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

Eike Christian Brechmann
Technische Universität München

Ulf Schepsmeier
Technische Universität München

Abstract

Dependence modeling using copulas is very common nowadays. While other 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. 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 recent years copula modeling has 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 a d-dimensional distribution function with margins $F_1, ..., F_d$. Then there exists a copula C such that for all $\boldsymbol{x} = (x_1, ..., x_d)' \in (\mathbb{R} \cup \{-\infty, \infty\})^d$,

$$F(\boldsymbol{x}) = C(F_1(x_1), ..., F_d(x_d)). \tag{1}$$

C is unique if $F_1, ..., F_d$ are continuous. Conversely, if C is a copula and $F_1, ..., F_d$ are distribution functions, then the function F defined by (1) is a joint distribution function with margins $F_1, ..., F_d$. 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.

While this motivates us to speak of *the* copula of continuous random variables $\mathbf{X} = (X_1, ..., X_d) \sim F$, 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 (cp. 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*. 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*.

These are derived as iterative pair-copula constructions, where the $d(d-1)/2$ pair-copulas can be arranged in $d-1$ 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. This gives the following decomposition of a multivariate density, the *C-vine density* w.l.o.g. root nodes $1, ..., d$ (otherwise nodes can be relabeled),

$$f(\boldsymbol{x}) = \prod_{k=1}^{d} f_k(x_k) \times \prod_{i=1}^{d-1} \prod_{j=1}^{d-i} c_{i,i+j|1:(i-1)}(F(x_i|x_1, ..., x_{i-1}), F(x_{i+j}|x_1, ..., x_{i-1})|\boldsymbol{\theta}_{i,i+j|1:(i-1)}), \tag{2}$$

where $f_k,\ k = 1, ..., d$, 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 $i_k, ..., i_m$). 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 $i = 1, ..., d-1$.

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 $1, ..., d$, 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. 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 $1, ..., d$; otherwise nodes can be relabeled):

$$f(\boldsymbol{x}) = \prod_{k=1}^{d} f_k(x_k) \times$$
$$\prod_{i=1}^{d-1} \prod_{j=1}^{d-i} c_{j,j+i|(j+1):(j+i-1)}(F(x_j|x_{j+1}, ..., x_{j+i-1}), F(x_{j+i}|x_{j+1}, ..., x_{j+i-1})|\boldsymbol{\theta}_{j,j+i|(j+1):(j+i-1)}). \tag{3}$$

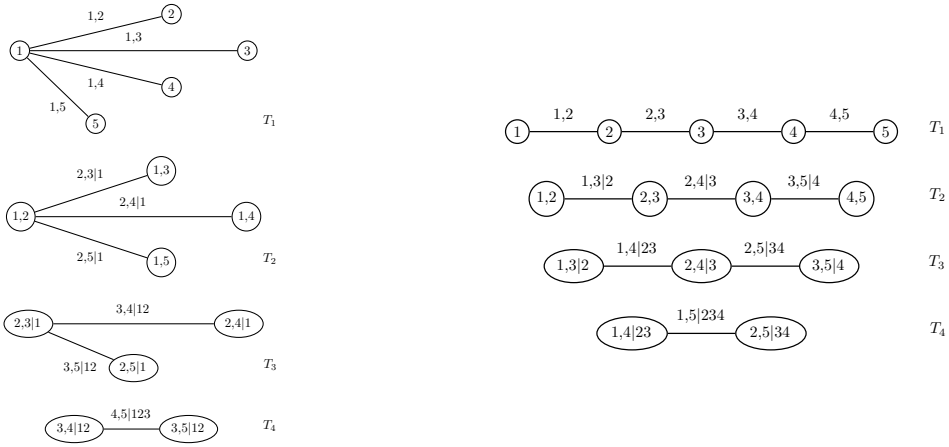

Figure 1: Examples of five-dimensional C- (left panel) and D-vine trees (right panel) with edge indices.

Again the outer product runs over the $d-1$ trees, while the pairs in each tree are designated by the inner product.

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

$$h(x|\boldsymbol{v}, \boldsymbol{\theta}) := F(x|\boldsymbol{v}) = \frac{\partial C_{xv_j|\boldsymbol{v}_{-j}}(F(x|\boldsymbol{v}_{-j}), F(v_j|\boldsymbol{v}_{-j})|\boldsymbol{\theta})}{\partial F(v_j|\boldsymbol{v}_{-j})}, \tag{4}$$

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

By allowing arbitrary bivariate copulas for each pair-copula term in the decompositions (2) and (3), the multivariate copulas obtained from C- and D-vine structures, so-called *C-* 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 (cp. Joe, Li, and Nikoloulopoulos 2010). 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 $1, ..., 5$, which also is the order of the first D-vine tree. Edge labels show the indices of the corresponding pair-copula terms.

Since Aas *et al.* (2009), C- and D-vine copulas have been very successful in many applications, see, e.g., Schirmacher and Schirmacher (2008), Chollete, Heinen, and Valdesogo (2009), Heinen and Valdesogo (2009), Mendes, Semeraro, and Leal (2010), and Czado, Schepsmeier, and Min (2011) as well as Min and Czado (2010), Min and Czado (2011), Smith, Min, Czado, and Almeida (2010), and Hofmann and Czado (2010) who take a Bayesian approach. Comparison studies of multivariate copulas showing the good performance of vine copulas are Berg and Aas (2009) and Fischer, Köck, Schlüter, and Weigert (2009). 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 packages **fCopulae** (Wuertz *et al.* 2009) and **QRMlib** (McNeil and Ulman 2010) and finally the package **nacopula** (Hofert and Maechler 2010) for so-called nested Archimedean copulas, a generalization of Archimedean copulas. Furthermore, our package depends on the packages **igraph** (Csardi 2010) for illustrations of vine trees and **mvtnorm** (Genz *et al.* 2011), 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.

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

2. 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 3.

In the following we further assume that the data we are working with is in $[0, 1]$ and has approximately uniform margins, so-called *copula data*. For general data sets this is typically established either by non-parametrically transforming the data with the empirical marginal distribution functions or by choosing (and fitting) appropriate marginal distributions and then applying the parametric distribution functions to the data (cp. Sklar's Theorem (1)).

To allow for reproducibility of the results, we preliminarily fix a seed.

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

2.1. Bivariate copula families

The **CDVine** package provides a wide range of bivariate copula families from the two major classes of elliptical and Archimedean copulas (cp. 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

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

and the bivariate Student-t copula

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

with dependence parameter $\rho \in (-1, 1)$ and degrees of freedom parameter $\nu > 1$ for the Student-t copula. Φ_ρ denotes the bivariate standard normal distribution function with correlation parameter ρ and Φ^{-1} the inverse of the univariate standard normal distribution function. Similarly, $t_{\rho,\nu}$ is the bivariate Student-t distribution function with correlation parameter ρ and ν degrees of freedom, while t_ν^{-1} denotes the inverse univariate Student-t distribution function with ν 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

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

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

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

φ 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 δ-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 τ 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

No.	Elliptical distribution	Parameter range	Kendall's τ	Tail dependence
1	Gaussian	$\rho \in (-1, 1)$	$\frac{2}{\pi}\arcsin(\rho)$	0
2	Student-t	$\rho \in (-1, 1),\ \nu > 1$	$\frac{2}{\pi}\arcsin(\rho)$	$2t_{\nu+1}\left(-\sqrt{\nu+1}\sqrt{\frac{1-\rho}{1+\rho}}\right)$

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

No.	Name	Generator function	Parameter range	Kendall's τ	Tail dependence (lower, upper)
3	Clayton	$\frac{1}{\theta}(t^{-\theta}-1)$	$\theta > 0$	$\frac{\theta}{\theta+2}$	$(2^{-1/\theta},\ 0)$
4	Gumbel	$(-\log t)^{\theta}$	$\theta \geq 1$	$1-\frac{1}{\theta}$	$(0,\ 2-2^{1/\theta})$
5	Frank[a]	$-\log[\frac{e^{-\theta t}-1}{e^{-\theta}-1}]$	$\theta \in \mathbb{R}\setminus\{0\}$	$1-\frac{4}{\theta}+4\frac{D_1(\theta)}{\theta}$	$(0,\ 0)$
6	Joe[b]	$-\log[1-(1-t)^{\theta}]$	$\theta > 1$	$\frac{2\theta-4+2\gamma+2\log 2+\Psi(\frac{1}{\theta})+\Psi(\frac{2+\theta}{2\theta})}{\theta-2}$	$(0,\ 2-2^{1/\theta})$
7	BB1	$(t^{-\theta}-1)^{\delta}$	$\theta > 0,\ \delta \geq 1$	$1-\frac{2}{\delta(\theta+2)}$	$(2^{-1/(\theta\delta)},\ 2-2^{1/\delta})$
8	BB6	$(-\log[1-(1-t)^{\theta}])^{\delta}$	$\theta \geq 1, \delta \geq 1$	$1+4\int_0^1\big(-\log(-(1-t)^{\theta}+1)\times\frac{(1-t-(1-t)^{-\theta}+t(1-t)^{-\theta})}{\delta\theta}\big)dt$	$(0, 2-2^{1/(\theta\delta)})$
9	BB7[c]	$[1-(1-t)^{\theta})]^{-\delta}-1$	$\theta \geq 1,\ \delta > 0$	$1-\frac{2}{\delta(2-\theta)}+\frac{4}{\theta^2\delta}B(\frac{2-\theta}{\theta},\delta+2)$	$(2^{-1/\delta},\ 2-2^{1/\theta})$
10	BB8	$-\log\left[\frac{1-(1-\delta t)^{\theta}}{1-(1-\delta)^{\theta}}\right]$	$\theta \geq 1, 0 < \delta \leq 1$	$1+4\int_0^1\big(-\log\left(\frac{(1-t\delta)^{\theta}-1}{(1-\delta)^{\theta}-1}\right)\times\frac{1-t\delta-(1-t\delta)^{-\theta}+t\delta(1-t\delta)^{-\theta}}{\theta\delta}\big)dt$	(0,0[d])

Table 2: Denotation and properties of bivariate Archimedean copula families included in **CDVine**.

[a] $D_1(\theta)=\int_0^{\theta}\frac{c/\theta}{\exp(x)-1}dx$ (Debey function)
[b] $\gamma=\lim_{n\to\infty}(\sum_{i=1}^{n}\frac{1}{i}-\log n)\approx 0.57721$ (Euler's constant), $\Psi(x)=\frac{d}{dx}\log(\Gamma(x))$ (Digamma function)
[c] $B(x,y)=\int_0^1 t^{x+1}(t-1)^{y-1}dt$ (Beta function)
[d] Except for $\delta=1$, then the upper tail dependence coefficient is $2-2^{1/\theta}$.

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 270 degrees are on the negative scale (cp. 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 $\theta \to 0$. As a reminder of the coding of the copula families the function `BiCopName` transforms a copula

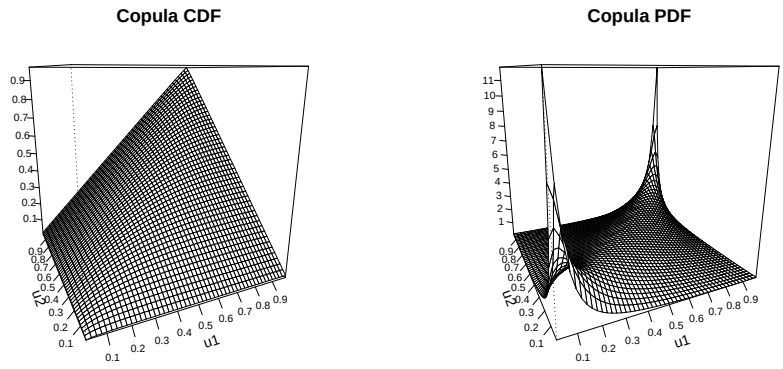

Figure 2: CDF and PDF of a bivariate Student-t copula with dependence parameter $\rho = 0.7$ and 4 degrees of freedom.

family name to its number analogue and vice versa.

The cumulative distribution functions (CDF's) and probability density functions (PDF's) 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 2.

```
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)
```

Conditional bivariate distribution functions, the so-called h-functions defined in (4), 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` in the first and of `u1` given `u2` in the second argument. The function call for a bivariate Frank copula (`family = 5`) with parameter $\theta = 6$ (`par = 6`) is as follows, where $u_1 = 0.7$ and $u_2 = 0.4$ and first $h(u_2|u_1, \boldsymbol{\theta})$ (`hfunc1`) and then $h(u_1|u_2, \boldsymbol{\theta})$ (`hfunc2`) are returned.

```
R> BiCopHfunc(u1 = 0.7, u2 = 0.4, family = 5, par = 6)

$hfunc1
[1] 0.1338
```

```
$hfunc2
[1] 0.8771
```

To account for the relationship between bivariate copula parameter(s) and Kendall's τ and vice versa, the package **CDVine** contains the functions `BiCopPar2Tau` and `BiCopTau2Par`. However note that the inverse relationship (Kendall's τ 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. The following code calculates the Kendall's τ corresponding to a bivariate Gaussian copula with parameter $\rho = 0.7$ and vice versa.

```
R> tau1 = BiCopPar2Tau(family = 1, par = 0.7)

[1] 0.4936

R> BiCopTau2Par(family = 1, tau = tau1)

[1] 0.7
```

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`. The usage of this function for a BB1 copula with parameters $\theta = 0.8$ and $\delta = 1.5$ is as follows:

```
R> BiCopPar2TailDep(family = 7, par = 0.8, par2 = 1.5)

$lower
[1] 0.5612

$upper
[1] 0.4126
```

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 uniform samples. Then $\boldsymbol{u} = (u_1, u_2)'$ 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 (4), 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 τ values of 0.5 for positive dependence (`family = 3` and `13`) and -0.5 for negative dependence (`family = 23` and `33`).

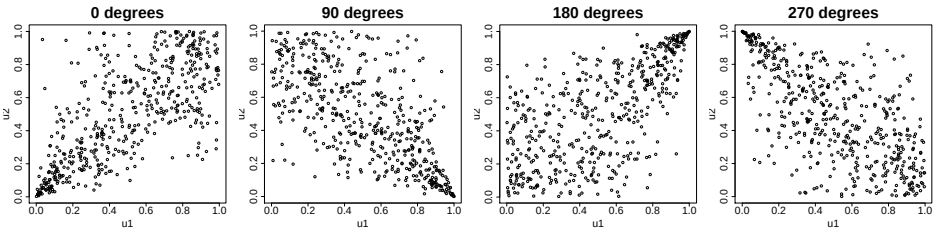

Figure 3: Samples from Clayton copulas rotated by 0, 90, 180 and 270 degrees with parameters corresponding to Kendall's τ values of 0.5 for positive dependence and -0.5 for negative dependence.

```
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))
```

Corresponding scatter plots are shown in Figure 3.

2.2. 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. **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 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.

The following example 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 $\theta = 2$, where no data is needed for the theoretical contour plots. The additional arguments `bw`, `size` and `levels` define the bandwidth, number of grid points and contour levels used. The resulting plots are shown in Figure 4.

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`.

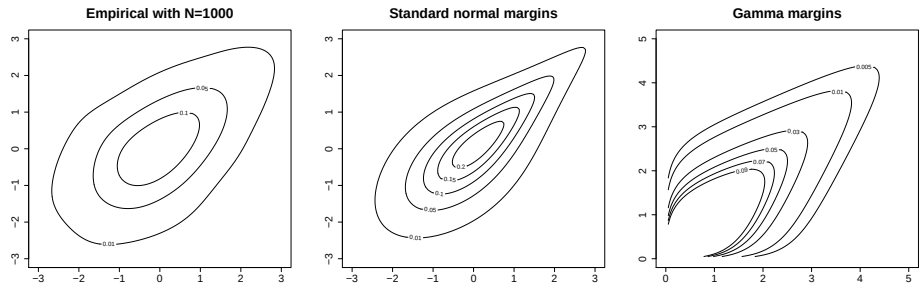

Figure 4: Left panel: empirical contour plot with standard normal margins for simulated data (`N = 1000`) of a Gumbel copula with parameter $\theta = 2$. Middle and right panel: meta Gumbel copula distribution with standard normal and Gamma margins with shape parameter 1.5 and scale parameter 0.75. The Gumbel copula parameter is $\theta = 2$.

```
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")
```

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 χ-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 7 of Section 4.

Genest and Rivest (1993) introduced a further method—the λ-function. The λ-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 with uniform margins. For

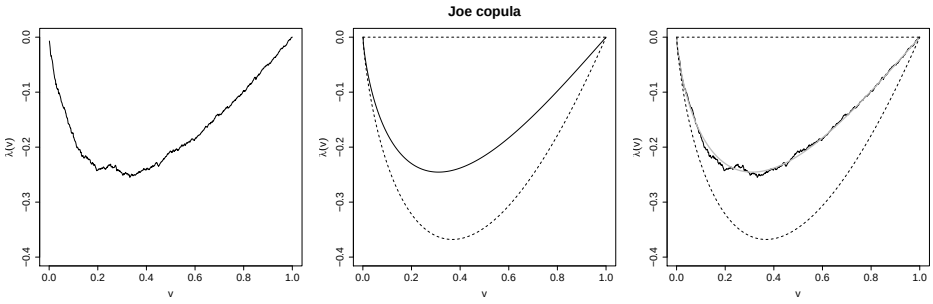

Figure 5: Left panel: empirical λ-function for simulated data (`N = 1000`) of a Joe copula with parameter $\theta = 2$. Middle panel: theoretical λ-function of a Joe copula with parameter $\theta = 2$. Right panel: both plots combined. The dashed lines in the two rightmost panels are bounds corresponding to independence and comonotonicity ($\lambda = 0$), respectively.

Archimedean copulas the λ-function is explicitly given in terms of the generator function φ and its derivative φ' (see Genest and Rivest 1993, for more details):

$$\lambda(v, \boldsymbol{\theta}) = \frac{\varphi(v)}{\varphi'(v)}.$$

In `BiCopLambda` we implemented the λ-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 λ-functions exist. Therefore they are simulated based on samples of size 1000. The plot of the theoretical λ-function also shows the bounds of the λ-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 λ-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 λ-functions of the corresponding non-rotated copula families can be considered.

Comparing empirical to theoretical λ-functions gives an indication which copula family might be appropriate to describe the observed dependence (cp. Section 4). An illustrative example for the Joe copula with parameter $\theta = 2$ is shown here: we first produce a plot of the empirical λ-function, then of the theoretical one, and finally a plot showing both (see Figure 5).

```
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)
```

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 τ. 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 τ of the data. The 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 (cp. the discussion of the λ-function above). Using the simulated data of the Joe copula from above we get:

```
R> gof = BiCopGofKendall(u1 = dat[, 1], u2 = dat[, 2], family = 5)
R> gof$p.value.CvM

[1] 0

R> gof$p.value.KS

[1] 0
```

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 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 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 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 models, which are non-nested. 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 $\hat{\boldsymbol{\theta}}_1$ and

$\hat{\boldsymbol{\theta}}_2$, respectively. For the Vuong test we then compute the standardized sum, ν, of the log differences of their pointwise likelihoods $m_i := \log\left[\frac{c_1(u_{i,1},u_{i,2}|\hat{\boldsymbol{\theta}}_1)}{c_2(u_{i,1},u_{i,2}|\hat{\boldsymbol{\theta}}_2)}\right]$ for observations $u_{i,j},\ i = 1,...,N,\ j = 1,2$, i.e.,

$$\nu = \frac{\frac{1}{n}\sum_{i=1}^{N} m_i}{\sqrt{\sum_{i=1}^{N}(m_i - \bar{m})^2}}. \tag{5}$$

Vuong (1989) showed that ν is asymptotically standard normal. We hence prefer copula model 1 to copula model 2 at level α 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 ($H_0 :\ E(m_i) = 0\ \forall i = 1,..,N$).

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

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

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

$$B = \sum_{i=1}^{N} \mathbf{1}_{(0,\infty)}(m_i), \tag{6}$$

where $\mathbf{1}$ is the indicator function, was proposed by Clarke (2007) and is 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 (5) and (6) 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 the Section 4. For given marginally uniform data `u1` and `u2` and a set of copula families to compare, as specified by the argument `familyset`, the function call is as follows:

```
R> BiCopVuongClarke(u1, u2, familyset, correction, level)
```

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 (cp. the discussion of BiCopEst in Section 2.3) and then selects the family based on the AIC (default) or the BIC. Furthermore, a preliminary independence test (cp. 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).

We use one more time the simulated data of the Joe copula from above to select among all implemented bivariate copula families (familyset = NA) with a preliminary independence test (indeptest = TRUE) at significance level 5% (level = 0.05). The function returns the selected bivariate copula family (family) and the corresponding estimated copula parameters (par and par2).

```
R> cop1 = BiCopSelect(u1 = dat[, 1], u2 = dat[, 2], familyset = NA,
+      indeptest = TRUE, level = 0.05)
R> cop1$family

[1] 6

R> cop1$par

[1] 1.967
```

2.3. 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. This can be established using the function BiCopEst which performs either a method of moments (inversion of Kendall's τ (method = "itau"); compare Tables 1 and 2 and the function BiCopTau2Par) or maximum likelihood estimation (MLE; method = "mle"). Note again that the inversion of Kendall's τ 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 τ, while optimization is performed using the L-BFGS-B algorithm for constraint optimization to account for the parameter ranges (cp. 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 τ 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 when running the following example with simulated data from a Gaussian copula with parameter $\rho = 0.7$.

```
R> dat = BiCopSim(N = 1000, family = 1, par = 0.7)
R> BiCopEst(u1 = dat[, 1], u2 = dat[, 2], family = 1, method = "mle",
+      se = TRUE)

$par
[1] 0.6949

$par2
[1] 0

$se
[1] 0.01343

$se2
[1] 0

R> BiCopEst(u1 = dat[, 1], u2 = dat[, 2], family = 2, method = "mle",
+      se = TRUE, max.df = 30)

$par
[1] 0.6946

$par2
[1] 30

$se
[1] 0.014

$se2
[1] NA

Warning:
In MLE_intern(cbind(u1, u2), c(theta1, delta), family = family,  :
  Degrees of freedom of the t-copula estimated to be larger than 30.
  Consider using the Gaussian copula instead.
```

3. 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.

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

As discussed in the introduction (Section 1), 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 **CDVine** package 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

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

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

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

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 τ value (cp. `BiCopPar2Tau`):

```
R> CDVinePar2Tau(family = family, par = par, par2 = par2)

[1]  0.3333  0.3939  0.3763 -0.1940  0.1282  0.1837
```

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 2.1. As an example we simulate `N = 500` samples from the four-dimensional C-vine copula model defined above.

```
R> dat = CDVineSim(N = 500, family = family, par = par, par2 = par2,
+      type = type)
R> head(dat)

        [,1]    [,2]    [,3]    [,4]
[1,] 0.50748 0.33433 0.55270 0.54163
[2,] 0.08514 0.09035 0.10338 0.09657
[3,] 0.61583 0.49753 0.72094 0.60271
[4,] 0.11351 0.34688 0.12842 0.23596
[5,] 0.05190 0.08701 0.08575 0.50173
[6,] 0.86472 0.78956 0.83610 0.78675
```

3.2. 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 2.2 or using the function `CDVineCopSelect`. 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. The function call is:

```
R> CDVineCopSelect(data, familyset, type, selectioncrit, indeptest,
+      level)
```

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 1). 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.* (2011) for a detailed description of sequential estimation in C-vines). This can be established using the function `CDVineSeqEst` which internally calls the function `BiCopEst` described in Section 2.3. Therefore, estimation can be carried out using inversion of Kendall's τ 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`. For a given marginally uniform data set (`data`) and pair-copula families (`family`) specified as described above, the function is then called as follows:

```
R> CDVineSeqEst(data, family, type, method, se, max.df, max.BB)
```

A detailed example will be given in Section 4. Note that the estimated parameters returned by `CDVineCopSelect` correspond to sequential estimates obtained by bivariate MLE for the parameter of each pair-copula.

Even though these sequential estimates often provide a good fit, one typically is interested in maximizing the (log-)likelihood of a vine copula specification (cp. (2) and (3)) for observations $\boldsymbol{u} = (u_{k,j})_{k=1,\dots,N,\ j=1,\dots,d}$:

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

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

  where $F_{j|i_1:i_m} := F(u_{k,j}|u_{k,i_1}, ..., u_{k,i_m})$ 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 log-likelihood with parameter set $\boldsymbol{\theta}_{DV}$ is:

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

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).

```
R> CDVineLogLik(data, family, par, par2, type)
```

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`. The function call is then as follows:

```
R> CDVineMLE(data, family, start, start2, type, maxit, max.df, max.BB)
```

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 found in Aas *et al.* (2009), Hobæk Haff (2010) and Czado *et al.* (2011).

3.3. 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. Besides the classical AIC and BIC, implemented in `CDVineAIC` and `CDVineBIC`, two such criteria are the Vuong and the Clarke tests described in Section 2.2. 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.

```
R> CDVineVuongTest(data, Model1.order, Model2.order, Model1.family,
+      Model2.family, Model1.par, Model2.par, Model1.par2, Model2.par2,
+      Model1.type, Model2.type)
```

`CDVineClarkeTest` is called similarly.

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 $\{1, ..., d-1\}$). 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 τ values corresponding to pair-copula families and parameters, or
- `"emptau"`: empirical Kendall's τ 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`.

```
R> CDVineTreePlot(data, family, par, par2, names, type, method,
+      max.df, max.BB, tree, edge.labels, P)
```

3.4. Implementation and numerical issues

In order to speed up computations we impemented 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 2.3, 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 τ, that is Kendall's τ 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 (cp. 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.

4. 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 6 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.

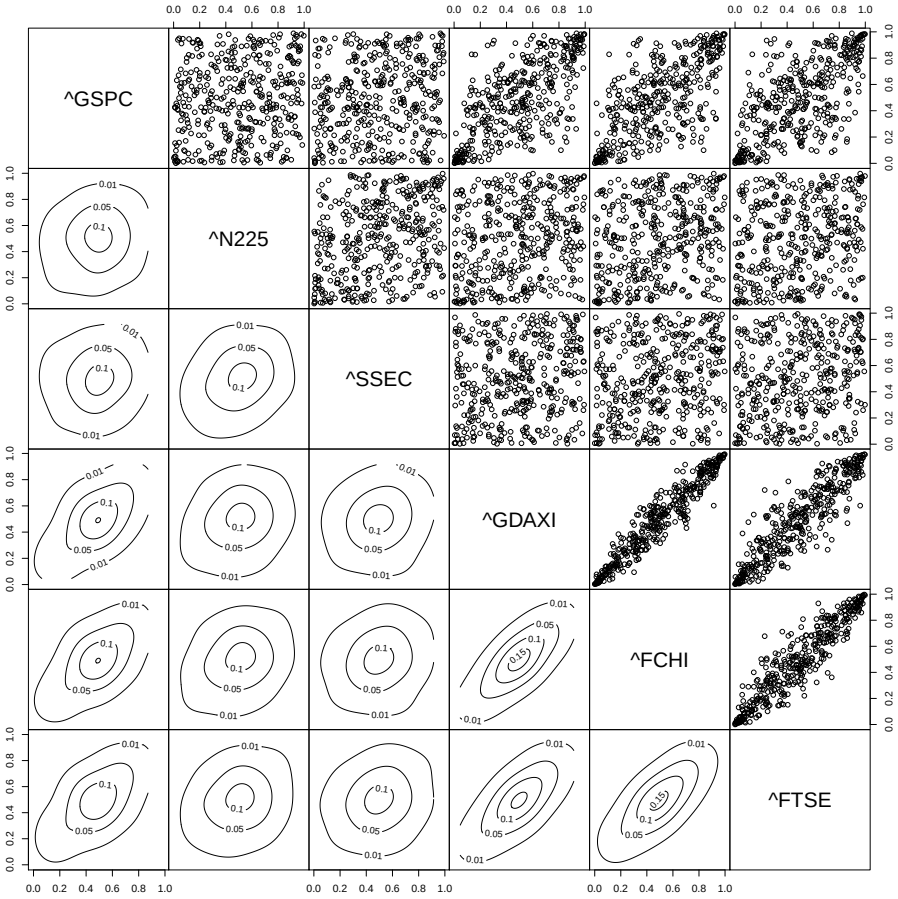

Figure 6: Pairs plot of the `worldindices` data set with scatter plots above and contour plots with standard normal margins below the diagonal. Axes of the contour plots range from -3 to 3 other than indicated here. (Compare the documentation of the R-function `pairs` for details on how to obtain such a plot.)

To illustrate the usefulness of our functions and their handling 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. 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.* (2011) we determine ^FCHI as the first root node (C-vine tree with strongest dependencies in terms of absolute empirical values of pairwise Kendall's τ's). We now exemplarily show the EDA for the pair (^FCHI,^FTSE) using the graphical tools `BiCopMetaContour` (see row 6, column 5 of Figure 6), `BiCopKPlot`, `BiCopChiPlot` and `BiCopLambda`, and the analytical tools `BiCopIndepTest`, `BiCopVuongClarke` and `BiCopSelect` in order to choose the best fitting copula.

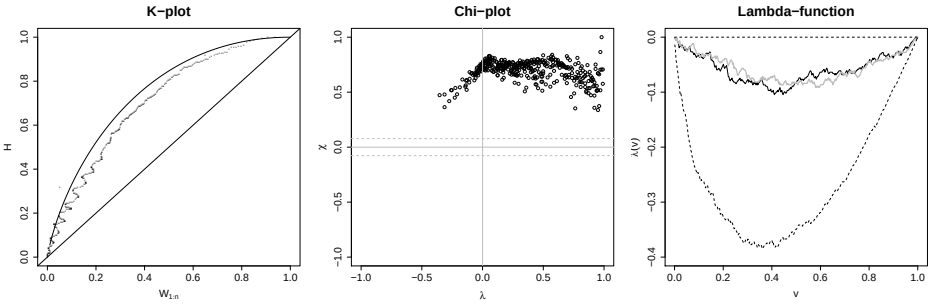

Figure 7: Left panel: K-plot. Middle panel: chi-plot. Right panel: empirical λ-function (black line), theoretical λ-function of a Student-t copula with parameters estimated using `BiCopEst` (grey line) as well as independence and comonotonicity limits (dashed lines).

```
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")
```

The scatter plot in row 5, column 6 of Figure 6 as well as the K- and chi-plots in Figure 7 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 6. The λ-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
```

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 ^FCHI). Based on these pair-copula families and the according estimated parameters, one can then use h-functions (4) 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.* (2011) the root node order in the data set is determined as ^FCHI, ^N225, ^GDAXI, ^SSEC and finally ^GSPC. Copula families (according to this order) are chosen as 9 (c_{12}), 2 (c_{13}), 2 (c_{14}), 19 (c_{15}), 19 (c_{16}), 0 ($c_{23|1}$), 34 ($c_{24|1}$), 1 ($c_{25|1}$), 0 ($c_{26|1}$), 0 ($c_{34|12}$), 1 ($c_{35|12}$), 0 ($c_{36|12}$), 0 ($c_{45|123}$), 4 ($c_{46|123}$), 0 ($c_{56|1234}$), where bivariate independence tests have been used to identify possibly independent conditional variable pairs.

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

Using the function `CDVineSeqEst` with `method = "mle"` we get the following sequential estimates of the pair-copula parameters.

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

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

$par2
 [1]  0.3014 13.4802 14.0548  0.1806  1.1203  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  0.9624  1.1081  1.9961  0.0000 -1.0864  0.2794  0.0000
[10]  0.0000  0.1120  0.0000  0.0000  1.1049  0.0000
```

```
$par2
 [1]  0.3136 13.4803 14.0551  0.1757  1.1132  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
      35       35

$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.

```
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 8 (cp. to Figure 1 which was produced in 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.

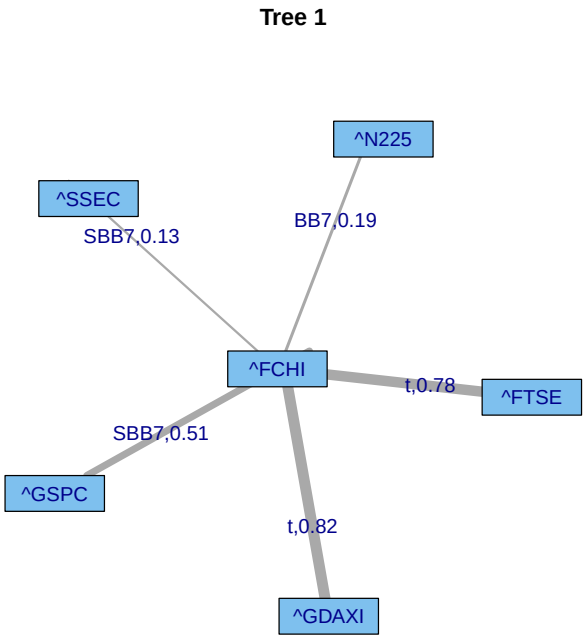

Figure 8: First tree of the specified C-vine for the `worldindices` data set with pair-copula families and Kendall's τ values corresponding to pair-copula parameters as edge labels.

```
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 shown 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.

5. 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 **CDVine** package 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 (2010), while a large scale financial application can be found in Brechmann and Czado (2011). Further possible extensions are Bayesian inference and model selection techniques as used in Min and Czado (2010) and Min and Czado (2011).

6. 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 princi-

ple." 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 (2011). "Extending the CAPM using pair copulas: The Regular Vine Market Sector model." *Submitted for publication*.

Brechmann EC, Czado C, Aas K (2010). "Truncated regular vines and their applications." *Submitted for publication*.

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 (2010). *igraph: Network analysis and visualization*. R package version 0.5.5-1, URL http://CRAN.R-project.org/package=igraph.

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, Berlin.

Czado C, Schepsmeier U, Min A (2011). "Maximum likelihood estimation of mixed C-vines with application to exchange rates." *To appear in Statistical Modelling*.

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, Ghoudi 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, *et al.* (2011). *mvtnorm: Multivariate Normal and t Distributions*. R package version 0.9-96, 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). "Estimating the parameters of a pair-copula construction." *Submitted for publication*.

Hofert M, Maechler M (2010). *nacopula: Nested Archimedean Copulas*. R package version 0.4-3, URL http://CRAN.R-project.org/package=nacopula.

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, Chichester.

Kurowicka D, Joe H (2011). *Dependence Modeling: Vine Copula Handbook*. World Scientific Publishing Co., Singapore.

McNeil A, Ulman S (2010). *QRMlib: Provides R-language code to examine Quantitative Risk Management concepts*. R package version 1.4.5, URL http://CRAN.R-project.org/package=QRMlib.

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, Berlin.

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 Acturaries: 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/.

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, schepsmeier@ma.tum.de
URL: http://www-m4.ma.tum.de/pers/