1, 3, 6, 2, 1, 5)
R> par = c(0.5, 1.3, 2.1, -0.3, 0.2, 1.7)
R> par2 = c(0, 0, 0, 3, 0, 0)

```

In particular, the pair-copula  $c_{2,3|1}$  is a Student-t with dependence parameter  $\rho = -0.3$  and 3 degrees of freedom, while pair  $c_{3,4|1,2}$  in the last tree is modeled by a Frank copula with parameter  $\theta = 1.7$ . The strength of dependence modeled by each pair-copula term can be illustrated by transforming the parameter(s) of each pair-copula term into the corresponding Kendall's  $\tau$  value using `CDVinePar2Tau` (see `BiCopPar2Tau`).

To simulate from a vine copula specification, the function `CDVineSim` can be used. The corresponding algorithms are given in [Aas et al. \(2009\)](#). They are based on the same idea as the bivariate simulation described in Section 3.1.

## 4.2. Estimation

Having decided the structure of the C- or D-vine to be used, one has to select pair-copula families for each (conditional) pair of variables as described in Section 3.2 or using the function `CDVineCopSelect` (Step 2 in Figure 2). Based on `BiCopSelect`, this function selects for a given copula data set (`data`) and vine type (`type`), appropriate bivariate copula families from a set of possible copula families (`familyset`) according to the AIC (default) or the BIC. As in `BiCopSelect` preliminary independence tests can also be performed for each (conditional) pair to obtain more parsimonious models.

This copula selection proceeds tree by tree, since the conditional pairs in trees 2, ...,  $d-1$  depend on the specification of the previous trees through the  $h$ -functions (see Section 2). Hence, initially C- and D-vine copula models are typically fitted sequentially by proceeding



iteratively tree by tree and thus only involving bivariate estimation for each individual pair-copula term (see, e.g., [Czado et al. \(2012\)](#) for a detailed description of sequential estimation in C-vines; Step 3 in Figure 2). This can be established using the function `CDVineSeqEst` which internally calls the function `BiCopEst` described in Section 3.3. Therefore, estimation of the parameter(s) of each pair-copula can be carried out using inversion of Kendall's  $\tau$  or MLE (`method = "itau" or "mle"`), standard errors can be computed (`se = TRUE or FALSE`) and upper bounds for the Student-t degrees of freedom and BB copula parameters can be set by `max.df` and `max.BB`. A detailed example will be given in Section 5.

Even though these sequential estimates often provide a good fit, one typically is interested in maximizing the (log-)likelihood of a vine copula specification (see (4) and (5)) for observations  $\mathbf{u} = (u_{k,j})_{k=1,\dots,N, j=1,\dots,d}$ :

- The C-vine copula log-likelihood with parameter set  $\boldsymbol{\theta}_{CV}$  is given by

$$\ell_{CV}(\boldsymbol{\theta}_{CV}|\mathbf{u}) = \sum_{k=1}^N \sum_{i=1}^{d-1} \sum_{j=1}^{d-i} \log[c_{i,i+j|1:(i-1)}(F_{i|1:(i-1)}, F_{i+j|1:(i-1)}|\boldsymbol{\theta}_{i,i+j|1:(i-1)})],$$

where  $F_{j|i_1:i_m} := F(u_{k,j}|u_{k,i_1}, \dots, u_{k,i_m})$  and the marginal distributions are uniform, i.e.,  $f_k(u_k) = \mathbf{1}_{[0,1]}(u_k)$ . Note that  $F_{j|i_1:i_m}$  depends on the parameters of pair-copula terms in tree 1 up to tree  $i_m$ .

- Similarly, the D-vine copula log-likelihood with parameter set  $\boldsymbol{\theta}_{DV}$  is:

$$\ell_{DV}(\boldsymbol{\theta}_{DV}|\mathbf{u}) = \sum_{k=1}^N \sum_{i=1}^{d-1} \sum_{j=1}^{d-i} \log[c_{j,j+i|(j+1):(j+i-1)}(F_{j|(j+1):(j+i-1)}, F_{j+i|(j+1):(j+i-1)}|\boldsymbol{\theta}_{j,j+i|(j+1):(j+i-1)})].$$

The log-likelihood of a vine copula for given data (`data`), pair-copula families (`family`) and parameters (`par` and `par2`) can be obtained using the function `CDVineLogLik` which implements the algorithms given in [Aas et al. \(2009\)](#).

Using these log-likelihood calculations, we can now estimate parameters jointly using MLE—in contrast to the pairwise sequential estimation discussed above. This can be established using the function `CDVineMLE` with arguments for the given data (`data`), the pair-copula families (`family`) and corresponding starting values for the parameters (`start` and `start2`), the vine type (`type`) as well as the maximum number of iterations of the optimizer (`maxit`), where the L-BFGS-B algorithm for constraint optimization problems is again used here. Upper bounds for the Student-t degrees of freedom and BB copula parameters can also be set by `max.df` and `max.BB`. Starting values, if not provided, are obtained using the function `CDVineSeqEst`. Note again that here MLE corresponds to the IFM and MPL methods depending on the marginal transformations of the data. More details on the estimation of vine copulas can be found in [Aas et al. \(2009\)](#), [Hobæk Haff \(2010\)](#) and [Czado et al. \(2012\)](#).

The usage of the estimation methods and the calculation of the log-likelihood will be illustrated in Section 5.

### 4.3. Selection among vine copula models

Having fitted different vine copula models to a given data set, one typically is interested in determining the “best” model in terms of one or more criteria (Step 4 in the suggested workflow

in Figure 2). Besides the classical AIC and BIC, implemented in `CDVineAIC` and `CDVineBIC`, two such criteria are the Vuong and the Clarke tests described in Section 3.2. They allow for pairwise comparisons of two competing models, e.g., a C- and a D-vine copula model, and can be performed using the functions `CDVineVuongTest` and `CDVineClarkeTest`. In these functions, models have to be specified as usual: `Model1.family`, `Model1.par`, `Model1.par2` and `Model1.type` for the first model and similarly for the second model. For each model an order of the variables has to be given, since the orders of C-vine root nodes or of the nodes in the first D-vine tree may be chosen differently in the two models. The arguments `Model1.order` and `Model2.order` therefore specify these orders corresponding to the respective vine type. As output, both functions return test statistics with and without correction for the number of parameters as well as corresponding  $p$ -values.

Furthermore, obtained vine specifications can be illustrated using the function `CDVineTreePlot` which plots one or all trees of a specified vine model (either `tree = "ALL"` or a tree number in  $\{1, \dots, d-1\}$ ). If no parameters are provided, these are obtained using sequential estimation, where arguments for `CDVineSeqEst` can be specified. The trees are plotted using the **igraph** package with individually chosen edge labels. As edge labels the user is free to combine the following information in a vector or choose `edge.labels = FALSE` for no edge labels:

- `"family"`: pair-copula family names (default),
- `"par"`: pair-copula parameters,
- `"par2"`: second pair-copula parameters,
- `"theotau"`: theoretical Kendall's  $\tau$  values corresponding to pair-copula families and parameters, or
- `"emptau"`: empirical Kendall's  $\tau$  values, which are available only if data for sequential estimation is provided.

Positions of the nodes are either determined automatically (default) or can be set by the argument `P` which gives  $x$ - and  $y$ -coordinates of the nodes. Node labels can be specified by the argument `names`. An example with code will be given in Figure 9 in Section 5 and in the appendix.

#### 4.4. Implementation and numerical issues

In order to speed up computations we implemented the major parts of the algorithms in C. In particular, the MLE is considerably faster by coding the log-likelihood of C- and D-vine copula models in C. We also implemented the method by Knight (1966) for efficiently computing the empirical Kendall's tau.

Even more important is the question of numerical stability. As noted in Section 3.3, it is advisable to set prudent upper bounds for the estimation of the degrees of freedom parameter of the Student-t copula as well as of the BB1, BB6, BB7 and BB8 copula parameters. In general, the user should be careful when working with parameters that correspond to extreme choices of Kendall's  $\tau$ , that is Kendall's  $\tau$  values close to  $-1$ ,  $0$  and  $1$ . This may for example lead to problems in sequential estimation of pair-copulas in higher order trees of C- and D-vines. For such pair-copulas, dependence is typically rather small and inevitable

rounding errors are amplified, so that weak negative dependence might be observed even if the dependence should actually be positive. If this pair is modeled by a copula family that can only accommodate positive dependence such as the standard Clayton, Gumbel or Joe copulas, the sequential estimation will abort. In such a case, it may be helpful to identify the “problematic” term by setting `progress = TRUE` in `CDVineSeqEst` and then check the copula choice for example using `BiCopSelect`. A simple countermeasure is often to simply set this pair-copula term to a Gaussian copula because copulas close to independence are rather similar anyway. Alternatively, running `CDVineCopSelect` also estimates parameters sequentially and chooses appropriate pair-copula families so that no such problems will occur.

When estimating copula parameters, mostly some bounds have to be set in order to respect the parameter ranges (see Tables 1 and 2). This has been done based on experience and extensive stability tests. Similarly, copula data is bounded to the interval  $[10^{-10}, 1 - 10^{-10}]$  because values too close to 0 or 1 lead to severe numerical problems. Apart from that, additional measures have been taken to improve the stability, but we will not go into the details here.

## 5. Example: Major world stock indices

As an example we choose the `worldindices` data set which is included in the package `CDVine`. This data set contains transformed standardized residuals of daily log returns of major world stock indices in 2009 and 2010 (396 observations). The considered indices are the leading stock exchanges of the six largest economies in the world: the US American S&P 500 (`^GSPC`), the Japanese Nikkei 225 (`^N225`), the Chinese SSE Composite Index (`^SSEC`), the German DAX (`^GDAXI`), the French CAC 40 (`^FCHI`) and the British FTSE 100 Index (`^FTSE`). Each time series is filtered using an ARMA(1,1)-GARCH(1,1) model with Student-t innovations and standardized residuals are transformed non-parametrically to copula data using the respective empirical distribution function.

```
R> data("worldindices")
```

For a first impression of the data Figure 7 shows a pairs plot with scatter plots above and contour plots with standard normal margins below the diagonal. In particular among the European indices there is evidently strong dependence, while the dependence to the two Asian indices is rather weak.

Following the proposed workflow in Figure 2 in Section 2 we will perform a detailed exploratory data analysis (EDA) of one particular variable pair and specify a C-vine copula model including copula selection, sequential estimation and MLE as well as log-likelihood computations and plotting of C-vine trees. This will illustrate the usefulness of our functions and their handling. The specification of a D-vine copula model is not explicitly discussed here, but could be covered in essentially the same way. Such a D-vine copula model is then compared to the selected C-vine copula model at the end of our presentation.

Using the C-vine structure selection criterion described by [Czado et al. \(2012\)](#) we determine `^FCHI` as the first root node (C-vine tree with strongest dependencies in terms of absolute empirical values of pairwise Kendall’s  $\tau$ ’s; this is the first part of Step 1 in the proposed workflow in Figure 2). We now exemplarily show the EDA for the pair (`^FCHI`, `^FTSE`) using the graphical tools `BiCopMetaContour` (see row 6, column 5 of Figure 7), `BiCopKPlot`, `BiCopChiPlot`

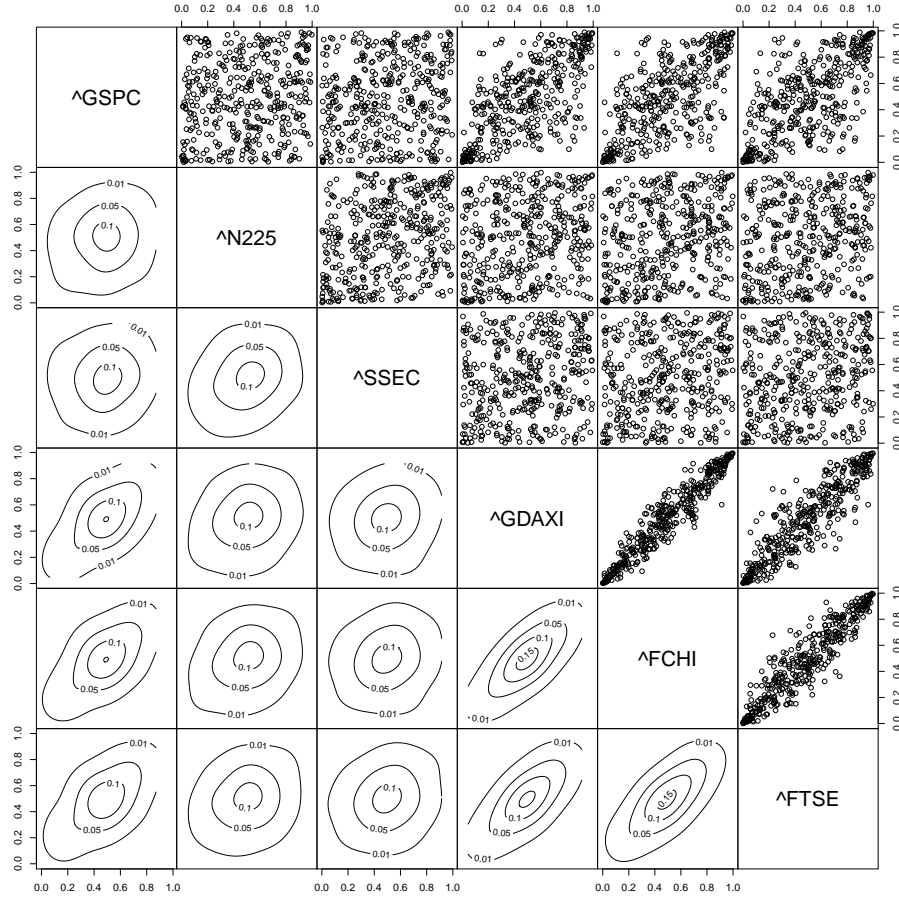


Figure 7: Pairs plot of the `worldindices` data set with scatter plots above and contour plots with standard normal margins below the diagonal. Axes of the contour plots range from  $-3$  to  $3$  other than indicated here (see the appendix for code how to generate this plot).

and `BiCopLambda`, as well as the analytical tools `BiCopIndepTest`, `BiCopVuongClarke` and `BiCopSelect` in order to choose the best fitting copula (Step 2 in Figure 2).

The scatter plot in row 5, column 6 of Figure 7 as well as the K- and chi-plots in Figure 8 show that the variables are strongly positively dependent. Evidence of symmetric tail dependence is also visible. The empirical contour plot confirms these properties which are characteristic for a Student-t copula. Additionally, the corresponding theoretical contour plot of the bivariate Student-t copula (not shown here) has a similar shape as the empirical one in Figure 7. The  $\lambda$ -function in the right panel supports our choice.

The following pro forma independence test with a  $p$ -value of zero confirms the strong dependence.

```
R> BiCopIndTest(worldindices[, 5], worldindices[, 6])$p.value
```

```
[1] 0
```

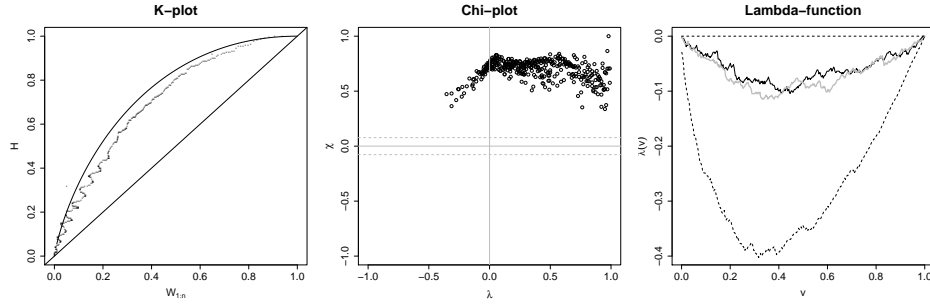


Figure 8: Left panel: K-plot. Middle panel: chi-plot. Right panel: empirical  $\lambda$ -function (black line), theoretical  $\lambda$ -function of a Student-t copula with parameters estimated using **BiCopEst** (grey line) as well as independence and comonotonicity limits (dashed lines). Code for this plot is given in the appendix.

The scoring test based on the Vuong and Clarke tests strongly tends to a Gaussian, Student-t or (survival) BB1 copula, where the Student-t is also selected using the AIC.

```
R> BiCopVuongClarke(worldindices[, 5], worldindices[, 6], familyset = c(1:10,
+ 13, 14, 16:20))
```

	1	2	3	4	5	6	7	8	9	10	13	14	16	17	18	19	20
Vuong	13	13	-11	4	0	-13	13	2	2	-9	-11	3	-13	13	1	2	-9
Clarke	13	16	-12	6	2	-12	13	4	1	-8	-12	4	-12	8	2	-4	-9

```
R> BiCopSelect(worldindices[, 5], worldindices[, 6], familyset = c(1:10,
+ 13, 14, 16:20))$family
```

```
[1] 2
```

Such an EDA or other selection methods as for example goodness-of-fit tests (**BiCopGofKendall** or the one based on the empirical copula process proposed by [Genest and Rémillard \(2008\)](#) and implemented in the package **copula**) can directly be used to select each pair-copula of the first C-vine tree (all pairs involving  $\hat{FCHI}$ ). Based on these pair-copula families and the according estimated parameters, one can then use  $h$ -functions (6) to calculate inputs of the pair-copulas of the second C-vine tree and specify them. This procedure is iterated tree by tree.

By selecting all further C-vine root nodes as described in [Czado et al. \(2012\)](#) the root node order in the data set is determined as  $\hat{FCHI}$ ,  $\hat{N225}$ ,  $\hat{FTSE}$ ,  $\hat{GSPC}$ ,  $\hat{GDAXI}$  and finally  $\hat{SSEC}$ . This completes Step 1 of the proposed workflow in Figure 2. The results of the second step are the following. Copula families (according to the selected order) are chosen as 9 ( $c_{12}$ ), 2 ( $c_{13}$ ), 19 ( $c_{14}$ ), 2 ( $c_{15}$ ), 19 ( $c_{16}$ ), 0 ( $c_{23|1}$ ), 0 ( $c_{24|1}$ ), 34 ( $c_{25|1}$ ), 1 ( $c_{26|1}$ ), 0 ( $c_{34|12}$ ), 0 ( $c_{35|12}$ ), 1 ( $c_{36|12}$ ), 4 ( $c_{45|123}$ ), 0 ( $c_{46|123}$ ), 0 ( $c_{56|1234}$ ), where bivariate independence tests have been used to identify possibly independent conditional variable pairs.

```
R> order = c(5, 2, 6, 1, 4, 3)
R> dat = worldindices[, order]
R> family = c(9, 2, 19, 2, 19, 0, 0, 34, 1, 0, 0, 1, 4, 0, 0)
```

Using the function `CDVineSeqEst` with `method = "mle"` we get the following sequential estimates of the pair-copula parameters (Step 3 of Figure 2).

```
R> seqPar = CDVineSeqEst(dat, family = family, type = 1, method = "mle")

$par
  [1] 1.1425 0.9388 1.9827 0.9631 1.1070 0.0000 0.0000 -1.0904 0.2794
 [10] 0.0000 0.0000 0.1120 1.1030 0.0000 0.0000

$par2
  [1] 0.3014 13.4803 1.1203 14.0548 0.1806 0.0000 0.0000 0.0000 0.0000
 [10] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
```

Although sequential estimation typically provides quite good parameter estimates, they can be improved by a joint MLE.

```
R> mlePar = CDVineMLE(dat, family = family, start = seqPar$par,
+   start2 = seqPar$par2, type = 1)

$par
  [1] 1.1331 0.9389 1.9961 0.9623 1.1082 0.0000 0.0000 -1.0865 0.2794
 [10] 0.0000 0.0000 0.1120 1.1049 0.0000 0.0000

$par2
  [1] 0.3136 13.4804 1.1131 14.0552 0.1757 0.0000 0.0000 0.0000 0.0000
 [10] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

$loglik
[1] 1186

$counts
function gradient
      36      36

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

`CDVineMLE` returns the parameters found, the optimized log-likelihood as well as information about the optimization. A direct comparison of the log-likelihoods using `CDVineLogLik` shows the slight improvement of the jointly estimated parameters over the sequential ones in terms of the log-likelihood.

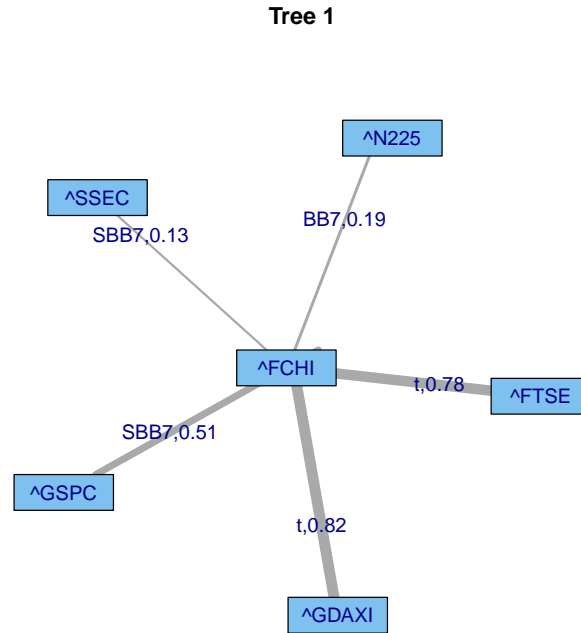


Figure 9: First tree of the specified C-vine for the `worldindices` data set with pair-copula families and Kendall's  $\tau$  values corresponding to pair-copula parameters as edge labels.

```
R> CDVineLogLik(dat, family = family, par = seqPar$par, par2 = seqPar$par2,
+             type = 1)$loglik
```

```
[1] 1185.55
```

```
R> CDVineLogLik(dat, family = family, par = mlePar$par, par2 = mlePar$par2,
+             type = 1)$loglik
```

```
[1] 1185.62
```

Finally we illustrate the C-vine trees using the function `CDVineTreePlot`. Because of limited space in this manuscript we only plot the first tree in Figure 9 (compare to Figure 1 which was produced in  $\text{\LaTeX}$ ).

```
R> P = CDVineTreePlot(data = NULL, family = family, par = mlePar$par,
+             par2 = mlePar$par2, names = colnames(dat), type = 1, tree = 1,
+             edge.labels = c("family", "theotau"))
```

Similarly we also fitted a D-vine copula model with order of variables `order_dvine`, pair-copula families `family_dvine` and corresponding parameters `par_dvine` and `par2_dvine`. In order to determine the better fitting vine copula model for the `worldindices` data set, we perform a Vuong test comparing both models (see Step 4 of Figure 2).



```

R> CDVineVuongTest(worldindices, Model1.order = order,
+   Model2.order = order_dvine, Model1.family = family,
+   Model2.family = family_dvine, Model1.par = mlePar$par,
+   Model2.par = par_dvine, Model1.par2 = mlePar$par2,
+   Model2.par2 = par2_dvine, Model1.type = 1, Model2.type = 2)

$statistic
[1] 0.2818

$statistic.Akaike
[1] 0.2818

$statistic.Schwarz
[1] 0.2818

$p.value
[1] 0.7781

$p.value.Akaike
[1] 0.7781

$p.value.Schwarz
[1] 0.7781

```

The test statistics close to zero (irrespective of the correction considered) and the large  $p$ -values indicate that the C- and the D-vine copula models for the `worldindices` data set cannot be distinguished statistically. Results from a Clarke test between both models, which are not reported here, confirm this.

To summarize, the above analysis showed strong positive dependencies among the European stock indices, where the French CAC 40 was determined to be central for explaining the overall dependence observed in the data. Further, we found evidence of medium to strong tail dependence as well as of some asymmetries in the dependence structure. Based on the data we could however not discriminate among fitted C- and D-vine copula models, where it should be noted that both models provide additional insights due to their specific structures.

## 6. Conclusion and outlook

In this paper, we present the R-package **CDVine** for statistical inference of C- and D-vine copulas and demonstrate its use and usefulness in a substantial example. For the first time, the package **CDVine** provides extensive functionality for vine copula inference and related data analysis. In the future, we are planning to extend this to the more general class of regular vines as defined in Kurowicka and Cooke (2006). Inference and model selection of these are treated in Dißmann, Brechmann, Czado, and Kurowicka (2011) and Brechmann, Czado, and Aas (2012), while a large scale financial application can be found in Brechmann and Czado (2012). Further possible extensions are Bayesian inference and model selection techniques as used in Min and Czado (2010) and Min and Czado (2011).

## 7. Acknowledgment

A first version of **CDVine** was based on and inspired by code from Daniel Berg (Norwegian Computing Center; <http://www.danielberg.no>) provided by personal communication. We further acknowledge substantial contributions by our working group at Technische Universität München, in particular by Carlos Almeida and Aleksey Min. In addition, we like to thank Shing (Eric) Fu, Feng Zhu, Guang (Jack) Yang, and Harry Joe for providing their implementation of the method by Knight (1966). We are especially grateful to Harry Joe for his contributions to the implementation of the bivariate Archimedean copulas. Numerical stability tests were performed on a Linux cluster supported by DFG grant INST 95/919-1 FUGG. Both authors gratefully acknowledge the support of the TUM Graduate School's International School of Applied Mathematics. Ulf Schepsmeier is further supported by the BMBF program "Mathematik für Innovationen in Industrie und Dienstleistungen", Eike Brechmann by a grant from Allianz Deutschland AG.

## References

- Aas K, Czado C, Frigessi A, Bakken H (2009). "Pair-Copula Constructions of Multiple Dependence." *Insurance: Mathematics and Economics*, **44**(2), 182–198.
- Akaike H (1973). "Information Theory and an Extension of the Maximum Likelihood Principle." In BN Petrov, F Csaki (eds.), *Proceedings of the Second International Symposium on Information Theory Budapest*, Akademiai Kiado, pp. 267–281.
- Bedford T, Cooke RM (2001). "Probability Density Decomposition for Conditionally Dependent Random Variables Modeled by Vines." *Annals of Mathematics and Artificial intelligence*, **32**, 245–268.
- Bedford T, Cooke RM (2002). "Vines - A New Graphical Model for Dependent Random Variables." *Annals of Statistics*, **30**, 1031–1068.
- Belgorodski N (2010). *Selecting Pair-Copula Families for Regular Vines with Application to the Multivariate Analysis of European Stock Market Indices*. Master's thesis, Technische Universität München.
- Berg D, Aas K (2009). "Models for Construction of Higher-Dimensional Dependence: A Comparison Study." *European Journal of Finance*, **15**, 639–659.
- Brechmann EC, Czado C (2012). "Risk Management with High-Dimensional Vine Copulas: An Analysis of the Euro Stoxx 50." *Submitted for publication*.
- Brechmann EC, Czado C, Aas K (2012). "Truncated Regular Vines in High Dimensions with Applications to Financial Data." *Canadian Journal of Statistics*, **40**(1), 68–85.
- Chollete L, Heinen A, Valdesogo A (2009). "Modeling International Financial Returns with a Multivariate Regime Switching Copula." *Journal of Financial Econometrics*, **7**, 437–480.
- Clarke KA (2007). "A Simple Distribution-Free Test for Nonnested Model Selection." *Political Analysis*, **15**(3), 347–363.

- Csardi G, Nepusz T (2006). “The igraph software package for complex network research.” *InterJournal, Complex Systems*, 1695. URL <http://igraph.sf.net>.
- Czado C (2010). “Pair-Copula Constructions of Multivariate Copulas.” In P Jaworski, F Durante, W Härdle, T Rychlik (eds.), *Copula Theory and Its Applications*. Springer-Verlag, Berlin.
- Czado C, Schepsmeier U, Min A (2012). “Maximum Likelihood Estimation of Mixed C-Vines with Application to Exchange Rates.” *Statistical Modelling*, **12**(3), 229–255.
- Dißmann J, Brechmann EC, Czado C, Kurowicka D (2011). “Selecting and Estimating Regular Vine Copulae and Application to Financial Returns.” *Submitted for publication*.
- Fischer M, Köck C, Schlüter S, Weigert F (2009). “An Empirical Analysis of Multivariate Copula Models.” *Quantitative Finance*, **9**(7), 839–854.
- Genest C, Favre AC (2007). “Everything You Always Wanted to Know about Copula Modeling but Were Afraid to Ask.” *Journal of Hydrologic Engineering*, **12**, 347–368.
- Genest C, Ghouli K, Rivest LP (1995). “A Semiparametric Estimation Procedure of Dependence Parameters in Multivariate Families of Distributions.” *Biometrika*, **82**, 543–552.
- Genest C, Rémillard B (2008). “Validity of the Parametric Bootstrap for Goodness-of-Fit Testing in Semiparametric Models.” *Annales de l’Institut Henri Poincaré: Probabilités et Statistiques*, **44**, 1096–1127.
- Genest C, Rivest LP (1993). “Statistical Inference Procedures for Bivariate Archimedean Copulas.” *Journal of the American Statistical Association*, **88**(423), 1034–1043.
- Genz A, Bretz F, Miwa T, Mi X, Leisch F, Scheipl F, Hothorn T (2012). *mvtnorm: Multivariate Normal and t Distributions*. R package version 0.9-9992, URL <http://CRAN.R-project.org/package=mvtnorm>.
- Heinen A, Valdesogo A (2009). “Asymmetric CAPM Dependence for Large Dimensions: The Canonical Vine Autoregressive Model.” *CORE discussion papers 2009069*, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE).
- Hobæk Haff I (2010). “Parameter estimation for pair-copula constructions.” *To appear in Bernoulli*.
- Hobæk Haff I, Aas K, Frigessi A (2010). “On the simplified pair-copula construction - Simply useful or too simplistic?” *Journal of Multivariate Analysis*, **101**(5), 1296–1310.
- Hofert M, Mächler M (2011). “Nested Archimedean Copulas Meet R: The nacopula Package.” *Journal of Statistical Software*, **39**(9), 1–20. URL <http://www.jstatsoft.org/v39/i09/>.
- Hofmann M, Czado C (2010). “Assessing the VaR of a Portfolio Using D-Vine Copula Based Multivariate GARCH Models.” *Submitted for publication*.
- Joe H (1996). “Families of  $m$ -Variate Distributions with Given Margins and  $m(m - 1)/2$  Bivariate Dependence Parameters.” In L Rüschendorf, B Schweizer, MD Taylor (eds.), *Distributions with Fixed Marginals and Related Topics*, pp. 120–141. Institute of Mathematical Statistics, Hayward.

- Joe H (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall, London.
- Joe H, Li H, Nikoloulopoulos AK (2010). “Tail Dependence Functions and Vine Copulas.” *Journal of Multivariate Analysis*, **101**(1), 252–270.
- Knight WR (1966). “A Computer Method for Calculating Kendall’s tau with Ungrouped Data.” *Journal of the American Statistical Association*, **61**(314), 436–439.
- Kojadinovic I, Yan J (2010a). “Comparison of Three Semiparametric Methods for Estimating Dependence Parameters in Copula Models.” *Insurance: Mathematics and Economics*, **47**(1), 52–63.
- Kojadinovic I, Yan J (2010b). “Modeling Multivariate Distributions with Continuous Margins Using the copula R Package.” *Journal of Statistical Software*, **34**(9), 1–20. URL <http://www.jstatsoft.org/v34/i09/>.
- Kurowicka D, Cooke RM (2006). *Uncertainty Analysis with High Dimensional Dependence Modelling*. John Wiley & Sons, Chichester.
- Kurowicka D, Joe H (2011). *Dependence Modeling: Vine Copula Handbook*. World Scientific Publishing Co., Singapore.
- Mendes BVdM, Semeraro MM, Leal RPC (2010). “Pair-Copulas Modeling in Finance.” *Financial Markets and Portfolio Management*, **24**(2), 193–213.
- Min A, Czado C (2010). “Bayesian Inference for Multivariate Copulas Using Pair-Copula Constructions.” *Journal of Financial Econometrics*, **8**(4), 511–546.
- Min A, Czado C (2011). “Bayesian Model Selection for Multivariate Copulas Using Pair-Copula Constructions.” *Canadian Journal of Statistics*, **39**(2), 239–258.
- Nelsen RB (2006). *An Introduction to Copulas*. 2nd edition. Springer-Verlag, Berlin.
- Nikoloulopoulos AK, Joe H, Li H (2012). “Vine copulas with asymmetric tail dependence and applications to financial return data.” *To appear in Computational Statistics & Data Analysis*.
- R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org>.
- Schirmacher D, Schirmacher E (2008). “Multivariate Dependence Modeling Using Pair-Copulas.” *Technical report*, Society of Actuaries: 2008 Enterprise Risk Management Symposium, April 14–16, Chicago.
- Schwarz G (1978). “Estimating the Dimension of a Model.” *The Annals of Statistics*, **6**(2), 461–464.
- Sklar A (1959). “Fonctions de Répartition à n Dimensions et Leurs Marges.” *Publications de l’Institut de Statistique de L’Université de Paris*, **8**, 229–231.

Smith M, Min A, Czado C, Almeida C (2010). “Modeling Longitudinal Data Using a Pair-Copula Decomposition of Serial Dependence.” *Journal of the American Statistical Association*, **105**(492), 1467–1479.

Vuong QH (1989). “Ratio Tests for Model Selection and Non-Nested Hypotheses.” *Econometrica*, **57**(2), 307–333.

Wuertz D, et al. (2009). *fCopulae: Rmetrics - Dependence Structures with Copulas*. R package version 2110.78, URL <http://CRAN.R-project.org/package=fCopulae>.

Yan J (2007). “Enjoy the Joy of Copulas: With a Package copula.” *Journal of Statistical Software*, **21**(4), 1–21. URL <http://www.jstatsoft.org/v21/i04/>.

## A. Code for figures

Code to generate Figure 3:

```
R> u1 = seq(0.02, 0.98, by = 0.02)
R> u2 = seq(0.02, 0.98, by = 0.02)
R> copCDF = matrix(0, 49, 49)
R> copPDF = matrix(0, 49, 49)
R> for (i in 1:49) for (j in 1:49) copCDF[i, j] = BiCopCDF(u1 = u1[i],
+   u2 = u2[j], family = 2, par = 0.7, par2 = 4)
R> for (i in 1:49) for (j in 1:49) copPDF[i, j] = BiCopPDF(u1 = u1[i],
+   u2 = u2[j], family = 2, par = 0.7, par2 = 4)
R> par(mfrow = c(1, 2), cex.axis = 0.7, cex.lab = 0.9)
R> persp(u1, u2, copCDF, theta = -25, main = "Copula CDF", ticktype = "detailed",
+   zlab = "", nticks = 9)
R> persp(u1, u2, copPDF, theta = -25, main = "Copula PDF", ticktype = "detailed",
+   zlab = "", nticks = 9)
```

Code to generate Figure 4:

```
R> dat0 = BiCopSim(N = 500, family = 3, par = BiCopTau2Par(family = 3,
+   tau = 0.5))
R> dat90 = BiCopSim(N = 500, family = 23, par = BiCopTau2Par(family = 23,
+   tau = -0.5))
R> dat180 = BiCopSim(N = 500, family = 13, par = BiCopTau2Par(family = 13,
+   tau = 0.5))
R> dat270 = BiCopSim(N = 500, family = 33, par = BiCopTau2Par(family = 33,
+   tau = -0.5))
R> par(mfrow = c(1, 4), cex.main = 3, cex.lab = 2, cex.axis = 2)
R> plot(dat0, main = "0 degrees", xlab = "u1", ylab = "u2")
R> plot(dat90, main = "90 degrees", xlab = "u1", ylab = "u2")
R> plot(dat180, main = "180 degrees", xlab = "u1", ylab = "u2")
R> plot(dat270, main = "270 degrees", xlab = "u1", ylab = "u2")
```

Code to generate Figure 5:

```

R> par(mfrow = c(1, 3), cex.main = 2, cex.lab = 1.5, cex.axis = 1.5)
R> dat = BiCopSim(N = 1000, family = 4, par = 2)
R> BiCopMetaContour(u1 = dat[, 1], u2 = dat[, 2], bw = 2, size = 100,
+   levels = c(0.01, 0.05, 0.1, 0.15, 0.2), par = 0, family = "emp",
+   main = "Empirical with N=1000")
R> BiCopMetaContour(u1 = NULL, u2 = NULL, bw = 1, size = 100,
+   levels = c(0.01, 0.05, 0.1, 0.15, 0.2), family = 4, par = 2,
+   main = "Standard normal margins")
R> BiCopMetaContour(u1 = NULL, u2 = NULL, bw = 1, size = 100,
+   family = 4, par = 2, margins = "gamma", margins.par = c(1.5,
+   0.75), levels = c(0.005, 0.01, 0.03, 0.05, 0.07, 0.09),
+   main = "Gamma margins")

```

Code to generate Figure 6:

```

R> dat = BiCopSim(N = 1000, family = 6, par = 2)
R> par(mfrow = c(1, 3), cex.main = 2, cex.lab = 1.5, cex.axis = 1.5)
R> BiCopLambda(u1 = dat[, 1], u2 = dat[, 2])
R> BiCopLambda(family = 6, par = 2)
R> BiCopLambda(u1 = dat[, 1], u2 = dat[, 2], family = 6, par = 2)

```

Code to generate Figure 7:

```

R> panel.contour <- function(x, y, bw = 2, size = 100) {
+   usr <- par("usr")
+   on.exit(par(usr))
+   par(usr = c(-3, 3, -3, 3), new = TRUE)
+   BiCopMetaContour(x, y, bw, size, axes = FALSE)
+ }
R> pairs(worldindices, lower.panel = panel.contour, gap = 0)

```

Code to generate Figure 8:

```

R> par(mfrow = c(1, 3), cex.main = 2, cex.lab = 1.5, cex.axis = 1.5)
R> BiCopKPlot(worldindices[, 5], worldindices[, 6], main = "K-plot")
R> BiCopChiPlot(worldindices[, 5], worldindices[, 6], xlim = c(-1,
+   1), ylim = c(-1, 1), main = "Chi-plot")
R> param = BiCopEst(worldindices[, 5], worldindices[, 6], 2)
R> BiCopLambda(worldindices[, 5], worldindices[, 6], family = 2,
+   par = param$par, par2 = param$par2, main = "Lambda-function")

```

## Affiliation:

Eike Christian Brechmann, Ulf Schepsmeier  
 Lehrstuhl für Mathematische Statistik

Zentrum Mathematik  
Technische Universität München  
85748 Garching b. München, Germany  
E-mail: [brechmann@ma.tum.de](mailto:brechmann@ma.tum.de), [schepsmeier@ma.tum.de](mailto:schepsmeier@ma.tum.de)  
URL: <http://www-m4.ma.tum.de/pers/>