Selecting and estimating regular vine copulae and application to financial returns

J. Dißmann[a], E. C. Brechmann[a,*], C. Czado[a], D. Kurowicka[b]

[a] *Center for Mathematical Sciences, Technische Universität München, Boltzmannstr. 3, 85747 Garching, Germany.*
[b] *Department of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, Netherlands.*

Abstract

Regular vine distributions which constitute a flexible class of multivariate dependence models are discussed. Since multivariate copulae constructed through pair-copula decompositions were introduced to the statistical community, interest in these models has been growing steadily and they are finding successful applications in various fields. Research so far has however been concentrating on so-called canonical and D-vine copulae, which are more restrictive cases of regular vine copulae. It is shown how to evaluate the density of arbitrary regular vine specifications. This opens the vine copula methodology to the flexible modeling of complex dependencies even in larger dimensions. In this regard, a new automated model selection and estimation technique based on graph theoretical considerations is presented. This comprehensive search strategy is evaluated in a large simulation study and applied to a 16-dimensional financial data set of international equity, fixed income and commodity indices which were observed over the last decade, in particular during the recent financial crisis. The analysis provides economically well interpretable results and interesting insights into the dependence structure among these indices.

Keywords: minimum spanning tree, model selection, multivariate copula, regular vines

1. Introduction

The most popular statistical dependence model is the multivariate Gaussian distribution. However there is a growing demand for non-Gaussian models especially in finance (Cherubini et al., 2004) but also in climate research (e.g., Schölzel and Friederichs (2008)), environmental sciences (Salvadori et al. (2007) and Kazianka and Pilz (2011)), medicine (e.g., Beaudoin and Lakhal-Chaieb (2008)) and physics (e.g., Sato et al. (2010)) to name a few areas. With the

*Corresponding author. E-mail: brechmann@ma.tum.de. Phone: +49 89 289 17439. Fax: +49 89 289 17435.

availability of large samples of multivariate data it is possible to investigate non-Gaussian dependency models and to estimate parameters efficiently. The backbone for such models is the famous theorem by Sklar (1959), which allows to construct general multivariate distributions from copulae and marginal distributions. The specification of the copula can be done independently from the margins. While there is a multitude of bivariate copulae (see the books of Joe (1997) and Nelsen (2006)), the class of multivariate copulae was quite restricted until recently. Especially two copula classes received attention, the class of elliptical copulae (Fang et al. (2002), Frahm et al. (2003)) and the class of Archimedean copulae (Nelsen, 2005). Typical elliptical copulae are the symmetric Gaussian and Student-t copulae (see for example Demarta and McNeil (2005)), while the class of Archimedean copulae includes the tail-asymmetric Clayton and Gumbel copulae.

For financial applications a flexible modeling of tails is vital to assess the most common risk measure Value-at-Risk (VaR) (for a definition see McNeil et al. (2005)). In particular the Gaussian copula does not allow for heavy tails and the approach suggested by Li (2000) was blamed by many for contributing to the recent financial crisis (see Salmon (2009)). This shows that there is a growing need for more flexible copulae. While the Student-t copula allows for symmetric tail dependence as measured by the tail dependence coefficient or tail dependence function (see for example Joe et al. (2010)) it has only a single parameter to control tail dependence of all pairs of variables. Standard Archimedean multivariate copulae may be tail-asymmetric, but are governed only by a single parameter. There has been effort to extend the class of Archimedean copulae (see Joe (1997), Savu and Trede (2010), and Hofert (2011)), however these models require additional parameter restrictions.

These problems were noted by Aas et al. (2009), who started to utilize a wider class of multivariate copulae. This class is constructed using only bivariate copula specifications as dependency models for the distribution of certain pairs of variables conditional on a specified set of variables. These independent building blocks are called pair-copulae and were used to construct multivariate distributions. This approach dates back to Joe (1996) and was investigated and organized systematically by Bedford and Cooke (2001, 2002). The identification of the needed pairs of variables and their corresponding set of conditioning variables is facilitated by a sequence of trees (see for example Chapter 4 of Kurowicka and Cooke (2006)). They called these trees regular vines (R-vines) and the corresponding multivariate distribution an R-vine distribution. For an n-dimensional R-vine distribution, the first tree identifies $n-1$ pairs of variables, whose distribution is modeled directly. The second tree identifies $n-2$ pairs of variables, whose distribution conditional on a single variable is modeled by a pair-copula. The conditioning variable is also determined in the second tree. The next tree again identifies pairs of variables, whose conditional distribution is specified by a pair-copula. Here the conditioning set has dimension 2 and is also determined. Proceeding in this way the last tree determines a single pair of variables, whose distribution conditional on all remaining variables is defined by a last pair-copula. Recent developments and applications are discussed

in Kurowicka and Joe (2011). Czado (2010) provides a current survey about these statistical model classes and Joe et al. (2010) investigate and discuss tail dependence properties of vine distributions.

Aas et al. (2009) popularized two subclasses of regular vines, canonical vines (C-vines) and drawable vines (D-vines). C-vines possess star structures in their tree sequence, while D-vines have path structures. Kurowicka and Cooke (2006) focused on vine distributions with Gaussian pair-copulae, but Aas et al. (2009) allowed for different pair-copula families, such as the bivariate Student-t copula, bivariate Gumbel and bivariate Clayton copula. While D-vine based models are started to be used in many applications (Fischer et al. (2009), Min and Czado (2010), Chollete et al. (2009), Hofmann and Czado (2010), Mendes et al. (2010), Salinas-Gutiérrez et al. (2010), Erdorf et al. (2011), Mercier and Frison (2009), Smith et al. (2010)), C-vines are less commonly used (Heinen and Valdesogo (2009), Czado et al. (2010)); Nikoloulopoulos et al. (2012) consider both classes.

Estimation in C- and D-vine copula models is often facilitated using maximum likelihood. Since this will require optimization with respect to at least $n(n-1)/2$ parameters, it is important to provide good starting values for the optimization. For this purpose a fast sequential estimation procedure was suggested and implemented in Aas et al. (2009), whose asymptotic properties are investigated in Hobæk Haff (2011). Since bootstrapping or inversion of high dimensional Hessian matrices are required to obtain interval estimates, Bayesian approaches have been followed for parameter estimation (Min and Czado, 2010) and pair-copula selection in specified D-vine copula models (Min and Czado (2011) and Smith et al. (2010)).

However the class of R-vine distributions is much larger than the class of D- and C-vine distributions and currently there are very few applications of R-vines. One reason for this is the enormous number of possible R-vine tree sequences (see Morales-Nápoles et al. (2010)) to choose from. The importance of a good selection choice has also been noted by Garcia and Tsafack (2009). This provides the starting point of this paper. We develop an automated strategy of jointly searching for an appropriate R-vine tree structure, the pair-copula families and the parameter values of the chosen pair-copula families. It is a sequential approach starting by identifying the first tree, its pair-copula families and estimating their parameters. Based on this the specification of the second tree utilizes transformed variables. The applied transformations depend on the choices made in the first tree. In this manner all trees together with their choice of pair-copula families and corresponding parameters are made. For each tree selection we use a maximum spanning tree algorithm, where edge weights are chosen appropriately to reflect large dependencies. Pair-copulae are chosen independently. Here we use the Akaike information criterion (Akaike, 1973), which performs well in this context (see Brechmann (2010, Chapter 5)). Finally the corresponding pair-copula parameter estimation follows the same sequential estimation approach as suggested for D- and C-vine copula distributions in Aas et al. (2009).

With this automated search strategy we identify for multivariate data on the n-dimensional cube $[0,1]^n$ useful multivariate copula models, as we show

in a large simulation study and meaningful models arise for the application considered later.

Once an appropriate R-vine distribution is found for a data set we perform maximum likelihood estimation for the parameters using the sequential estimates as starting values. We also like to perform this task in an automated setup. This requires an efficient storage of the R-vine tree specification, its pair-copula families and the corresponding parameters. This is facilitated in a set of lower triangular arrays and we proof how the corresponding joint density making up the likelihood can be evaluated recursively. This setup is also used to provide an algorithm for simulating from an R-vine distribution. Pseudo code for the corresponding algorithms is given.

Finally we like to note that the developed search strategies are able to work not only in an automated fashion but also for higher dimensional problems. Before full maximum likelihood estimation was implemented for problems in at most 10 dimension. In our 16-dimensional application to financial data we show the usefulness of our approach and demonstrate that R-vine distributions provide better fit than C- and D-vines for this data set. These results have already spawned new research on finding more parsimonious specifications, which replace higher pair-copulae by independence copulae. See Brechmann et al. (2012) for details. This allows us to extend the implementation to higher dimensions, which are especially needed for the risk assessment of larger financial portfolios.

To summarize, our contributions: We develop novel algorithms for evaluating an R-vine density and simulating from specified R-vines. That is we effectively provide statistical inference techniques for R-vines. We further propose an innovative R-vine selection and estimation method and thus, for the first time, allow to actually *select* and *fit* arbitrary non-Gaussian R-vines to data. This is exploited to analyze the returns of important financial indices.

The paper is organized as follows: Section 2 introduces R-vine distributions and copulae. Necessary background from graph theory can be found in Diestel (2006). Then the efficient storage of the R-vine specification and its statistical inference are developed. Selection of the R-vine tree structure, the pair-copula families and its parameters are tackled in Section 3. This includes a simulation study presented in Appendix A and shows that the proposed models by the search strategy are reasonable. The search and estimation algorithm is then successfully applied to a 16-dimensional financial data set involving daily equity, fixed income and commodity indices. In addition to sequential estimates full ML estimates are also provided. The paper closes with a summary and discussion.

2. Parametric regular-vine distributions

2.1. Regular vines

We begin this section with the theoretical background of a *regular vine (R-vine)*, we then give its representation as an array and show how the R-vine copula density can be written in a convenient way using this array form. The following summarizes some definitions and results from Bedford and Cooke (2001),

Bedford and Cooke (2002, Part 4) and Kurowicka and Cooke (2006, Chapter 4.4), where a tree is a graph in which each two nodes are connected by a unique sequence of edges.

Definition 2.1 (R-vine). *$\mathcal{V} = (T_1, \ldots, T_{n-1})$ is an R-vine on n elements if*

(i) *T_1 is a tree with nodes $N_1 = \{1, \ldots, n\}$ and a set of edges denoted E_1.*
(ii) *For $i = 2, \ldots, n-1$, T_i is a tree with nodes $N_i = E_{i-1}$ and edge set E_i.*
(iii) *For $i = 2, \ldots, n-1$ and $\{a, b\} \in E_i$ with $a = \{a_1, a_2\}$ and $b = \{b_1, b_2\}$ it must hold that $\#(a \cap b) = 1$* (proximity condition)*, where $\#$ denotes the cardinality of a set.*

In other words, an R-vine on n elements is a nested set of $n-1$ trees such that the edges of tree j become the nodes of tree $j+1$. The proximity condition insures that two nodes in tree $j+1$ are only connected by an edge if these nodes share a common node in tree j. We notice that the set of nodes in the first tree contains all indices $1, ..., n$, while the set of edges is a set of $n-1$ pairs of these indices. In the second tree the set of nodes contains sets of pairs of indices and the set of edges is built of pairs of pairs of indices, etc.

To further study properties of R-vines we define three sets associated with its edges. The *complete union* of an edge is a set of all indices that this edge contains. If two nodes a and b are joined by an edge, then the *conditioned* and *conditioning sets* of this edge are the symmetric difference and the intersection of the complete unions of a and b, respectively.

Definition 2.2 (Complete union, conditioning and conditioned sets of an edge). *The complete union of an edge $e_i \in E_i$ is the set $U_{e_i} = \{n_1 \in N_1 | \exists e_j \in E_j, j = 1, \ldots, i-1,$ with $n_1 \in e_1 \in e_2 \in \ldots \in e_{i-1} \in e_i\} \subset N_1$. For $e_i = \{a, b\} \in E_i$, $a, b \in N_i$, $i = 1, \ldots, n-1$, the conditioning set of an edge e_i is $D_{e_i} = U_a \cap U_b$, and the conditioned sets of an edge e_i are $C_{e_i,a} = U_a \setminus D_{e_i}$, $C_{e_i,b} = U_b \setminus D_{e_i}$ and $C_{e_i} = C_{e_i,a} \cup C_{e_i,b} = U_a \triangle U_b$, where $A \triangle B := (A \setminus B) \cup (B \setminus A)$ denotes the symmetric difference of two sets.*

The complete union of the edge a between $(1, 2)$ and $(2, 3)$ in tree T_2 shown in Figure 1 is $\{1, 2, 3\}$, since for instance $1 \in \{1, 2\} \in \{\{1, 2\}, \{2, 3\}\} = \{a, b\}$ and $3 \in \{2, 3\} \in \{\{1, 2\}, \{2, 3\}\} = \{a, b\}$, and the complete union of the edge b between $(2, 3)$ and $(3, 6)$ is $\{2, 3, 6\}$. The conditioning and the conditioned sets of the edge joining a and b are $\{2, 3\}$ and $\{1, 6\}$, respectively.

The conditioned and conditioning sets of all edges of $\mathcal{V}$ are collected in a set called *constraint set*. Each element of this set is composed of a pair of indices corresponding to the conditioned set and a set containing indices corresponding to the conditioning set.

Definition 2.3 (Constraint set). *The constraint set for $\mathcal{V}$ is a set:*

$$\mathcal{CV} = \big\{(\{C_{e,a}, C_{e,b}\}, D_e) | e \in E_i, e = \{a, b\}, i = 1, \ldots, n-1\big\}.$$

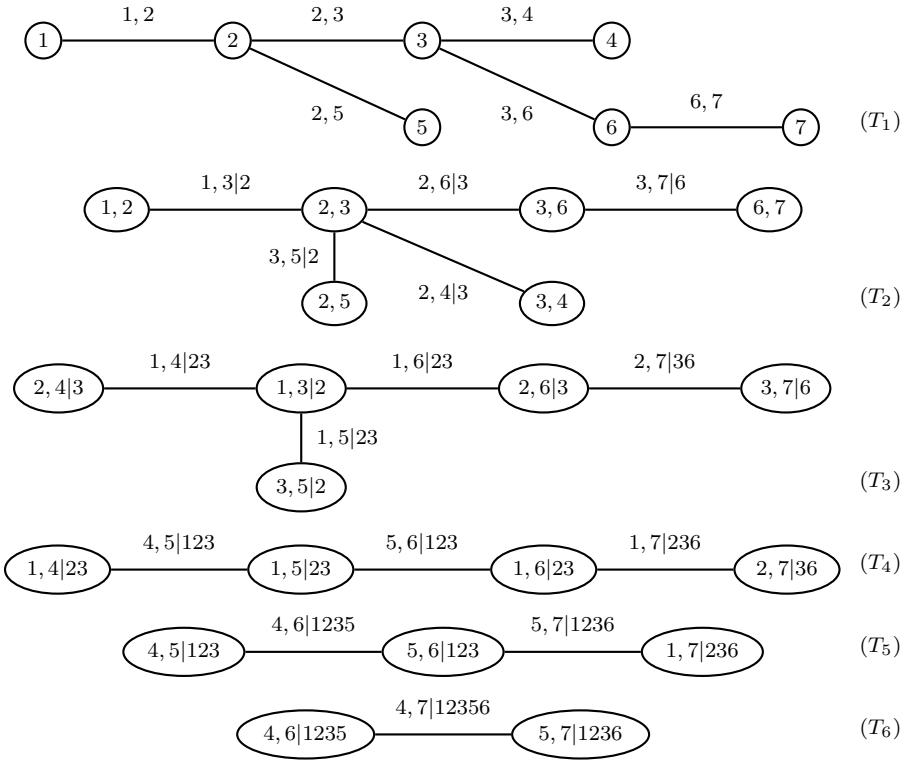


Figure 1: An example R-vine on seven variables. At each edge $e = \{a, b\} \in E_i$, the terms $C_{e,a}$ and $C_{e,b}$ are separated by a comma and given to the left of the '|' sign, while D_e appears on the right.

It is convenient to enumerate nodes of the trees in an R-vine using their conditioned and conditioning sets. In Figure 1 each edge of the R-vine has been assigned with its conditioned sets printed before '|' and the conditioning set shown after '|'. Moreover we notice that the constraint set of an R-vine $\mathcal{CV}$ contains all necessary information needed to distinguish it from other R-vines.

Two special types of R-vines namely the *canonical (C-)* and the *D-vine* have been used extensively in the literature. A D-vine is an R-vine for which the first tree has nodes with degree two or less (path structure). A C-vine is an R-vine which contains a node with maximal degree in each tree (star structure). It is convenient to work with these two R-vine types as the first tree (D-vine) and the ordering of the root nodes (C-vine) determine their structure completely.

R-vines have many interesting properties that can be found in Bedford and Cooke (2002) and Kurowicka and Joe (2011).

2.2. Regular vine copulae

The graphical structure of R-vines is used to specify necessary copulae for a so-called pair-copula construction, where a copula is a multivariate distribu-

tion on the unit hypercube $[0,1]^n$ with uniform marginal distributions (see Joe (1997) and Nelsen (2006)). To build an R-vine copula one must specify $n-1$ unconditional bivariate copulae between variables indexed by the conditioned sets of the edges in the first tree of the R-vine. For the second tree of the R-vine one needs to specify the bivariate copulae between variables indexed by the conditioned sets conditional on variables indexed by the conditioning sets of edges of R-vine. We formally define the R-vine copula specification corresponding to an R-vine as in Bedford and Cooke (2002).

Definition 2.4 (R-vine copula specification). *$(\boldsymbol{F}, \mathcal{V}, B)$ is an R-vine copula specification if $\boldsymbol{F} = (F_1, \ldots, F_n)$ is a vector of continuous invertible distribution functions, $\mathcal{V}$ is an n-dimensional R-vine and $B = \{B_e | i = 1, \ldots, n-1; e \in E_i\}$ is a set of copulae with B_e being a bivariate copula, a so-called pair-copula.*

A joint distribution F of a random vector $(X_1, \ldots, X_n)$ is said to realize an R-vine copula specification $(\boldsymbol{F}, \mathcal{V}, B)$ or exhibit R-vine dependence if, for each $e \in E_i$, $i = 1, \ldots, n-1$, $e = \{a, b\}$, B_e is the bivariate copula of $X_{C_{e,a}}$ and $X_{C_{e,b}}$ given $\boldsymbol{X}_{D_e} = \{X_i | i \in D_e\}$, where it is assumed that this conditional copula is independent of the conditioning variables $\boldsymbol{X}_{D_e}$ (see Aas et al. (2009) and Hobæk Haff et al. (2010)). We call such a distribution also an *R-vine distribution*. Additionally, the marginal distribution of X_j has to be F_j for $j = 1, \ldots, n$. We denote the copula density of the copula B_e for the edge $e = \{a, b\}$ as $c_{C_{e,a}, C_{e,b} | D_e}$.

For the R-vine from Figure 1 we need to assign six unconditional copulae $c_{1,2}, c_{2,3}$, $c_{3,4}, c_{2,5}, c_{3,6}$ and $c_{6,7}$ in the first tree, five conditional copulae in the second tree $c_{1,3|2}, c_{2,6|3}$, $c_{3,7|6}, c_{3,5|2}$ and $c_{2,4|3}$, etc. All copulae can be of a different type and their parameters can be specified independently from each other. However, since the copulae specified in a tree will affect the conditioned variables used in later trees the choice of the different copulae will influence each other.

The density of an R-vine copula specified through assigning appropriate bivariate copulae to edges of the R-vine has been shown in Bedford and Cooke (2001, 2002) to be equal to the product of conditional and unconditional copulae assigned to its edges.

Theorem 2.5. *Let $(\boldsymbol{F}, \mathcal{V}, B)$ be an R-vine copula specification on n elements. There is a unique distribution F that realizes this R-vine copula specification with density*

$$f_{1\ldots n}(\boldsymbol{x}) = \prod_{k=1}^{n} f_k(x_k) \prod_{i=1}^{n-1} \prod_{e \in E_i} c_{C_{e,a}, C_{e,b} | D_e}\big(F_{C_{e,a}|D_e}(x_{C_e,a} | \boldsymbol{x}_{D_e}), F_{C_{e,b}|D_e}(x_{C_e,b} | \boldsymbol{x}_{D_e})\big), \tag{1}$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)$, $e = \{a, b\}$ and $\boldsymbol{x}_{D_e}$ stands for the variables in D_e, i.e., $\boldsymbol{x}_{D_e} = \{x_i | i \in D_e\}$. Moreover f_i denotes the density of F_i for $i = 1, \ldots, n$.

Notice that the copulae in (1) are indexed by elements of the set $\mathcal{CV}$ (see Definition 2.3). To obtain the conditional distributions $F_{C_{e,a}|D_e}(x_{C_{e,a}}|\boldsymbol{x}_{D_e})$ and $F_{C_{e,b}|D_e}(x_{C_{e,b}}|\boldsymbol{x}_{D_e})$ let $E_i \ni e = \{a, b\}, a = \{a_1, a_2\}, b = \{b_1, b_2\}$ be the edge which connects $C_{e,a}$ with $C_{e,b}$ given the variables D_e. Joe (1996) showed that

$$\begin{aligned} F_{C_{e,a}|D_e}(x_{C_{e,a}}|\boldsymbol{x}_{D_e}) &= \frac{\partial C_{C_a|D_a}(F_{C_{a,a_1}|D_a}(x_{C_{a,a_1}}|\boldsymbol{x}_{D_a}), F_{C_{a,a_2}|D_a}(x_{C_{a,a_2}}|\boldsymbol{x}_{D_a}))}{\partial F_{C_{a,a_2}|D_a}(x_{C_{a,a_2}}|\boldsymbol{x}_{D_a})} \\ &=: h(F_{C_{a,a_1}|D_a}(x_{C_{a,a_1}}|\boldsymbol{x}_{D_a}), F_{C_{a,a_2}|D_a}(x_{C_{a,a_2}}|\boldsymbol{x}_{D_a})), \end{aligned} \tag{2}$$

where $F_{C_{a,a_1}|D_a}(x_{C_{a,a_1}}|\boldsymbol{x}_{D_a})$ and $F_{C_{a,a_2}|D_a}(x_{C_{a,a_2}}|\boldsymbol{x}_{D_a}))$ have to be obtained recursively as shown in the next section. The notation of the h-function is introduced for convenience.

Similarly, we obtain $F_{C_{e,b}|D_e}(x_{C_{e,b}}|\boldsymbol{x}_{D_e})$. We call $F_{C_{e,a}|D_e}(x_{C_{e,a}}|\boldsymbol{x}_{D_e})$ and $F_{C_{e,b}|D_e}(x_{C_{e,b}}|\boldsymbol{x}_{D_e})$ *transformed variables.*

For C- and D-vines the density (1) can be rewritten in a more convenient way. For more information on how to exploit the structure of C- and D-vines see Berg and Aas (2009), Min and Czado (2010, 2011) and Czado et al. (2010).

2.3. Array representation of regular vines

To develop statistical inference algorithms for R-vines we need a convenient way of representing an R-vine. Storing the nested set of trees is too expensive and does not allow for an easy way to describe inference algorithms.

Morales-Nápoles (2008) uses a lower triangular array to store an R-vine. The idea is to store the constraint set of an R-vine in columns of an n-dimensional lower triangular array. We hence specify how the information from the lower triangular array should be read by defining a constraint set for the array. In the next section we introduce a way how the structure of R-vine arrays can be used to encode corresponding pair-copula types and parameters. While Morales-Nápoles (2008) used the array representation of R-vines for counting the number of different R-vines, we will subsequently exploit this structure for likelihood computation and a sampling procedure.

Definition 2.6 (Array constraint set). *Let $M = (m_{i,j})_{i,j=1,\dots n}$ be a lower triangular array. The i-th constraint set for M is*

$$\mathcal{C}_M(i) = \big\{(\{m_{i,i}, m_{k,i}\}, D) | k = i+1, \dots, n, D = \{m_{k+1,i}, \dots, m_{n,i}\}\big\} \tag{3}$$

for $i = 1, \dots, n-1$. If $k = n$ we set $D = \emptyset$. The constraint set for array M is the union $\mathcal{C}M = \mathcal{C}_M(1) \cup \dots \cup \mathcal{C}_M(n-1)$. For the elements of the constraint set $(\{m_{i,i}, m_{k,i}\}, D) \in \mathcal{C}M$ we call $\{m_{i,i}, m_{k,i}\}$ the conditioned set and D the conditioning set.

Every element of the constraint set is made up of an diagonal entry $m_{i,i}$, an entry in the same column below the diagonal $m_{k,i}$ and all the elements following in that column $\{m_{k+1,i}, \dots, m_{n,i}\}$, $k = i+1, \dots, n$, $i = 1, \dots, n$.

To demonstrate this idea, we can compare the constraint sets defined by the example array M^* with the constraint sets of the R-vine in Figure 1.

$$M^* = \begin{pmatrix} 7 & & & & & & \\ 4 & 4 & & & & & \\ 5 & 6 & 6 & & & & \\ 1 & 5 & 5 & 5 & & & \\ 2 & 1 & 1 & 1 & 1 & & \\ 3 & 2 & 2 & 3 & 3 & 3 & \\ 6 & 3 & 3 & 2 & 2 & 2 & 2 \end{pmatrix}. \tag{4}$$

In the first column of M^* we have the diagonal entry $m_{1,1} = 7$ and the element $m_{4,1} = 1$ in the fourth row. According to the definition above this gives $(\{7, 1\}, \{2, 3, 6\}) \in \mathcal{C}M^*$ which corresponds to the constraint set of the rightmost edge of T_4 in the R-vine in Figure 1.

Before we formally define an R-vine array (that will be shown to code all information included in an R-vine) we need two sets that will help us characterize the array form and will ensure the proximity condition required for R-vines (see Definition 2.1). For a lower triangular array $M = (m_{i,j})_{i,j=1,\dots n}$ set for $i = 1, ..., n-1$,

$$B_M(i) := \big\{(m_{i,i}, D) | k = i+1, \dots, n; D = \{m_{k,i}, \dots, m_{n,i}\}\big\},$$
$$\widetilde{B}_M(i) := \big\{(m_{k,i}, D) | k = i+1, \dots, n; D = \{m_{i,i}\} \cup \{m_{k+1,i}, \dots, m_{n,i}\}\big\}.$$

Now we can define an R-vine array.

Definition 2.7 (R-vine array). *A lower triangular array $M = (m_{i,j})_{i,j=1,\dots n}$ is called an R-vine array if for $i = 1, \dots, n-1$ and for all $k = i+1, \dots, n-1$ there is an j in $i+1, \dots, n-1$ with*

$$(m_{k,i}, \{m_{k+1,i}, \dots, m_{n,i}\}) \in B_M(j) \text{ or } \in \widetilde{B}_M(j). \tag{5}$$

It can be shown that the following two properties follow from (5):

(i) $\{m_{i,i}, \dots, m_{n,i}\} \subset \{m_{j,j}, \dots m_{n,j}\}$ for $1 \le j < i \le n$,
(ii) $m_{i,i} \notin \{m_{i+1,i+1}, \dots, m_{n,i+1}\}$ for $i = 1, \dots, n-1$.

Condition (i) states that every column contains all the entries that a column to the right contains, while condition (ii) assures that there is a new entry on the diagonal in every column. Condition (5) is the essential counterpart to the proximity condition in the definition of an R-vine (see Definition 2.1). Note that Morales-Nápoles (2008) used a different condition to ensure the proximity condition.

As an example, one may check that M^* given in (4) fulfills condition (5) and is in fact an R-vine array.

The following simple properties of an R-vine array can be seen directly from the definition.

Properties 2.8. (i) *All elements in a column are different.*

(ii) *Deleting the first row and column from an n-dimensional R-vine array gives an* $(n-1)$*-dimensional R-vine array.*

We have seen that the array M^* codes all information needed to represent the R-vine in Figure 1. The proof that there is an equivalent R-vine array with the same constraint set for every R-vine and vice versa can be found in Dißmann (2010). In the proof it is shown that the constraint set $\mathcal{CV}$ of an R-vine is in fact equal to the constraint set $\mathcal{C}M$ of a corresponding R-vine array M. Note however that the array corresponding to an R-vine is not unique. As a simple example consider the array obtained after an exchange of the elements 2 and 3 in the lower right 2 by 2 corner of M^*. It defines the same R-vine as M^*.

2.4. Evaluation of the joint regular vine density

We now use the array representation for R-vines presented in the previous section to make more visible which copulae have to be used to build a density of the R-vine distribution. In particular, we provide a novel algorithm on how to efficiently evaluate the conditional distribution functions of an arbitrary R-vine copula. This is a non-trivial task, since the order of the conditioning variables required is not obvious. For this purpose we require an R-vine array that codes information about conditioned and conditioning variables. Let $M = (m_{i,j})_{i,j=1,\dots,n}$ be an R-vine array corresponding to the R-vine $\mathcal{V}$.

The R-vine distribution is a product of copulae indexed by $\mathcal{CV}$ which is equal to $\mathcal{C}M$ defined in (3). Hence the R-vine distribution density is:

$$f_{1\dots n} = \prod_{j=1}^{n} f_j \prod_{k=n-1}^{1} \prod_{i=n}^{k+1} c_{m_{k,k},m_{i,k}|m_{i+1,k},\dots,m_{n,k}}\left(F_{m_{k,k}|m_{i+1,k},\dots,m_{n,k}}, F_{m_{i,k}|m_{i+1,k},\dots,m_{n,k}}\right), \tag{6}$$

where arguments of all functions have been omitted to shorten the notation.

We now have to show how the conditional distributions which are arguments of bivariate copulae in (6) are obtained. We will show this in the algorithm below where the evaluation of the fully parametric form of an R-vine distribution is described. For this purpose we first need to specify two additional square arrays $T = (t_{i,j})_{i,j=1,\dots,n}$ and $P = (p_{i,j})_{i,j=1,\dots,n}$ that will contain information about types and parameters of the bivariate copulae in (6).

Since for all $j = 1, \dots, n-1$, $i = j+1, \dots, n$ the entry $m_{i,j}$ of M codes the copula of the variables indexed by $m_{j,j}$ and $m_{i,j}$ conditional on the variables indexed by $\{m_{i+1,j}, \dots, m_{n,j}\}$ we let $t_{i,j}$ describe the type of this copula (e.g., Normal, Clayton, etc.) and let $p_{i,j}$ contain parameters of this copula (note that some copulae require more than one parameter; we can store them, e.g., in additional arrays). An example of such a specification for M^* (see (4)) is shown in Figure 2.

Next, we find a recursive algorithm to calculate the conditional distributions. For convenience we will assume that the diagonal entries of M are ordered from

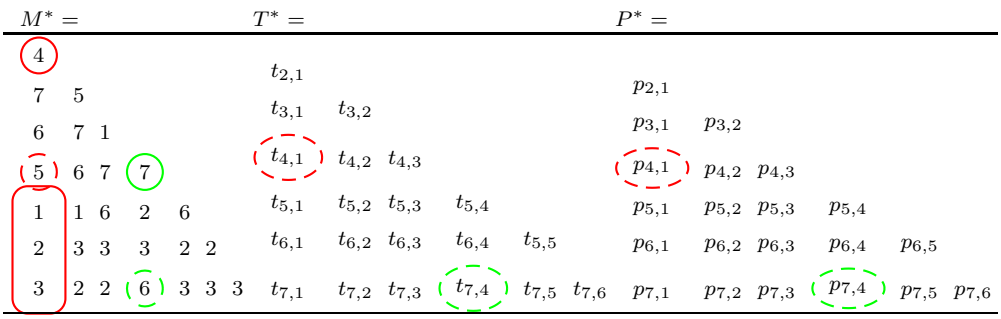


Figure 2: The copula with conditioned variables indexed by $\{4, 5\}$ and conditioning variables indexed by $\{1, 2, 3\}$, i.e., $c_{4,5|123}$, is of the type $t_{4,1}$ with parameter $p_{4,1}$. The copula $c_{7,6}$ is of the type $t_{7,4}$ and has the parameter $p_{7,4}$.

n to 1, i.e., $m_{k,k} = n - k + 1$. Note that the reordered array is equivalent to the original array which means it induces the same R-vine but with relabeled indices. The copula type and parameter arrays are unaffected by this reordering. To proceed, we introduce the maximum array of M denoted by $\mathbb{M}$. It is $\mathbb{M} = (\mathbf{m}_{i,k})_{i,k=1,\dots,n}$ with $\mathbf{m}_{i,k} = \max\{m_{i,k}, \dots, m_{n,k}\}$ for all $k = 1, \dots, n$ and $i = k, \dots, n$. In words, $\mathbf{m}_{i,k}$ is the maximum of all entries in the k-th column of M from the bottom up to the i-th element. Note that $\mathbf{m}_{n,k} = m_{n,k}$ for all $k = 1, \dots, n$, since $\mathbf{m}_{n,k}$ is the maximum over only one element and since the element on the diagonal is a new element in each column, it is $\mathbf{m}_{k,k} = m_{k,k} = n - k + 1$ for all $k = 1, \dots, n$.

Algorithm 2.1 shows how to compute the density for a given R-vine copula specification, where $h(\cdot, \cdot|t_{i,k}, p_{i,k})$ in Line 15 denotes the h-function (2) for the copula type $t_{i,k}$ with parameters $p_{i,k}$ and the arrays V^{direct} and V^{indirect} are introduced to store the arguments of the bivariate copulae in (6), where their notation is due to the order of the arguments in Line 15.

The outer **for**-loop of the algorithm iterates over the columns of M from right to left, starting with $n - 1$. The inner **for**-loop iterates over the rows from the bottom up to one element below the diagonal entry of M. Therefore, Line 14 of Algorithm 2.1 is executed once for every edge of the R-vine with the corresponding copula type and parameters.

Note that we do not need $(v_{n,1}^{\text{indirect}}, v_{n,2}^{\text{indirect}}, \dots, v_{n,n}^{\text{indirect}})$ because it is $\mathbf{m}_{n,k} = m_{n,k}$ for all $k = 1, \dots, n - 1$ and hence, we always select a v^{direct} in Line 9 for $i = n$.

The crucial point in the algorithm is how the conditional distributions that are arguments of bivariate copulae in (6) denoted as $z_{i,k}^{(1)}$ and $z_{i,k}^{(2)}$ are selected.

Therefore, we show that $z_{i,k}^{(1)} = F_{m_{k,k}|\{m_{i+1,k},\dots,m_{n,k}\}}(x_{m_{k,k}}|x_{m_{i+1,k}}, \dots, x_{m_{n,k}})$ and $z_{i,k}^{(2)} = F_{m_{i,k}|\{m_{i+1,k},\dots,m_{n,k}\}}(x_{m_{i,k}}|x_{m_{i+1,k}}, \dots, x_{m_{n,k}})$ for $k = n - 1, \dots, 1$ and $i = n, \dots, k + 1$.

We argue by induction and start with $i = n$ and k arbitrary in $1, \dots, i$.

It is $z_{n,k}^{(1)} = v_{n,k}^{\text{direct}} = F_{n-k+1}(x_{n-k+1}) = F_{m_{k,k}}(x_{m_{k,k}})$, and since $\mathbf{m}_{n,k} =$

Algorithm 2.1 Density of an R-vine specification.

Input: R-vine specification in array form, i.e., M, T, P, where $m_{k,k} = n - k + 1,\ k = 1, ..., n$.

Output: Density of the R-vine distribution at $(x_1, \dots x_n)$ for the given R-vine specification.

1: Set $f = 1$.
2: Allocate $V^{\text{direct}} = (v_{i,k}^{\text{direct}} | i, k = 1, \dots, n)$.
3: Allocate $V^{\text{indirect}} = (v_{i,k}^{\text{indirect}} | i, k = 1, \dots, n)$.
4: Set $(v_{n,1}^{\text{direct}}, v_{n,2}^{\text{direct}}, \dots, v_{n,n}^{\text{direct}}) = (F_n(x_n), F_{n-1}(x_{n-1}), \dots, F_1(x_1))$.
5: Let $\mathbb{M} = (\mathbf{m}_{i,k} | i, k = 1, \dots, n)$ with $\mathbf{m}_{i,k} = \max\{m_{i,k}, \dots, m_{n,k}\}$ for all $k = 1, \dots, n$ and $i = k, \dots, n$.
6: **for** $k = n-1, \dots, 1$ **do** {Iteration over the columns of M}
7: **for** $i = n, \dots, k+1$ **do** {Iteration over the rows of M}
8: Set $z_{i,k}^{(1)} = v_{i,k}^{\text{direct}}$.
9: **if** $\mathbf{m}_{i,k} = m_{i,k}$ **then**
10: Set $z_{i,k}^{(2)} = v_{i,(n-\mathbf{m}_{i,k}+1)}^{\text{direct}}$.
11: **else**
12: Set $z_{i,k}^{(2)} = v_{i,(n-\mathbf{m}_{i,k}+1)}^{\text{indirect}}$.
13: **end if**
14: Set $f = f \cdot c(z_{i,k}^{(1)}, z_{i,k}^{(2)} | t_{i,k}, p_{i,k})$.
15: Set $v_{i-1,k}^{\text{direct}} = h(z_{i,k}^{(1)}, z_{i,k}^{(2)} | t_{i,k}, p_{i,k})$ and $v_{i-1,k}^{\text{indirect}} = h(z_{i,k}^{(2)}, z_{i,k}^{(1)} | t_{i,k}, p_{i,k})$, where h is the conditional distribution function as defined in (2).
16: **end for**
17: **end for**
18: **return** Return the joint density f.

$m_{n,k}$, it is $z_{n,k}^{(2)} = v_{n,n-m_{n,k}+1}^{\text{direct}} = F_{m_{n,k}}(x_{m_{n,k}})$. Thereby, the statement is valid for $i = n$.

We assume that for all $n \geq i > I$ for an $I > 2$, i.e., for all $k = i, \dots, 1$ it is

$$v_{i-1,k}^{\text{direct}} = F_{m_{k,k}|\{m_{i,k}, m_{i+1,k}, \dots, m_{n,k}\}}(x_{m_{k,k}} | x_{m_{i,k}}, x_{m_{i+1,k}}, \dots, x_{m_{n,k}}) \quad (7)$$

and

$$v_{i-1,k}^{\text{indirect}} = F_{m_{i,k}|\{m_{k,k}, m_{i+1,k}, \dots, m_{n,k}\}}(x_{m_{i,k}} | x_{m_{k,k}}, x_{m_{i+1,k}}, \dots, x_{m_{n,k}}). \quad (8)$$

If we proceed with step I, the algorithm selects $z_{I,k}^{(1)} = v_{I,k}^{\text{direct}}$ in Line 8. By Equation (7) it is $z_{I,k}^{(1)} = F_{m_{k,k}|\{m_{I+1,k}, \dots, m_{n,k}\}}(x_{m_{k,k}} | x_{m_{I+1,k}}, \dots x_{m_{n,k}})$ which proves that the algorithm selects the correct entry for $z_{I,k}^{(1)}$.

By Definition 2.7, Property (iii) we know that there exists a j in $k+1, \dots, n-1$ with

$$(m_{I,k}, \{m_{I+1,k}, \dots, m_{n,k}\}) \in B_M(j) \cup \widetilde{B}_M(j). \quad (9)$$

Let $(x, D) \in B_M(j)$, then x and D consist of elements of the j-th column of M. Thus, $\max\{x, \max D\} = m_{j,j}$. This is also true for $(x, D) \in \widetilde{B}_M(j)$. If we

take the maximum over all elements on the left and right side of (9), it must hold that $\mathbf{m}_{I,k} = m_{j,j}$, and since $m_{j,j} = n-j+1$ we know that $j = n-\mathbf{m}_{I,k}+1$. This explains the indexation of v in Lines 10 and 12.

Now we distinguish between the cases $(m_{I,k}, \{m_{I+1,k}, \dots, m_{n,k}\}) \in B_M(j)$ and
$(m_{I,k}, \{m_{I+1,k}, \dots, m_{n,k}\}) \in \widetilde{B}_M(j)$. For $(m_{I,k}, \{m_{I+1,k}, \dots, m_{n,k}\}) \in B_M(j)$ it is

$$(m_{I,k}, \{m_{I+1,k}, \dots, m_{n,k}\}) = (m_{j,j}, \{m_{I+1,j}, \dots, m_{n,j}\}) \in B_M(j). \tag{10}$$

Hence, it follows $m_{I,k} = m_{j,j} = \mathbf{m}_{I,k}$. Thus, it is $m_{I,k} = \mathbf{m}_{I,k}$ in Line 9 of the algorithm, and the algorithm defines $z_{I,k}^{(2)} = v_{I,(n-\mathbf{m}_{I,k}+1)}^{\text{direct}} = v_{I,j}^{\text{direct}}$. Using the induction assumption (7) it follows

$$z_{I,k}^{(2)} = F_{m_{j,j}|\{m_{I+1,j}, m_{I+2,j}, \dots, m_{n,j}\}}(x_{m_{j,j}} | x_{m_{I+1,j}}, x_{m_{I+2,j}}, \dots x_{m_{n,j}}),$$

and by (10)

$$z_{I,k}^{(2)} = F_{m_{I,k}|\{m_{I+1,k}, \dots, m_{n,k}\}}(x_{m_{I,k}} | x_{m_{I+1,k}}, \dots x_{m_{n,k}}).$$

The argumentation for $(m_{I,k}, \{m_{I+1,k}, \dots, m_{n,k}\}) \in \widetilde{B}_M(j)$ is similar. This proves the statement.

2.5. Inference of regular vines

Having now established Algorithm 2.1 to evaluate a given R-vine copula density, the determination of the corresponding log likelihood expression L is straightforward by substituting Line 1 through "$L = 0$" and Line 14 through "$L = L + \log c(z_{i,k}^{(1)}, z_{i,k}^{(2)} | t_{i,k}, p_{i,k})$", and by returning L instead of f in the last line. The log likelihood can then be used, for example, for maximum likelihood estimation of the pair-copula parameters.

For vines there is a second estimation procedure which is typically used in the literature, namely *sequential estimation*. This method exploits the tree by tree structure of vines by separately estimating the parameter(s) of each pair-copula in the first tree, then computing the transformed variables for the second tree using h-functions, again separately estimating the conditional pair-copulae in the second tree, and so on. In doing so, only bivariate estimation is required and hence this method is quite fast. Moreover, the estimated parameters are typically good starting values for joint maximum likelihood estimation.

With regard to Algorithm 2.1, this means that we only have to insert a new line before Line 14, where the copula parameter $p_{i,k}$ is estimated based on the observations $z_{i,k}^{(1)}$ and $z_{i,k}^{(2)}$ and for copula family $t_{i,k}$.

Furthermore, sampling from R-vine specifications can be performed using the inverse probability integral transform (see Devroye (1986)). E.g., in the bivariate case, let C be the copula under consideration and let v_1 and v_2 be two

Algorithm 2.2 Simulation of an R-vine specification.

Input: R-vine specification in array form, i.e., M, T, P, where $m_{k,k} = n - k + 1,\ k = 1, ..., n$.

Output: Random observations $(x_1, \dots, x_n)$ from the R-vine specification.

Let $u_1, \dots, u_n$ be independent uniform samples.
Allocate $V^{\text{direct}} = (v_{i,k}^{\text{direct}} | i, k = 1, \dots, n)$.
Allocate $V^{\text{indirect}} = (v_{i,k}^{\text{indirect}} | i, k = 1, \dots, n)$.
Set $(v_{n,1}^{\text{direct}}, v_{n,2}^{\text{direct}}, \dots, v_{n,n}^{\text{direct}}) = (u_1, u_2, \dots u_n)$.
Let $\mathbb{M} = (\mathbf{m}_{i,k} | i, k = 1, \dots, n)$ with $\mathbf{m}_{i,k} = \max\{m_{i,k}, \dots, m_{n,k}\}$ for all $k = 1, \dots n-1$ and $i = k, \dots, n$.
$x_1 = v_{n,n}^{\text{direct}}$
for $k = n-1, \dots, 1$ **do** {Iteration over the columns of M}
  for $i = k+1, \dots, n$ **do** {Iteration over the rows of M}
    if $\mathbf{m}_{i,k} = m_{i,k}$ **then**
      Set $z_{i,k}^{(2)} = v_{i,(n-\mathbf{m}_{i,k}+1)}^{\text{direct}}$.
    else
      Set $z_{i,k}^{(2)} = v_{i,(n-\mathbf{m}_{i,k}+1)}^{\text{indirect}}$.
    end if
    Set $v_{n,k}^{\text{direct}} = h^{-1}(v_{n,k}^{\text{direct}}, z_{i,k}^{(2)} | t_{i,k}, p_{i,k})$
  end for
  $x_{n-k+1} = v_{n,k}^{\text{direct}}$
  for $i = n, \dots, k+1$ **do** {Iteration over the rows of M}
    Set $z_{i,k}^{(1)} = v_{i,k}^{\text{direct}}$
    Set $v_{i-1,k}^{\text{direct}} = h(z_{i,k}^{(1)}, z_{i,k}^{(2)} | t_{i,k}, p_{i,k})$ and $v_{i-1,k}^{\text{indirect}} = h(z_{i,k}^{(2)}, z_{i,k}^{(1)} | t_{i,k}, p_{i,k})$.
  end for
end for
return Return sample $(x_1, \dots, x_n)$.

independent uniform samples. Using the inverse of the h-function as defined in (2), $\boldsymbol{u} = (u_1, u_2)'$ given by

$$u_1 = v_1, \quad \text{and} \quad u_2 = h^{-1}(v_2, u_1) = F_{2|1}^{-1}(v_2 | u_1),$$

then is a sample from the copula C with uniform margins.

This idea can be generalized to R-vines and the corresponding algorithm is given in Algorithm 2.2, where we again assume that entries of the R-vine array are ordered from n to 1, and in particular the selection of the different $z_{i,k}^{(1)}$ and $z_{i,k}^{(2)}$ is the same as in Algorithm 2.1. More details on this can be found in Dißmann (2010, Section 5.3).

3. Selecting regular vine distributions

Fitting an R-vine copula specification to a given dataset requires the following separate tasks:

(a) Selection of the R-vine (structure), i.e., selecting which unconditioned and conditioned pairs to use.
(b) Choice of a bivariate copula family for each pair selected in (a).
(c) Estimation of the corresponding parameter(s) for each copula.

Since all three steps are needed for an R-vine copula specification, one way of finding the "best" model is to accomplish steps (b) and (c) for all possible R-vine constructions. Since the number of possible R-vines on n variables increases very rapidly with n ($n!/2 \times 2^{\binom{n-2}{2}}$ as shown in Morales-Nápoles et al. (2010)), this is not feasible. In addition to the fast growing number of possible R-vines, some methods to decide which bivariate copula family to use depend on the interpretation of plots, e.g., K- or Chi-Plots (see Genest and Favre (2007)), and therefore need manual interaction. On the one hand, we do not use such methods to obtain objectivity and, on the other hand, this again is not feasible to do for every possible copula in every possible R-vine decomposition. In particular, in Section 4 we will fit a model to a 16-dimensional dataset leaving 120 copulae to select. This is not practicable to do manually.

Therefore, we developed a sequential, heuristic method to select the tree structure of the R-vine. Since our proposed method for (a) depends on the copulae selected in (b) and estimated in (c), copula selection is covered in Section 3.2. A simulation study to evaluate our approach is presented in Appendix A.

In Section 4 we will apply the techniques.

3.1. Sequential method to select an regular vine copula specification based on Kendall's tau

To select one possible R-vine for a given dataset it is necessary to decide for which pairs of variables we want to specify copulae. We proceed sequentially, starting by defining the first tree $T_1 = (N_1, E_1)$ for the R-vine, continuing with the second tree, and so on. The trees are selected in such a way that the chosen pairs model the strongest pairwise dependencies present (more details below). Later, we will refer to this method as the *sequential method*. Since we examine every tree separately, it is not guaranteed to find a global optimum, where global optimum is meant in terms of model fit, e.g., higher likelihood, smaller AIC/BIC or superior in terms of the likelihood-ratio based test for comparing non-nested models proposed by Vuong (1989). However, we think this sequential approach is reasonable because

- the copula families specified in the first tree of the R-vine often have the greatest influence on the model fit.
- it is more important to model the dependence structure between random variables that have high dependencies correctly, because most copula families can model independence and the copulae distribution functions for parameters close to independence are very similar.
- this approach minimizes the influence of rounding errors in later trees, which pairs with strong pairwise dependence are most prone to, e.g., when

assessing the joint tail behavior of two variables. For pairs of variables close to independence, such issues are less relevant.

- for real applications it is natural to assume that randomness is driven by the dependence of only some variables and not all. Therefore, if you choose the copulae with high dependence in the first trees, the transformed variables for the later trees will often be rather independent. We exemplify this using the multivariate normal distribution, since we can easily compute conditional dependence for multivariate normal distributions using well known properties of the normal distribution (see, e.g., Anderson (2003)).

  For example consider the following three jointly normal distributed random variables.

$$\begin{pmatrix} X_1 \\ X_2 \\ X_2 \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho_{1,2} & \rho_{1,3} \\ \rho_{1,2} & 1 & \rho_{2,3} \\ \rho_{1,3} & \rho_{2,3} & 1 \end{pmatrix} \right),$$

  with pairwise correlations $\rho_{1,2}$, $\rho_{1,3}$ and $\rho_{2,3}$.

  For the normal distribution we know that the correlation of X_1 and X_2 given X_3 can be calculated as following

$$\rho_{1,2|3} := \rho\left(X_1|X_3, X_2|X_3\right) = \frac{\rho_{1,2} - \rho_{1,3}\rho_{2,3}}{\sqrt{1-\rho_{1,3}^2}\sqrt{1-\rho_{2,3}^2}}.$$

  Defining $\rho_{1,3} = \rho_{2,3} > \rho_{1,2} > 0$ we have $\rho_{1,2|3} = (\rho_{1,2} - \rho_{1,3}^2)/(1-\rho_{1,3}^2) < \rho_{1,2}$, since $\rho_{1,2} \leq 1$, and $\rho_{1,2|3} > 0$ because of the positive-definiteness of the correlation matrix. Hence, if we fit the dependence for the two pairs with higher correlation first (assumption $\rho_{1,3} = \rho_{2,3} > \rho_{1,2} > 0$) the remaining correlation of X_1 and X_2 becomes smaller given X_3.

  This is a desirable feature especially for datasets with a large number of variables, because we can truncate the R-vine specification and assume independence for the k last trees to reduce the number of parameters needed. For more information on this see Section 4 and Brechmann et al. (2012).

We use Kendall's tau as a measure of dependence, since it measures dependence independently of the assumed distribution and hence, is especially useful when combining different (non-Gaussian) copula families. However the described method works in the same way for every other measure of dependence (see Brechmann (2010, Chapter 3) for an extensive discussion).

Kurowicka (2011) proposes another method to generate R-vines. She builds the trees the other way around, starting with the last tree. By this method she tries to generate an R-vine with the lowest dependencies in the top trees. This method depends on the partial correlations which contradicts the fact that we want to use other, non-Gaussian copulae. Partial correlations are used, since

Algorithm 3.1 Sequential method to select an R-vine model based on Kendall's tau.

Input: Data $(x_{\ell 1}, \dots x_{\ell n})$, $\ell = 1, ..., N$ (realizations of i.i.d. random vectors).
Output: R-vine copula specification, i.e., $\mathcal{V}$, B.

1: Calculate the empirical Kendall's tau $\hat{\tau}_{j,k}$ for all possible variable pairs $\{j,k\}, 1 \leq j < k \leq n$.
2: Select the spanning tree that maximizes the sum of absolute empirical Kendall's taus, i.e.,

$$\max \sum_{e=\{j,k\} \text{ in spanning tree}} |\hat{\tau}_{j,k}|.$$

3: For each edge $\{j,k\}$ in the selected spanning tree, select a copula and estimate the corresponding parameter(s). Then transform $\widehat{F}_{j|k}(x_{\ell j}|x_{\ell k})$ and $\widehat{F}_{k|j}(x_{\ell k}|x_{\ell j})$, $\ell = 1, ..., N$, using the fitted copula $\widehat{C}_{jk}$ (see (2)).
4: **for** $i = 2, \dots, n-1$ **do** {Iteration over the trees}
5: Calculate the empirical Kendall's tau $\hat{\tau}_{j,k|D}$ for all conditional variable pairs $\{j,k|D\}$ that can be part of tree T_i, i.e., all edges fulfilling the proximity condition (see Definition 2.1).
6: Among these edges, select the spanning tree that maximizes the sum of absolute empirical Kendall's taus, i.e.,

$$\max \sum_{e=\{j,k|D\} \text{ in spanning tree}} |\hat{\tau}_{j,k|D}|.$$

7: For each edge $\{j,k|D\}$ in the selected spanning tree, select a conditional copula and estimate the corresponding parameter(s). Then transform $\widehat{F}_{j|k\cup D}(x_{\ell j}|x_{\ell k}, \boldsymbol{x}_{\ell D})$ and $\widehat{F}_{k|j\cup D}(x_{\ell k}|x_{\ell j}, \boldsymbol{x}_{\ell D})$, $\ell = 1, ..., N$, using the fitted copula $\widehat{C}_{jk|D}$ (see (2)).
8: **end for**

they can be calculated without knowing the exact R-vine structure of the first trees.

Our method is summarized in Algorithm 3.1. To select the tree that maximizes the sum of absolute empirical Kendall's taus (Steps 2 and 6) we use a maximum spanning tree (MST) algorithm such as the Algorithm of Prim (Cormen et al., 2001, Section 23.2). Typically such algorithms are described in a way to find a *minimal* spanning tree. But the algorithms work in both ways. Also note that in Steps 2 and 6 we are looking for a tree. We could look for a star or a path instead, to obtain a C- or a D-vine structure, respectively. Note that for a D-vine a Hamiltonian path has to found which corresponds to solving a Traveling Salesman Problem. This is however NP-equivalent and therefore rather inefficient to find a solution for, especially in higher dimensions.

Notice that an MST algorithm does not depend on the the actual values of the edges, instead it only uses their rank. Therefore, the algorithm leads to the

same results if we transform the edge values by a monotone increasing function. Hence, in our field of application, where we want to find a tree with maximal values of taus we would get the same tree even if we took other weights like squared taus or another monotone increasing transformation.

How to select a copula, i.e., Steps 3 and 7 of Algorithm 3.1 is explained in more detail in Section 3.2. A proof that this algorithm creates an R-vine, i.e., that we always find a tree in Steps 2 and 6 and further explanations are given in the following.

An MST algorithm always leads to a tree when the input graph is connected. Therefore, we need to check this assumption to verify our method.

This is obviously true for T_1, since we start with a complete graph. When conducting the i-th step, we know that T_{i-1} is a tree. The node set of tree T_i is then given by $N_i = E_{i-1}$. Let E_i' be the set of all possible edges in T_i (see Step 5 of Algorithm 3.1). This edge set is defined by

$$E_i' = \{\{a, b\} \in N_i^2 | \#(a \cap b) = 1\}. \tag{11}$$

The requirement $\#(a \cap b) = 1$ ensures the proximity condition of an R-vine. To show that (N_i, E_i') is connected recall that connected means there is a path from every single node to every other node. Let $a, b \in N_i$ be arbitrary nodes. Further, let $n_1, n_2 \in N_{i-1}$ be two nodes from the previous tree with $n_1 \in a$ and $n_2 \in b$. Since n_1 and n_2 are nodes of a tree, there is a path in T_{i-1} from n_1 to n_2, $n_1 \in e_1 \to \ldots \to e_l \ni n_2$, $e_1, \ldots, e_l \in E_{i-1}$, $l \geq 1$. We know that $n_1 \in a$ and $n_1 \in e_1$. Without loss of generality we can assume that $a = e_1$. Otherwise, if $e_1 \neq a$, we can extend the path

$$\begin{aligned} e_{l+1} &= e_l \\ &\vdots \\ e_2 &= e_1 \\ e_1 &= a \\ l &= l+1. \end{aligned}$$

Similarly we can assume $b = e_l$. Since $e_1, \ldots, e_l$ induce a path, we know that $\#(e_i \cap e_{i+1}) = 1$ for all $i = 1, \ldots, l-1$. Hence $\{e_i, e_{i+1}\} \in E_i'$ for all $i = 1, \ldots, l-1$. Thus, we know that there is a path from $e_1 = a$ to $e_l = b$ and (N_i, E_i') is a connected graph. Table 1 shows a concrete example of this idea.

Finally, we give some more insight on how to calculate the empirical Kendall's taus and select copula families. Define E_i' like it was done in (11). For all $e \in E_i'$ we have to calculate the value of Kendall's tau, and for some of them (those selected in the MST) we need to fit a copula based on two conditioned variables. If $e \in E_i'$, $e = \{a, b\}$ connects variables $x_{C_{e,a}}$ with $x_{C_{e,b}}$ given the variables $\boldsymbol{x}_{D_e}$, we hence need the transformed variables $F_{C_{e,a}|D_e}(x_{C_{e,a}}|\boldsymbol{x}_{D_e})$ and $F_{C_{e,b}|D_e}(x_{C_{e,b}}|\boldsymbol{x}_{D_e})$ which are obtained as described in (2). For these it is then straightforward to calculate the empirical Kendall's tau and select a bivariate copula family as outlined in the following section.

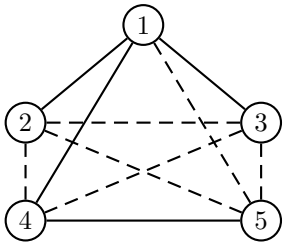
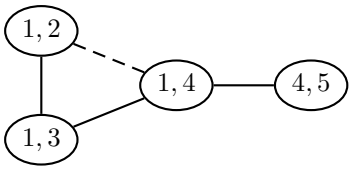
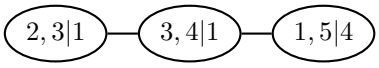
i	Graph	Description
1		Assume that we have 5 variables $N_1 = \{1, 2, 3, 4, 5\}$. The first graph is always a complete graph, where we can connect every node with every other node. Let us assume the Algorithm of Prim selects the solid edges. The concrete edge values (Kendall's taus) are not of interest in this example.
2		All edges from the previous step are now nodes. An edge is drawn whenever the nodes share a common node in the previous tree (dashed and solid). We see that the graph is connected and select the tree indicated by the solid edges.
3		There are no options in this step. We need all edges to form a tree. Note, as soon as a graph has a D-vine structure, there are no more options in the following trees because they it uniquely determines all following conditioned and conditioning sets.

Table 1: Exemplification of the model selection Algorithm 3.1.

3.2. Selecting pair-copula families sequentially

Besides the steps described above we need to select a copula family for every pair of variables. In the later application we take the following copula families into consideration (some properties are given in brackets):

- Gaussian/Normal (tail-symmetric, no tail dependence),
- Student-t (tail-symmetric, tail dependence),
- Gumbel (tail-asymmetric, upper tail dependence) and survival Gumbel (tail-asymmetric, lower tail dependence),
- rotated Gumbel by 90 and 270 degrees (tail-asymmetric, no tail dependence),
- Frank (tail-symmetric, no tail dependence).

In case of positive dependence this means that we can select among the Gaussian, Student-t, (survival) Gumbel and Frank copulae, while rotated Gumbel copulae can be used instead of Gumbel and survival Gumbel copulae when modeling negative dependence. Further, we will not use a Student-t copula if the maximum likelihood estimation leads to a degrees of freedom parameter higher than 30 because then the Student-t copula is too close to the Gaussian which can be used instead.

Given these options we still have to decide which copula fits "best". We do this using the AIC (Akaike, 1973) which corrects the log likelihood of a copula for the number of parameters, i.e., the use of the Student-t copula is penalized compared to the other copulae, since it is the only two parameter family under consideration. Bivariate copula selection using the AIC has previously been investigated in Manner (2007) and Brechmann (2010, Section 5.4) who found that it is a quite reliable criterion, in particular in comparison to alternative criteria such as copula goodness-of-fit tests. Selection proceeds by computing the AIC's for each possible family and then choosing the copula with smallest AIC. We will also include the independence copula in the selection by performing a preliminary independence test based on Kendall's tau as described in Genest and Favre (2007). If this test indicates independence, no further steps are taken and the independence copula is chosen.

Given the wide range of bivariate copula families available the above list of copulae clearly is not complete. For instance, we could also consider two parameter copula families such as the BB1 or BB7 with different lower and upper tail dependence. These have previously been used as building blocks of C- and D-vine copulae by Czado et al. (2010) and Nikoloulopoulos et al. (2012). While already including copula families able to account for very different types of dependence, the above list can easily be extended by such families, which however increases the computational burden of the copula selection step. Using appropriate diagnostic tools for asymmetry and tail dependence as in the above two references, the required computational time can however be reduced.

4. Modeling the residual dependency among daily returns of international financial indices

Copula based models are very commonly used in the area of multivariate modeling of financial returns. Here first appropriate marginal time series models are fitted to each financial return series and standardized residuals are formed. The dependency among these residuals is then modeled using a multivariate copula after a transformation to marginally uniform data using either an empirical or parametric probability integral transformation. There has been empirical evidence that different asymmetric and tail dependencies are present for different pairs of variables, which cannot be captured using a multivariate Gaussian or Student-t copula with a common degree of freedom (see, amongst others, Longin and Solnik (1995, 2001) and Ang and Bekaert (2002)). Especially D-vines have been shown to be very successful in the modeling of such dependency patterns (see Aas et al. (2009), Min and Czado (2010) and Mendes et al. (2010)),

but also C-vines have recently been successfully applied (Czado et al., 2010). Mendes et al. (2010) however suggested that there should more research on how to choose D-vines including both the choice of the order of the nodes as well as how to choose the pair-copula families. This paper is exactly answering these questions and in our application we will investigate whether R-vine copulae other than C- or D-vine and standard multivariate copulae are needed in modeling the residual dependencies among financial returns.

For this we selected 16 international indices, including five equity, nine fixed income (bonds) and two commodity indices observed daily from 12/29/2001 until 12/14/2009 (2337 daily returns). All returns are unhedged against currency fluctuations and quoted in their home currency except for global indices which are stated in USD. In particular we choose the equity indices DAX, STOXX50, S&P500, MSCI-World and MSCI-EE, the fixed income indices IBOXX-G-3-5, IBOXX-G-7-10, IBOXX-E-1-3, IBOXX-E-5-7, IBOXX-E-10+, IBOXX-E-A, IBOXX-E-AA, IBOXX-E-AAA, IBOXX-E-BBB and the commodity indices Comm and Gold. For the bonds we selected maturities such that those of the German and the Euro bonds are disjoint, since German bonds (IBOXX-G) account for a large proportion of the Euro indices (IBOXX-E) giving rise to extremely high Pearson correlations which are also observed between consecutive maturities (see the corresponding pairs in Figure 4 below). More information about the selected indices can be found in Table 6.13 of Dißmann (2010).

For the first step we fitted univariate ARMA(1,1)-GARCH(1,1) models with Student-t innovations using maximum likelihood estimation to all equity and commodity indices and Gauss innovations for all bond indices, separate residual analyses in Dißmann (2010, Section 6.3.1 and Appendix B.3) show no volatility clusters and a good fit of the chosen innovation distribution for equity and commodity indices. For bond indices the innovation distributions are only reasonable. Corresponding Ljung-Box tests indicate independence of the standardized residuals. Since the sample size is large and there is always some uncertainty in the innovation distribution we selected the empirical probability integral transformation to obtain marginally uniform data. The resulting pairwise scatter plots of the resulting copula data (top triangular matrix) and their estimated Kendall's tau values (lower triangular matrix) for six representatives from the different indices are given in Figure 3 indicating different strengths and signs of pairwise dependencies.

For model selection we want to demonstrate the superior fit of R-vines with individually chosen pair-copula families and assess the gain over R-vines with only bivariate t or with only Gauss pair-copulae as well as over standard C- and D-vines. In particular we apply the selection algorithm of Section 3 to select among five different R-vine classes given by

- **mixed R-vine**: R-vine with pair-copula terms chosen individually from seven bivariate copula types (Gauss, Student-t, Gumbel, survival Gumbel, rotated Gumbel (90 and 270 degrees), Frank).
- **mixed C-vine**: C-vine with pair-copula terms chosen individually from seven bivariate copula types (see above).

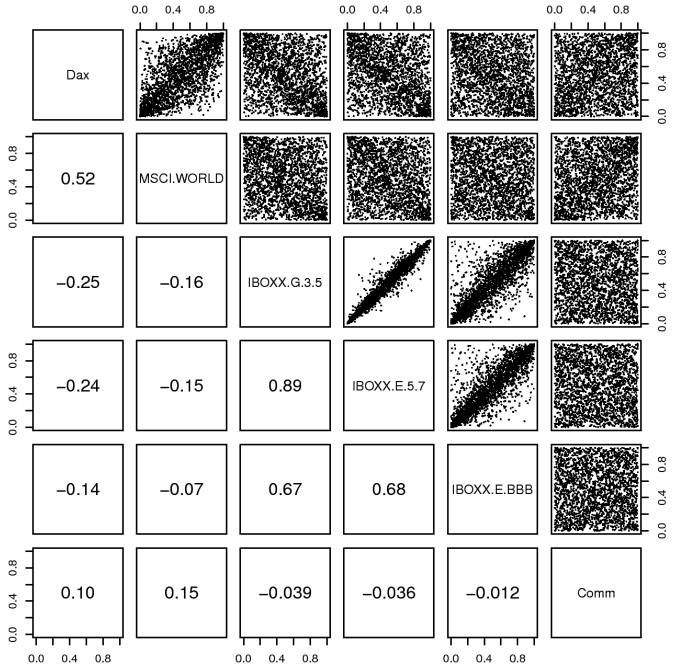


Figure 3: Pairs-plots and Kendall's taus for representatives of each index group.

- **mixed D-vine**: D-vine with pair-copula terms chosen individually from seven bivariate copula types (see above).
- **all t R-vine**: R-vine with each pair-copula term chosen as bivariate Student-t copula. If the degrees of freedom parameter of a pair is estimated to be larger than 30, we set the copula to the Gaussian.
- **multivariate Gauss**: R-vine with each pair-copula term chosen as bivariate Gaussian copula, i.e., this corresponds to a multivariate Gaussian copula, where unconditional correlations can be obtained from conditional ones by inverting a generalized version of Equation (3.1).

The top tree is common to all R-vines (in contrast to the C- and D-vines which are determined as maximal stars and paths as noted in Section 3), since the selection of the top tree does not depend on the pair-copula choice (but only on the empirical Kendall's taus) and is given in Figure 4. The structure in Figure 4 reflects expected relationships among the residuals of the indices. The government bond indices are grouped so that consecutive maturities are connected. Similarly corporate bond indices are aligned according to their ratings from lowest (BBB) to highest (AAA). These two groups are connected by an average representative, i.e., IBOXX-E-5-7 and IBOXX-E-AA. Since STOXX50 is

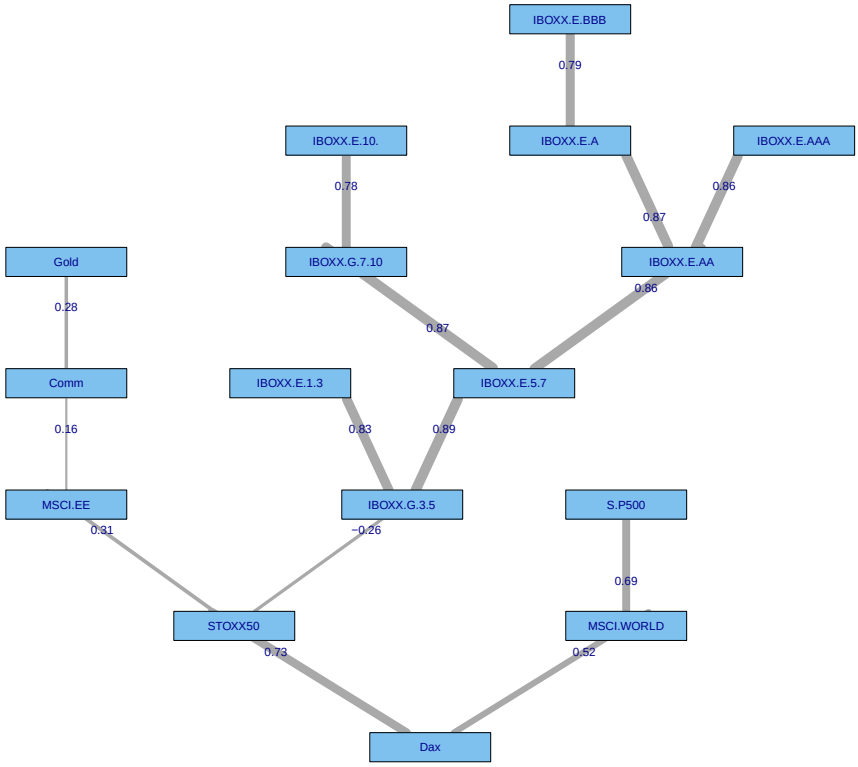


Figure 4: T_1 for an R-vine from the model selection algorithm.

a European equity index the residual dependency is highest to the predominant Euro bond index (IBOXX-G-3-5).

For the copula family selection of each pair-copula term the AIC is used as described in Section 3.2, where pair-copula parameters are estimated by maximum likelihood estimation. In applying the selection algorithm we also observed that empirical Kendall's tau values tend to be small for higher order trees. In this cases it might be sufficient to replace the corresponding pair-copula term by the independence copula. Therefore we also fitted an R-vine using the preliminary independence test based on Kendall's tau for each pair ("**indep. R-vine**"). If the p-value of the test is larger than 5%, then we choose the independence copula for this pair-copula term. The issue of large numbers of independence copulae in later trees is further investigated in Brechmann et al. (2012) who call an R-vine *truncated* if all pair-copulae in higher order trees are

set to bivariate independence copulae.

Applying the selection procedure to the **R-vine mixed case** 16 Gauss, 51 Student-t, 4 Gumbel, 7 survival Gumbel, 12 rotated Gumbel and 30 Frank bivariate copula terms requiring 171 parameter estimates were chosen. If the choice for a pairwise independence copula is allowed, the total number of parameters was significantly reduced to 108, since 55 copula terms were replaced by an independence copula. These models correspond to the mixed/t scenario of the simulation study in Appendix A and hence we can assume that our models give rather adequate fits compared to the (unknown) "true" model.

Selection results for all models are summarized in Table 2. It shows the log likelihood achieved for sequential estimates in the first row, while the second row gives the log likelihood after joint optimization of the chosen regular vine tree specification and copula types (see Section 2.5). The next rows indicate the number of pair-copula types chosen and the final rows give the test statistics together with the p-values in parentheses of a Vuong test with and without Akaike and Schwarz corrections, respectively, testing the R-vine mixed model against the alternative indicated by the respective column. This shows that the sequential log likelihood is quite close to the one obtained by joint maximization for all model classes considered. Especially the top four ranks are maintained. We also observe only small differences in the parameter estimates. The non-zero number of (survival/rotated) Gumbel pair-copula terms shows non-symmetric heavy tailed conditional dependencies present in the residual data. From the Vuong tests we see that the mixed R-vine is to be preferred over the mixed D-vine and the multivariate Gaussian copula. The difference to the all t R-vine and to the mixed C-vine is also more pronounced when using the (parsimonious) Schwarz correction, the mixed R-vine model is marginally superior in that case. The choice of Gaussian copulae for Student-t copulae with too many degrees of freedom means that the number of parameters in the all t R-vine is still close to that of the mixed R-vine. If we chose Student-t copulae for all terms, the number of parameters would be 240 and hence the influence of the corrections for the number of parameters used would be stronger. Finally, the mixed R-vine model reduced by independence pair-copula terms is preferred over the non-reduced mixed R-vine model if a Schwarz correction is used, since the reduced model has significantly less parameters to be estimated.

Overall this example demonstrates the usefulness of R-vine copulae with individually chosen copula types for each pair-copula term. In addition the R-vine tree selection procedure gives directly economically interpretable results for this data set.

A note on the required computing time: In our implementation the sequential selection and estimation Algorithm 3.1 took only between 5 minutes for the reduced mixed R-vine model and 9 minutes for the mixed C-vine on a Linux cluster computer with 32 processing cores (AMD Opteron, 2.6Ghz). In contrast the maximum likelihood estimation was computationally much more demanding. While the computing time for the non-reduced mixed R-vine model was only 1.5 hours, it increased to about 9 hours for the all t R-vine and the mixed C- and D-vine models.

		R-vine mixed	R-vine all t	R-vine all Gauss	R-vine indep.	C-vine mixed	D-vine mixed
Seq. log likelihood		36431	36417	30445	36331	36366	36300
Log likelihood		36514	36513	31784	36396	36476	36422
No. of parameters		171	179	120	108	178	176
No. of copulae	Indep.	0	0	0	55	0	0
	Gauss	16	61	120	8	19	18
	Student-t	51	59	0	43	58	56
	Gumbel	4	0	0	1	8	7
	Surv. Gumbel	7	0	0	1	8	6
	Rot. Gumbel	12	0	0	2	11	9
	Frank	30	0	0	10	16	24
Vuong tests	no correction		0.03	14.59	6.32	1.00	3.49
			(0.97)	(0.00)	(0.00)	(0.32)	(0.00)
	Akaike corr.		0.49	14.44	2.92	1.18	3.68
			(0.63)	(0.00)	(0.00)	(0.24)	(0.00)
	Schwarz corr.		1.79	13.98	-6.85	1.71	4.23
			(0.10)	(0.00)	(0.00)	(0.09)	(0.00)

Table 2: Log likelihoods, numbers of parameters and of copulae for all models as well as results of the Vuong tests (test statistics and p-values in parentheses) comparing the R-vine model with mixed copulae to all other models. The positive values of Vuong test statistics indicate that the test favors the R-vine model over the respective alternative model (inconclusive region at the 5%-level: $[-1.96, 1.96]$).

5. Summary and discussion

This paper provides a significant contribution towards making R-vine copulae a standard building block for copula based models. While already the introduction of C- and D-vine copulae provided flexibility in modeling dependencies, R-vine copulae provide even more modeling capabilities. Before the availability of such pair-copula constructions for multivariate copulae, the choices were rather limited. With R-vine copulae together with different choices for individual choices of copula types for each pair-copula term, the problem of too few modeling choices has shifted to the problem of too many choices to be investigated.

In this paper we provided a general selection approach to sequentially choose the tree representation together with choosing the copula type for each copula term from a large class of bivariate copula families and estimate the corresponding parameters. The selection approach involves sequentially the use of any graph theoretic algorithm which finds a maximum spanning tree. Absolute empirical Kendall's tau values are used as weights, but other weights are possible. In finance the use of empirical tail dependence or other measures of joint tail behavior might be useful to investigate.

The output of the selection procedure gives an R-vine tree structure, their corresponding pair-copula types and parameter estimates. These so-called se-

quential estimates can be used as starting values for determining the maximum likelihood estimates (see also Hobæk Haff (2011) for more details on the asymptotic behavior of these estimates). The paper also uses an array representation of an R-vine and provides a novel algorithm to evaluate the joint density for any arbitrary R-vine copula. The selection procedure is completely operational, it is implemented in the statistical software *R* and is capable to handle medium sized dimensions of up to 20 dimensions.

As noted in Section 4 it might be worthwhile to replace pair-copula terms by independence copula terms or simpler copula type choices in higher order trees. This issue has been investigated in the related work by Brechmann et al. (2012) who developed testing procedures to determine *truncation* after a certain tree. This further balances the model flexibility with the desired parsimony of the model and opens R-vines to applications in large dimensions (see also Brechmann and Czado (2011)).

In future, we will also investigate the model selection problem described in Section 3 more closely. This includes the choice of other weights than Kendall's tau as well as the selection of C- and D-vines. In particular, the selection of the order in the first D-vine tree corresponds to a Traveling Salesman Problem and therefore is NP-equivalent. Here, tailor-made approaches for the D-vine methodology have to be considered.

Acknowledgement

We acknowledge the helpful comments of the referees, which further improved the manuscript. The numerical computations were performed on a Linux cluster supported by DFG grant INST 95/919-1 FUGG.

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 likelihood ratio principle. In: Petrov, B. N. (Ed.), Proceedings of the Second International Synposium of Information Theory. Akademiai Kiado, Budapest, pp. 257–281.

Anderson, T. W., 2003. An introduction to multivariate statistical analysis. Wiley, Chichester.

Ang, A., Bekaert, G., 2002. International asset allocation with regime shifts. Review of Financial Studies 15 (4), 1137–1187.

Beaudoin, D., Lakhal-Chaieb, L., 2008. Archimedean copula model selection under dependent truncation. Statistics in Medicine 27 (22), 4440–4454.

Bedford, T., Cooke, R. M., 2001. Probability density decomposition for conditionally dependent random variables modeled by vines. Annals of Mathematics and Artificial Intelligence 32, 245–268.

Bedford, T., Cooke, R. M., 2002. Vines - a new graphical model for dependent random variables. Annals of Statistics 30 (4), 1031–1068.

Berg, D., Aas, K., 2009. Models for construction of higher-dimensional dependence: A comparison study. European Journal of Finance 15, 639–659.

Brechmann, E. C., 2010. Truncated and simplified regular vines and their applications. Diploma thesis, Technische Universität München.

Brechmann, E. C., Czado, C., 2011. Risk management with high-dimensional vine copulas: An analysis of the Euro Stoxx 50. Submitted for publication.

Brechmann, E. C., Czado, C., Aas, K., 2012. Truncated regular vines in high dimensions with applications to financial data. Canadian Journal of Statistics 40 (1), 68–85.

Cherubini, U., Luciano, E., Vecchiato, W., 2004. Copula Methods in Finance. Wiley, Chichester.

Chollete, L., Heinen, A., Valdesogo, A., 2009. Modeling international financial returns with a multivariate regime switching copula. Journal of Financial Econometrics 7, 437–480.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2001. Introduction to Algorithms, 2nd Edition. The MIT Press, Cambridge.

Czado, C., 2010. Pair-copula constructions of multivariate copulas. In: Jaworski, P., Durante, F., Härdle, W., Rychlik, T. (Eds.), Copula Theory and Its Applications. Springer, Berlin.

Czado, C., Schepsmeier, U., Min, A., 2010. Maximum likelihood estimation of mixed C-vines with application to exchange rates. Statistical Modelling 12 (3), 229–255.

Demarta, S., McNeil, A. J., 2005. The t copula and related copulas. International Statistical Review 73 (1), 111–129.

Devroye, L., 1986. Non-Uniform Random Variate Generation. Springer, New York.

Diestel, R., 2006. Graph Theory, 3rd Edition. Springer, Berlin.

Dißmann, J., 2010. Statistical inference for regular vines and application. Diploma thesis, Technische Universität München.

Erdorf, S., Hartmann-Wendels, T., Heinrichs, N., 2011. Diversification in firm valuation: A multivariate copula approach. Cologne Graduate School Working Paper Series 02-01, Cologne Graduate School in Management, Economics and Social Sciences.
URL http://econpapers.repec.org/RePEc:cgr:cgsser:02-01

Fang, H. B., Fang, K. T., Kotz, S., 2002. The meta-elliptical distributions with given marginals. Journal of Multivariate Analysis 82, 1–16.

Fischer, M., Köck, C., Schlüter, S., Weigert, F., 2009. An empirical analysis of multivariate copula models. Quantitative Finance 9 (7), 839–854.

Frahm, G., Junker, M., Szimayer, A., 2003. Elliptical copulas: applicability and limitations. Statistics & Probability Letters 63 (3), 275–286.

Garcia, R., Tsafack, G., 2009. Dependence structure and extreme comovements in international equity and bond markets. CIRANO Working Papers 2009s-21, CIRANO.
URL http://ideas.repec.org/p/cir/cirwor/2009s-21.html

Genest, C., Favre, A.-C., 2007. Everything you always wanted to know about copula modeling but were afraid to ask. Journal of Hydrologic Engineering 12 (4), 347–368.

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., 2011. Parameter estimation for pair-copula constructions. Forthcoming 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., 2011. Efficiently sampling nested Archimedean copulas. Computational Statistics & Data Analysis 55 (1), 57–70.

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: Rüschendorf, L., Schweizer, B., Taylor, M. D. (Eds.), Distributions with fixed marginals and related topics. Institute of Mathematical Statistics, Hayward, pp. 120–141.

Joe, H., 1997. Multivariate Models and Dependence Concepts. Chapman & Hall, London.

Joe, H., Li, H., Nikoloulopoulos, A. K., 2010. Tail dependence functions and vine copulas. Journal of Multivariate Analysis 101 (1), 252–270.

Kazianka, H., Pilz, J., 2011. Bayesian spatial modeling and interpolation using copulas. Computational Geosciences 37, 310–319.
URL http://dx.doi.org/10.1016/j.cageo.2010.06.005

Kurowicka, D., 2011. Optimal truncation of vines. In: Kurowicka, D., Joe, H. (Eds.), Dependence Modeling: Handbook on Vine Copulae. World Scientific Publishing Co., Singapore.

Kurowicka, D., Cooke, R., 2006. Uncertainty Analysis with High Dimensional Dependence Modelling. Wiley, Chichester.

Kurowicka, D., Joe, H. (Eds.), 2011. Dependence Modeling: Handbook on Vine Copulae. World Scientific Publishing Co., Singapore.

Li, D. X., 2000. On default correlation: A copula function approach. Journal of Fixed Income 9 (4), 43–54.

Longin, F., Solnik, B., 1995. Is the correlation in international equity returns constant: 1960-1990? Journal of International Money and Finance 14, 3–26.

Longin, F., Solnik, B., 2001. Extreme correlation of international equity markets. Journal of Finance 56 (2), 649–676.

Manner, H., 2007. Estimation and model selection of copulas with an application to exchange rates. METEOR research memorandum 07/056, Maastricht University.

McNeil, A. J., Frey, R., Embrechts, P., 2005. Quantitative Risk Management: Concepts Techniques and Tools. Princeton University Press, Princeton.

Mendes, B. V. d. M., Semeraro, M. M., Leal, R. P. C., 2010. Pair-copulas modeling in finance. Financial Markets and Portfolio Management 24 (2), 193–213.

Mercier, G., Frison, P.-L., 2009. Statistical characterization of the Sinclair matrix: Application to polarimetric image segmentation. In: IGARSS (3)'09. pp. 717–720.

Min, A., Czado, C., 2010. Bayesian inference for multivariate copulas using pair-copula constructions. Journal of Financial Econometrics 8 (4).

Min, A., Czado, C., 2011. Bayesian model selection for multivariate copulas using pair-copula constructions. Canadian Journal of Statistics 39 (2), 239–258.

Morales-Nápoles, O., 2008. Bayesian belief nets and vines in aviation safety and other applications. Ph.D. thesis, Technische Universiteit Delft.

Morales-Nápoles, O., Cooke, R., Kurowicka, D., 2010. About the number of vines and regular vines on n nodes. Submitted for publication.

Nelsen, R. B., 2005. Dependence modeling with Archimedean copulas. In: Kolev, N., Morettin, P. (Eds.), Proceedings of the Second Brazilian Conference on Statistical Modelling in Insurance and Finance. Institute of Mathematics and Statistics, University of São Paulo, pp. 45–54.

Nelsen, R. B., 2006. An Introduction to Copulas, 2nd Edition. Springer, New York.

Nikoloulopoulos, A. K., Joe, H., Li, H., 2012. Vine copulas with asymmetric tail dependence and applications to financial return data. Forthcoming in Computational Statistics & Data Analysis.

Salinas-Gutiérrez, R., Hernández-Aguirre, A., Villa-Diharce, E. R., 2010. D-vine EDA: A new estimation of distribution algorithm based on regular vines. In: Proceedings of the 12th annual conference on Genetic and evolutionary computation. GECCO '10. ACM, New York, NY, USA, pp. 359–366.
URL http://doi.acm.org/10.1145/1830483.1830550

Salmon, F., 2009. Recipe for disaster: The formula that killed wall street. Wired Magazine 17 (3).
URL http://www.wired.com/techbiz/it/magazine/17-03/wp_quant

Salvadori, G., De Michele, C., Kottegoda, N. T., Rosso, R., 2007. Extremes in Nature: An Approach Using Copulas. Springer, Dordrecht.

Sato, M., Ichiki, K., Takeuchi, T. T., 2010. Precise estimation of cosmological parameters using a more accurate likelihood function. Physical Review Letters 105 (25).

Savu, C., Trede, M., 2010. Hierarchical Archimedean copulas. Quantitative Finance 10, 295–304.

Schölzel, C., Friederichs, P., 2008. Multivariate non-normally distributed random variables in climate research introduction to the copula approach. Nonlinear Processes in Geophysics 15, 761–772.

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, Q. H., 1989. Likelihood ratio tests for model selection and non-nested hypotheses. Econometrica 57 (2), 307–333.

Appendix A. Simulation study

In order to evaluate the approach of sequentially selecting and estimating R-vines proposed in Section 3, we set up a comprehensive simulation study based on the R-vine shown in Figure 1. In total we simulated samples of size 500, 1000 and 2000 according to twelve different scenarios, i.e., twelve different choices of pair-copula families and parameters. We repeated this 1000 times each. The considered scenarios are:

- **all Gaussian, all t, all Gumbel and all Frank R-vines**: all pair-copula families are chosen as Gaussian, Student-t, Gumbel and Frank copulae, respectively. Degrees of freedom of the Student-t copula are linearly increased by 1 for pair-copula terms in higher order trees and start with 3 in the first tree.
- **mixed R-vine**: different families for each pair-copula term.
- **t/mixed R-vine**: Student-t copulae for pair-copulae in first two trees, mixed copulae for remaining pairs. Degrees of freedom of the Student-t copulae are also mixed.

In each of these scenarios, parameters are chosen according to two different settings of Kendall's taus (first, constant values per tree except for increased values of the "central" copulae $c_{2,3}$, $c_{3,6}$ and $c_{2,6|3}$, and second, mixed values; see (A.1) and (A.2), respectively) so that we end up with twelve scenarios. While the R-vine structure array is given by (4), corresponding arrays of Kendall's tau values as well as of copula types for the mixed and t/mixed R-vines are shown in Appendix A.1 below.

Having simulated from the respective true model, we sequentially select and estimate by maximum likelihood estimation an R-vine model as described above and determine the following three quantities to evaluate the adequacy of our selection and estimation approach:

- **general tau-difference**: we compute the mean absolute difference between pairwise empirical Kendall's taus of simulated data from the true and from the selected models. The mean over all repetitions is reported.
- **lower** and **upper tau-difference**: similarly we compute the mean absolute difference between pairwise empirical *lower* and *upper exceedance Kendall's taus* which are defined for two variables U_1 and U_2 as (Brechmann, 2010, Section 3.1.3)

$$\tau^{lower}(U_1, U_2) := \tau(U_1, U_2 | U_1 \leq \delta_1, U_2 \leq \delta_2)$$
$$\tau^{upper}(U_1, U_2) := \tau(U_1, U_2 | U_1 > 1 - \delta_1, U_2 > 1 - \delta_2),$$

  and measure the strength of the joint tail behavior of U_1 and U_2. As thresholds δ_1 and δ_2 we choose $\delta_1 = \delta_2 = 0.2$ as recommended by Brechmann (2010). Again the means over all repetitions are reported.

		Const. Kendall's taus per tree			Mixed Kendall's taus		
	Scenario	lower tau-diff.	general tau-diff.	upper tau-diff.	lower tau-diff.	general tau-diff.	upper tau-diff.
$N = 500$	all Gauss	0.083	0.015	0.083	0.087	0.016	0.087
	all t	0.077	0.019	0.078	0.080	0.020	0.082
	all Gumbel	0.094	0.018	0.066	0.098	0.019	0.073
	all Frank	0.101	0.014	0.100	0.102	0.015	0.101
	mixed	0.090	0.019	0.090	0.090	0.021	0.090
	t/mixed	0.079	0.018	0.080	0.086	0.018	0.084
$N = 1000$	all Gauss	0.058	0.010	0.057	0.061	0.011	0.060
	all t	0.053	0.013	0.054	0.056	0.014	0.057
	all Gumbel	0.066	0.013	0.048	0.068	0.014	0.050
	all Frank	0.077	0.010	0.077	0.075	0.011	0.075
	mixed	0.065	0.016	0.066	0.065	0.017	0.065
	t/mixed	0.056	0.013	0.056	0.058	0.013	0.059
$N = 2000$	all Gauss	0.041	0.007	0.040	0.042	0.008	0.043
	all t	0.038	0.009	0.037	0.040	0.010	0.039
	all Gumbel	0.047	0.009	0.040	0.048	0.010	0.042
	all Frank	0.062	0.008	0.062	0.058	0.008	0.058
	mixed	0.049	0.013	0.050	0.048	0.013	0.048
	t/mixed	0.039	0.010	0.039	0.041	0.010	0.041

Table A.3: Results of the simulation study. The second column indicates the respective scenario for sample sizes of $N = 500$, $N = 1000$ and $N = 2000$. The results corresponding to the first setting of Kendall's tau values are shown in columns 3-5, while those for the second setting are displayed in columns 6-8.

The results of the simulations are shown in Table A.3 and can be summarized as follows.

In terms of all three criteria, the performance improves with increasing sample size due to a higher estimation accuracy and the smaller simulation error. Across both settings of parameters (chosen according to Kendall's tau values), the performance is very similar and only slightly worse in the case of mixed Kendall's taus. According to the general tau-difference criterion, the (non-tail dependent) all Gaussian and all Frank R-vines are identified best. The criteria based on exceedance Kendall's taus show that the all t and the t/mixed R-vines as well as the upper tail of the all Gumbel R-vine are accurately modeled. That is our selection and estimation approach appropriately takes into account the characteristic properties of the copula models.

Comparing the all t, the t/mixed and the mixed scenarios, it is evident that models with larger numbers of Student-t copulae (combined with mixed copulae) can be identified very well. This is in particular true when Kendall's tau values are mixed, which is typical for practical applications.

Appendix A.1. Setting of the simulation study

In the following we show the arrays of Kendall's tau values for parameter choice in the above simulation study as well as the copula type arrays for the mixed and t/mixed scenarios. First, the two settings of Kendall's taus are specified as follows.

- Constant Kendall's taus per tree:

$$\tau_{const} = \begin{pmatrix} 0.05 & & & & & \\ 0.10 & 0.10 & & & & \\ 0.15 & 0.15 & 0.15 & & & \\ 0.20 & 0.20 & 0.20 & 0.20 & & \\ 0.40 & 0.40 & 0.40 & 0.40 & 0.50 & \\ 0.60 & 0.60 & 0.60 & 0.60 & 0.70 & 0.70 \end{pmatrix} \tag{A.1}$$

- Mixed Kendall's taus:

$$\tau_{mixed} = \begin{pmatrix} 0.05 & & & & & \\ 0.10 & 0.10 & & & & \\ 0.15 & 0.15 & 0.15 & & & \\ 0.20 & 0.20 & 0.20 & 0.20 & & \\ 0.25 & 0.30 & 0.35 & 0.40 & 0.45 & \\ 0.50 & 0.55 & 0.60 & 0.65 & 0.70 & 0.75 \end{pmatrix} \tag{A.2}$$

Using abbreviations for copula types (N=Gaussian, t=Student-t, G=Gumbel, SG=Survival Gumbel, F=Frank) the copula type arrays of the mixed and t/mixed scenarios are given by:

- mixed R-vine:

$$T_{mixed} = \begin{pmatrix} N & & & & & \\ F & N & & & & \\ N & F & N & & & \\ G & SG & G & SG & & \\ F & N & F & N & t & \\ SG & G & SG & G & t & t \end{pmatrix}$$

- t/mixed R-vine:

$$T_{t/mixed} = \begin{pmatrix} N & & & & & \\ F & N & & & & \\ N & F & N & & & \\ G & SG & G & SG & & \\ t & t & t & t & t & \\ t & t & t & t & t & t \end{pmatrix}$$