Vine Copula Based Modeling

Czado, Claudia; Nagler, Thomas

DOI
10.1146/annurev-statistics-040220-101153

Publication date
2022

Document Version
Final published version

Published in
Annual Review of Statistics and Its Application

ANNUAL REVIEWS

Annual Review of Statistics and Its Application

Vine Copula Based Modeling

Claudia Czado[1] and Thomas Nagler[2]

[1]Department of Mathematics and Munich Data Science Institute, Technical University of Munich, 85748 Garching bei München, Germany; email: czado@tum.de

[2]Delft Institute of Applied Mathematics, Delft University of Technology, 2628 CD Delft, The Netherlands

Annu. Rev. Stat. Appl. 2022. 9:453–77

First published as a Review in Advance on November 2, 2021

The *Annual Review of Statistics and Its Application* is online at statistics.annualreviews.org

https://doi.org/10.1146/annurev-statistics-040220-101153

Keywords

copula, dependence, distribution, multivariate, tail, vine

Abstract

With the availability of massive multivariate data comes a need to develop flexible multivariate distribution classes. The copula approach allows marginal models to be constructed for each variable separately and joined with a dependence structure characterized by a copula. The class of multivariate copulas was limited for a long time to elliptical (including the Gaussian and *t*-copula) and Archimedean families (such as Clayton and Gumbel copulas). Both classes are rather restrictive with regard to symmetry and tail dependence properties. The class of vine copulas overcomes these limitations by building a multivariate model using only bivariate building blocks. This gives rise to highly flexible models that still allow for computationally tractable estimation and model selection procedures. These features made vine copula models quite popular among applied researchers in numerous areas of science. This article reviews the basic ideas underlying these models, presents estimation and model selection approaches, and discusses current developments and future directions.

1. INTRODUCTION TO COPULAS

For the analysis of large multivariate data sets, flexible multivariate statistical models are required that can adequately describe also the multivariate tail behavior. Standard distributions, such as the multivariate normal or Student's t-distribution, are too inflexible in their marginal and joint behavior. They often require that all univariate and multivariate marginal distributions are of the same type and only allow for highly symmetric dependence structures. These characteristics are rarely satisfied in applications. The copula approach allows us to separate the univariate margins from the dependence structure. In particular, a d-dimensional copula C is a multivariate distribution function on the d-dimensional hypercube $[0, 1]^d$ with uniformly distributed marginals. For an absolutely continuous copula, the corresponding copula density can be obtained by partial differentiation, i.e., $c(u_1, \ldots, u_d) := \frac{\partial^d}{\partial u_1 \ldots \partial u_d} C(u_1, \ldots, u_d)$ for all $\boldsymbol{u}$ in $[0, 1]^d$. Sklar (1959) proved the following fundamental representation theorem.

Theorem 1 (Sklar's Theorem). Let $\boldsymbol{X}$ be a d-dimensional continuous random vector with joint distribution function F, marginal distribution functions F_j, and marginal density functions f_j for $j = 1, \ldots, d$. Then the joint distribution function can be expressed as

$$F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)) \qquad 1.$$

with associated density $f(x_1, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)) f_1(x_1) \cdots f_d(x_d)$ for some d-dimensional copula C with copula density c.

For absolutely continuous distributions the copula C is unique. Equation 1 also holds for discrete random variables; however, the probability mass function is different than the density above. For simplicity, we will work in the continuous case in what follows. Using this theorem, flexible multivariate distributions can be constructed from d-dimensional copulas. By inversion of Equation 1 we can use any d-dimensional distribution function to obtain the corresponding copula. Examples are the Gaussian and the Student's t-copula. Using these copula families together with arbitrary margins results in multivariate distributions that are much more flexible than the multivariate distribution classes used in the inversion. Archimedean copulas are another class of parametric copulas that are built directly using generator functions. The Gumbel, Clayton, and Frank copula families are prime examples. Two-parameter copulas such as the biparameter bivariate (BB) class allow for different, nonzero upper and lower tail behavior and are discussed by Joe (1997, section 5.2). Specific members of this class are enumerated by BB1 to BB8. A nice elementary introduction to copulas is given by Genest & Favre (2007), and more theoretical treatments are the books by Nelsen (2007) and Joe (1997).

From Theorem 1 for $d = 2$, we can immediately derive expressions for the conditional density and distribution functions, which are needed later. In particular, the conditional density $f_{1|2}$ and distribution function $F_{1|2}$ can be expressed as

$$f_{1|2}(x_1|x_2) = c_{12}(F_1(x_1), F_2(x_2)) f_2(x_2), \qquad 2.$$

$$F_{1|2}(x_1|x_2) = \frac{\partial}{\partial F_2(x_2)} C_{12}(F_1(x_1), F_2(x_2)) = \frac{\partial}{\partial v} C_{12}(F_1(x_1), v)|_{v=F_2(x_2)}. \qquad 3.$$

Since vine copulas are built out of bivariate copulas, we now discuss properties of bivariate copulas. To investigate the dependence properties, we consider several dependence measures. Since the Pearson correlation $\rho(X_1, X_2) = \mathrm{Cor}(X_1, X_2)$ is not invariant with respect to monotone transformations of the margins, it is more useful to consider invariant dependence measures, such as

Kendall's τ and Spearman's ρ. In particular, Spearman's rank correlation is defined as the Pearson correlation of the random variables $F_1(X_1)$ and $F_2(X_2)$, i.e., $\rho_s(X_1, X_2) = \mathrm{Cor}(F_1(X_1), F_2(X_2))$. Another popular measure invariant to marginal transformations is Kendall's τ, defined as

$$\tau(X_1, X_2) = P((X_{11} - X_{21})(X_{12} - X_{22}) > 0) - P((X_{11} - X_{21})(X_{12} - X_{22}) < 0),$$

where (X_{11}, X_{12}) and (X_{21}, X_{22}) are independent and identically distributed (i.i.d.) copies of (X_1, X_2). Since $\tau(X_1, X_2)$ and $\rho_s(X_1, X_2)$ are invariant with regard to margins, they depend only on the underlying copula. More specifically, it holds that

$$\tau(X_1, X_2) = 4 \int_0^1 \int_0^1 C(u_1, u_2) dC(u_1, u_2) - 1,$$

$$\rho_s(X_1, X_2) = 12 \int_0^1 \int_0^1 u_1 u_2 dC(u_1, u_2) - 3.$$

In contrast to these central measures of dependence, tail dependence coefficients are used to characterize dependence among extreme events. We consider the probability of the joint occurrence of extremely small or large values and define the upper and lower tail dependence coefficients as

$$\lambda^{\text{upper}} = \lim_{t \to 1^-} P(X_2 > F_2^{-1}(t) | X_1 > F_1^{-1}(t)) = \lim_{t \to 1^-} \frac{1 - 2t + C(t, t)}{1 - t},$$

$$\lambda^{\text{lower}} = \lim_{t \to 0^+} P(X_2 \leq F_2^{-1}(t) | X_1 \leq F_1^{-1}(t)) = \lim_{t \to 0^+} \frac{C(t, t)}{t}.$$

The Gaussian and Frank copulas do not exhibit tail dependence—i.e., $\lambda^{\text{upper}} = \lambda^{\text{lower}} = 0$. The Student's t-copula has symmetric tail dependence—i.e., $\lambda^{\text{upper}} = \lambda^{\text{lower}}$. The Clayton and Gumbel copulas have only lower or upper tail dependence, respectively.

To allow for a visual comparison between different bivariate copula families, marginally normalized contour plots are helpful. For this we consider 3 different scales: the original scale (X_1, X_2), the copula scale $(U_1, U_2) = (F_1(X_1), F_2(X_2))$, and the marginally normalized scale (z-scale) $(Z_1, Z_2) = (\Phi^{-1}(U_1), \Phi^{-1}(U_2))$. Here, Φ denotes the standard normal distribution function. Comparison of contours on the copula scale for different families is difficult, since copula densities are in general unbounded at the corners of $[0, 1]^2$. This is not the case if one works on the z-scale. Here, (Z_1, Z_2) has N(0, 1) margins, and thus any nonelliptical contour shape indicates a deviation from a Gaussian dependence. An example is given in **Figure 1**. The perfect circle in **Figure 1*a*** corresponds to independence, and the elliptical shape in **Figure 1*b*** corresponds to a Gaussian

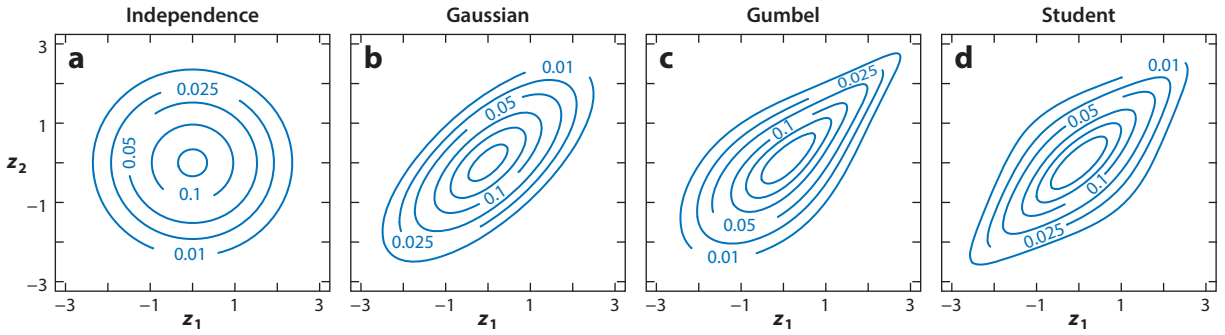

Figure 1

Examples of normalized contour plots: (*a*) independence, (*b*) Gaussian, (*c*) Gumbel, and (*d*) Student's t-copula.

copula. Deviations from Gaussian dependence can be seen in the two right panels—**Figure 1*c***, a Gumbel copula, shows a spike in the upper-right tail, which is an indication of upper tail dependence. **Figure 1*d*** shows a Student's t-copula, which has tail dependence in both the upper and lower tails. To allow for further flexibility of bivariate parametric copulas, their survival and reflection versions can be considered. For example, the survival version of a bivariate copula density c is given by $\bar{c}(u_1, u_2) = c(1 - u_1, 1 - u_2)$. These can be visualized through rotations of the normalized contours (Czado 2019, section 3.6).

We now turn to estimation in the parametric setting. In a copula based model specified by Equation 1, we have to estimate both the marginal and copula parameters. Joint maximum likelihood estimation can be used if the number of parameters is not too large. It is more common to use a two-step approach, however. In a first step, the marginal parameters are estimated based on the i.i.d. observations $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})$ for $i = 1, \ldots, n$. This can be done separately for each of the d margins. In a second step, pseudo copula data are formed by setting

$$\boldsymbol{u}_i = (u_{i1}, \ldots, u_{id}) = (\hat{F}_1(x_{i1}), \ldots, \hat{F}_d(x_{id})), \qquad i = 1, \ldots, n, \tag{4.}$$

from which the copula parameters are estimated. If parametric marginal models are used, we speak of an inference for margins approach (Joe 2005). If the empirical distribution function is used for the margins, we have a semiparametric approach (Genest et al. 1995). Genest et al. (2011) have further proposed to estimate copula parameters by the inversion of empirical Kendall's τ estimates, when there is a one-to-one relationship between τ and the copula parameter. However, this approach is less efficient.

To help the reader to follow all reviewed concepts, we use the abalone data set taken from the University of California–Irvine database (**http://archive.ics.uci.edu/ml/datasets/Abalone**). Abalones are marine snails whose shells have a spiral structure. The data set contains measurements of diameter, height, several types of weight (whole, shucked, viscera, and shell), and the number of rings. In our illustrations, we restrict the data to female abalone shells. The upper-right triangle of **Figure 2** shows scatter plots of pseudo copula data along with their empirical Kendall's τ for the weight variables shuck, viscera (vis), and shell. The empirical Kendall's τ values between shuck and vis, shuck and shell, and vis and shuck are 0.73, 0.65 and 0.69, respectively. Using empirical margins (diagonal of **Figure 2**) shows that the pseudo data (u-scale) are approximate uniform. Normalized pairwise contour plots (z-scale) are shown in the lower-left triangle of panels in **Figure 2**. The panels indicate that dependence is asymmetric, with stronger dependence in the lower-left tail. Since the normalized pairwise contours are not elliptical, a Gaussian copula is not appropriate. We start by searching for an appropriate parametric copula model for the variable pair shuck and shell. We allow for the Gaussian, Student's t, (survival) Clayton, and (survival) Gumbel copula. Corresponding log-likelihood, Akaike information criterion (AIC), and Bayesian information criterion (BIC) values based on the pseudo copula data are given in **Table 1**, which shows that a survival Gumbel model is the best among the studied models.

In the case where copula models do not fit the data, we can also use a nonparametric approach. Transformation kernel estimators were shown to perform best by Nagler et al. (2017) and were implemented in the `kdecopula` and `rvinecopulib` R packages (Nagler 2018, Nagler & Vatter 2020a). Finally, we note that copula estimation based on pseudo copula data requires approximately i.i.d. data. This is usually not the case for multivariate time series data or in the presence of covariates. Here, we advise to first remove the time series or regression structure for each margin by fitting appropriate univariate time series or regression models, respectively. For time series, the resulting standardized residuals can then be used to form the pseudo copula data. In regression models, the fitted conditional response distribution should be used for the transformation to pseudo observations.

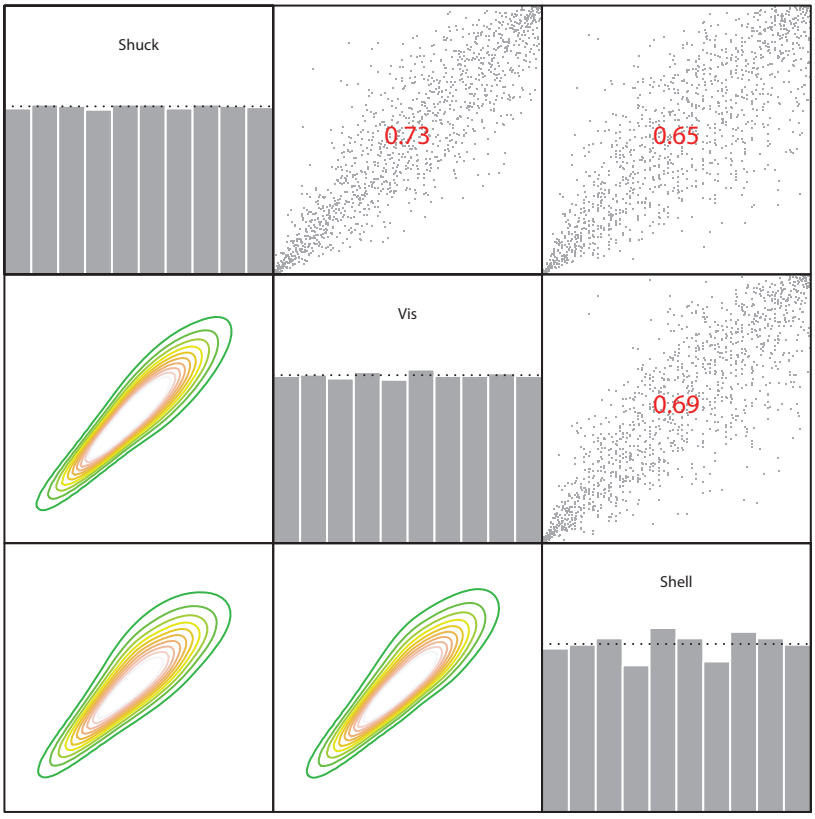

Figure 2

Dependence exploration for the weight variables shuck, viscera (vis), and shell. Along the diagonal are histograms of the pseudo copula data using empirical margins. The upper triangle shows pairwise scatter plots of the pseudo copula data, and the lower triangle shows pairwise normalized contour plots on the z-scale.

2. PAIR COPULA DECOMPOSITIONS AND CONSTRUCTIONS IN THREE DIMENSIONS

While the catalogue of bivariate parametric copula families is large, this is not the case for $d > 2$. The motivation for vine copula models was to find a way to construct multivariate copulas using only bivariate copulas as building blocks. The appropriate tool to obtain such a construction is

Table 1 Parametric copula estimation for the variable pair shuck and shell

Family	Log-likelihood	AIC	BIC
Gaussian	802.79	−1,603.58	−1,598.40
Student's *t*	825.01	−1,646.02	−1,635.68
Clayton	894.73	−1,787.46	−1,782.29
Gumbel	660.07	−1,318.14	−1,312.96
Survival Clayton	464.26	−926.52	−921.35
Survival Gumbel	914.87	−1,827.74	−1,822.57

Abbreviations: AIC, Akaike information criterion; BIC, Bayesian information criterion.

conditioning. Joe (1996) gave the first pair copula construction in terms of distribution functions, while Bedford & Cooke (2001, 2002) independently developed constructions in terms of densities. They also provided a framework to identify all possible constructions. We first illustrate this construction for $d = 3$ by starting with the recursive factorization

$$f(x_1, x_2, x_3) = f_{3|12}(x_3|x_1, x_2) f_{2|1}(x_2|x_1) f_1(x_1) \quad 5.$$

and treat each part separately. Here $F_{j|D}$ and $f_{j|D}$ denote the conditional distribution or density function of X_j given $\boldsymbol{X}_D$, respectively. To determine $f_{3|12}(x_3|x_1, x_2)$ we consider the bivariate conditional density $f_{13|2}(x_1, x_3|x_2)$. The copula density $c_{13;2}(\cdot, \cdot; x_2)$ denotes the copula density associated with the conditional distribution of (X_1, X_3) given $X_2 = x_2$. Using Theorem 1 for $f_{13|2}(x_1, x_3|x_2)$ gives

$$f_{13|2}(x_1, x_3|x_2) = c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2); x_2) f_{1|2}(x_1|x_2) f_{3|2}(x_3|x_2). \quad 6.$$

Now $f_{3|12}(x_3|x_1, x_2)$ is the conditional density of X_3 given $X_1 = x_1, X_2 = x_2$, which can be determined using Equation 2 applied to Equation 6, yielding

$$f_{3|12}(x_3|x_1, x_2) = c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2); x_2) f_{3|2}(x_3|x_2). \quad 7.$$

Finally, direct application of Equation 3 gives

$$f_{2|1}(x_2|x_1) = c_{12}(F_1(x_1), F_2(x_2)) f_2(x_2), \quad 8.$$

$$f_{3|2}(x_3|x_2) = c_{23}(F_2(x_2), F_3(x_3)) f_3(x_3). \quad 9.$$

Inserting Equations 7–9 into Equation 5 yields a pair copula decomposition of an arbitrary three-dimensional density $f(x_1, x_2, x_3)$ as

$$\begin{aligned} f(x_1, x_2, x_3) = {} & c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2); x_2) \times c_{23}(F_2(x_2), F_3(x_3)) \\ & \times c_{12}(F_1(x_1), F_2(x_2)) f_3(x_3) f_2(x_2) f_1(x_1). \end{aligned} \quad 10.$$

We see that the joint density can be expressed in terms of bivariate copula densities, marginal densities, and conditional distribution functions. However, this decomposition is not unique:

$$\begin{aligned} f(x_1, x_2, x_3) = {} & c_{23;1}(F_{2|1}(x_2|x_1), F_{3|1}(x_3|x_1); x_1) \times c_{13}(F_1(x_1), F_3(x_3)) \\ & \times c_{12}(F_1(x_1), F_2(x_2)) f_3(x_3) f_2(x_2) f_1(x_1) \text{ and} \end{aligned} \quad 11.$$

$$\begin{aligned} f(x_1, x_2, x_3) = {} & c_{12;3}(F_{1|3}(x_1|x_3), F_{2|1}(x_2|x_1); x_3) \times c_{13}(F_1(x_1), F_3(x_3)) \\ & \times c_{23}(F_2(x_2), F_3(x_3)) f_3(x_3) f_2(x_2) f_1(x_1) \end{aligned} \quad 12.$$

are two different decompositions using a reordering of the variables in Equation 5.

All decompositions of the density have a conditional copula term of the form $c_{ij;k}(\cdot, \cdot; x_k)$, called a pair copula. To facilitate estimation, we normally neglect the dependence on the specific conditioning value x_k. This is called the simplifying assumption. For a three-dimensional density with copula parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_{12}, \boldsymbol{\theta}_{23}, \boldsymbol{\theta}_{12;3})$, we then get the following simplified pair copula construction:

$$\begin{aligned} f(x_1, x_2, x_3; \boldsymbol{\theta}) = {} & c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2); \boldsymbol{\theta}_{13;2}) \times c_{23}(F_2(x_2), F_3(x_3); \boldsymbol{\theta}_{23}) \\ & \times c_{12}(F_1(x_1), F_2(x_2); \boldsymbol{\theta}_{12}) f_3(x_3) f_2(x_2) f_1(x_1), \end{aligned} \quad 13.$$

where $c_{13;2}(\cdot, \cdot; \boldsymbol{\theta}_{13;2}), c_{12}(\cdot, \cdot; \boldsymbol{\theta}_{12})$ and $c_{23}(\cdot, \cdot; \boldsymbol{\theta}_{23})$ are arbitrary parametric bivariate copula densities. The dependence on marginal parameters has been suppressed to ease notation. This is no longer a decomposition but a construction, where the dependence on x_2 in $c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2); \boldsymbol{\theta}_{13;2})$

is solely captured by the arguments. If the margins in Equation 13 are uniform, we have a three-dimensional parametric copula density.

For the estimation of the parameter $\boldsymbol{\theta}$ based on an i.i.d. sample $\boldsymbol{x}_k = (x_{k1}, x_{k2}, x_{k3}), k = 1, \ldots, n$, we follow the two-step approach discussed in Section 1. We create the associated pseudo data $u_{k,j}, k = 1, \ldots, n, j = 1, \ldots, 3$, as in Equation 4. This allows us to write the joint (pseudo) likelihood for the trivariate copula density associated with Equation 13 as

$$\begin{aligned}\ell(\boldsymbol{\theta}; \boldsymbol{u}) = \prod_{k=1}^{n} & c_{13;2}(C_{1|2}(u_{k,1}|u_{k,2}; \boldsymbol{\theta}_{12}), C_{3|2}(u_{k,1}|u_{k,2}; \boldsymbol{\theta}_{23}); \boldsymbol{\theta}_{13;2}) \\ & \times c_{23}(u_{k,2}, u_{k,3}; \boldsymbol{\theta}_{23}) c_{12}(u_{k,1}, u_{k,2}; \boldsymbol{\theta}_{12}). \end{aligned} \quad 14.$$

Maximizing Equation 14 gives the joint maximum likelihood estimator $\widehat{\boldsymbol{\theta}}$. However, there is an alternative sequential estimation method, which remains computationally tractable in high dimensions. First, we find parameter estimates $\widehat{\boldsymbol{\theta}}_{12}$ and $\widehat{\boldsymbol{\theta}}_{23}$ by maximizing

$$\prod_{k=1}^{n} c_{12}(u_{k,1}, u_{k,2}; \boldsymbol{\theta}_{12}) \quad \text{and} \quad \prod_{k=1}^{n} c_{23}(u_{k,2}, u_{k,3}; \boldsymbol{\theta}_{23})$$

over $\boldsymbol{\theta}_{12}$ and $\boldsymbol{\theta}_{23}$, respectively. Second, we define the pseudo observations

$$u_{k,1|2,\widehat{\boldsymbol{\theta}}_{12}} = C_{1|2}(u_{k,1}|u_{k,2}; \widehat{\boldsymbol{\theta}}_{12}) \quad \text{and} \quad u_{k,3|2,\hat{\boldsymbol{\theta}}_{23}} = C_{3|2}(u_{k,3}|u_{k,2}; \hat{\boldsymbol{\theta}}_{23}), \quad 15.$$

for $k = 1, \ldots, n$. Under the simplifying assumption, these provide an approximate i.i.d. sample from the pair copula $C_{13;2}$. Further, the marginal distribution associated with the pseudo observations Equation 15 is approximately uniform, since the transformation in Equation 15 is a probability integral transform with estimated parameter values. Therefore, we use them to estimate the parameter(s) of the pair copula $c_{13;2}$ by maximizing

$$\prod_{k=1}^{n} c_{13;2}(u_{k,1|2,\widehat{\boldsymbol{\theta}}_{12}}, u_{k,3|2,\widehat{\boldsymbol{\theta}}_{23}}; \boldsymbol{\theta}_{13;2})$$

over $\boldsymbol{\theta}_{13;2}$. This splits the estimation $\boldsymbol{\theta}$ into three simpler problems. The sequential estimate can also be used as a starting value for the joint maximum likelihood estimation. A similar sequential approach can also be followed when estimating pair copulas nonparametrically.

We now estimate all three possible pair copula constructions for the weight variables shuck, vis, and shell in the abalone data set. With the aid of normalized contour plots, we select appropriate pair copula families for the three possible constructions, and the sequential parameter estimation results are contained in **Table 2**. We see that in all three models, dependence is strong in the unconditional copulas but weaker in the conditional copula. We should note that the latter estimates are only valid if the model assumptions are satisfied. In **Table 3**, fit statistics of the three models are compared, showing the superior performance of the pair copula construction $c_{23} - c_{12} - c_{13;2}$. This illustrates that it matters which pair copula construction is fitted to the data.

In more than three dimensions, it will be helpful to represent the density factorization as a graph called a vine tree structure. **Figure 3** shows the graphical representations of the three factorizations given in Equations 10–12 (from left to right). **Figure 3*a*** consists of two levels. In the top level, the nodes labeled 1, 2, and 3 represent the random variables X_1, X_2, and X_3, respectively. They are connected by two edges, 1–2 and 2–3, corresponding to the pair copula terms involving

Table 2 Sequential parameter estimates, selected copula families, and implied dependence measures for the weight variables shuck (1), viscera (vis) (2), and shell (3)

$c_{23} - c_{12} - c_{13;2}$	Term	Family	Parameter(s)	Kendall's τ	λ^{upper}	λ^{lower}
	c_{23}	Survival Gumbel	3.21	0.69	0.00	0.76
	c_{12}	Survival Gumbel	3.65	0.73	0.00	0.79
	$c_{13;2}$	Survival Gumbel	1.23	0.19	0.00	0.24
$c_{13} - c_{12} - c_{23;1}$	Term	Family	Parameter(s)	Kendall's τ	λ^{upper}	λ^{lower}
	c_{13}	Survival Gumbel	2.93	0.66	0.00	0.73
	c_{12}	Survival Gumbel	3.65	0.73	0.00	0.79
	$c_{23;1}$	Survival Gumbel	1.43	0.30	0.00	0.38
$c_{23} - c_{13} - c_{12;3}$	Term	Family	Parameter(s)	Kendall's τ	λ^{upper}	λ^{lower}
	c_{23}	Survival Gumbel	3.21	0.69	0.00	0.76
	c_{13}	Survival Gumbel	2.93	0.66	0.00	0.73
	$c_{12;3}$	Student's t	(0.62, 12.06)	0.43	0.11	0.11

c_{12} and c_{23} in Equation 10. The graph in the bottom level is formed by turning the edges above into nodes and connecting them by an edge. This edge represents the pair copula term involving the conditional density $c_{13;2}(\cdot,\cdot;x_2)$, where the conditioning variable 2 is identified as the common element of the nodes 1, 2 and 2, 3. Hence, each edge in the vine graph is associated with a copula density. The factorization of the joint density corresponding to a given vine graph is then simply the product of marginal densities and all copula densities associated with the edges of the vine tree structure. The graphs in **Figure 3*b,c*** correspond to the factorizations in Equations 11 and 12 in a similar manner.

3. REGULAR VINE COPULAS AND DISTRIBUTIONS

As shown in the previous section, there are three pair copula constructions available for $d = 3$. Similar arguments can be used to derive factorizations of the joint density $f(x_1,\dots,x_d)$ for general d, but additional complications arise. For example, for $d = 4$ we can derive the two factorizations

$$\begin{aligned} &f(x_1,x_2,x_3,x_4)\\ &= c_{14;23}(F_{1|23}(x_1|x_2,x_3),F_{4|23}(x_4|x_2,x_3);x_2,x_3)\\ &\quad\times c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)\times c_{24;3}(F_{2|3}(x_2|x_3),F_{4|3}(x_4|x_3);x_3)\\ &\quad\times c_{34}(F_3(x_3),F_4(x_4))\times c_{23}(F_2(x_2),F_3(x_3))\times c_{12}(F_1(x_1),F_2(x_2))\\ &\quad\times f_4(x_4)f_3(x_3)f_2(x_2)f_1(x_1) \end{aligned} \qquad 16.$$

Table 3 Fit statistics for three possible pair copula constructions for the weight variables shuck (1), viscera (vis) (2), and shell (3)

Model	Log-likelihood	Number of parameters	AIC	BIC
$c_{23} - c_{12} - c_{13;2}$	2253.91	3	−4501.82	−4486.29
$c_{13} - c_{12} - c_{23;1}$	2247.47	3	−4488.95	−4473.42
$c_{13} - c_{23} - c_{12;3}$	2253.25	4	−4498.50	−4477.80

Abbreviations: AIC, Akaike information criterion; BIC, Bayesian information criterion.

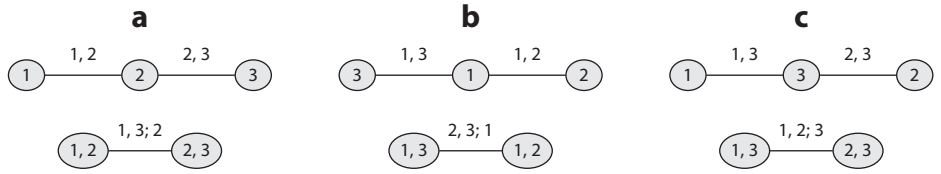

Figure 3

Graphical representation of the three pair copula constructions (*a*) $c_{23} - c_{12} - c_{13;\,2}$, (*b*) $c_{13} - c_{12} - c_{23;\,1}$, and (*c*) $c_{13} - c_{23} - c_{12;\,3}$.

and

$$
\begin{aligned}
& f(x_1, x_2, x_3, x_4) \\
& = c_{34;12}(F_{3|12}(x_3|x_1, x_2), F_{4|12}(x_4|x_1, x_2); x_1, x_2) \\
& \quad \times c_{23;1}(F_{2|1}(x_2|x_1), F_{3|1}(x_3|x_1); x_1) \times c_{24;1}(F_{2|1}(x_2|x_1), F_{4|1}(x_4|x_1); x_1) \\
& \quad \times c_{14}(F_1(x_1), F_4(x_4)) \times c_{13}(F_1(x_1), F_3(x_3)) \times c_{12}(F_1(x_1), F_2(x_2)) \\
& \quad \times f_4(x_4) f_3(x_3) f_2(x_2) f_1(x_1).
\end{aligned} \tag{17.}
$$

The factorizations are represented as regular vine (R-vine) tree structures in **Figure 4**, where **Figure 4*a*** corresponds to Equation 16 and **Figure 4*b*** to Equation 17. In **Figure 4*a***, each graph level consists of a path. In contrast, in **Figure 4*b***, each graph level consists of a star (where one node is connected to all the others).

By permuting the variable indices, the two types of vine structures generate 12 factorizations each for a total of 24 (compared with only three possible factorizations for $d = 3$). When $d \geq 5$, the subgraphs are no longer restricted to be paths or stars and the number of possible decompositions increases superexponentially in d (Morales-Nápoles 2011). Furthermore, it becomes increasingly difficult to verify whether a factorization like those in Equation 16 and 17 actually represents a valid density. To characterize and organize all valid factorizations, Bedford & Cooke (2001, 2002) developed a convenient graphical tool called an R-vine tree structure. An R-vine consists of linked trees, where the edges in one tree become the nodes of the next.

More formally, recall that a tree is a connected acyclic graph $T = (N, E)$ with node set N and edge set E. The set of graphs $\mathcal{V} = (T_1, \ldots, T_{d-1})$ is an R-vine tree sequence on d elements if

1. T_1 is a tree with node set $N_1 = \{1, \ldots, d\}$ and edge set E_1.
2. For $j \geq 2$, T_j is a tree with node set $N_j = E_{j-1}$ and edge set E_j.
3. For $j = 2, \ldots, d-1$ and $\{a, b\} \in E_j$ it must hold that $|a \cap b| = 1$ (proximity condition).

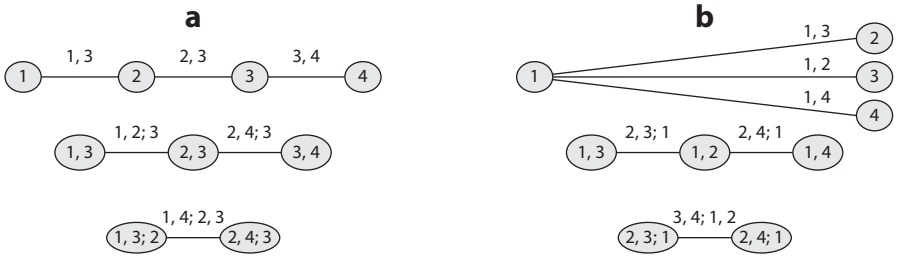

Figure 4

The two types of regular vine structures in four dimensions.

The proximity condition states that an edge between nodes in tree T_j is only possible if the corresponding edges in T_{j-1} share a common node.

The R-vine tree structure corresponding to Equation 10 has nodes $N_1 = \{1, 2, 3\}$ and edges $E_1 = \{\{1, 2\}, \{2, 3\}\}$ in tree T_1 and nodes $N_2 = \{\{1, 2\}, \{2, 3\}\}$ and edges $E_2 = \{\{1, 2\}, \{2, 3\}\}$ in tree T_2. Since this set notation quickly becomes unwieldy, a simpler notation is needed. For any edge $e \in E_i$, define the complete union

$$A_e = \left\{ j \in N_1 | \exists\, e_1 \in E_1, \ldots, e_{i-1} \in E_{i-1} \text{ such that } j \in e_1 \in \ldots \in e_{i-1} \in e \right\}.$$

The conditioning set D_e of an edge $e = \{a, b\}$ is defined as $D_e := A_a \cap A_b$, and the conditioned sets $\mathcal{C}_{e,a}$ and $\mathcal{C}_{e,b}$ are given by

$$\mathcal{C}_{e,a} = A_a \setminus D_e \text{ and } \mathcal{C}_{e,b} = A_b \setminus D_e.$$

Kurowicka & Cooke (2006) showed that $\mathcal{C}_{e,a}$ and $\mathcal{C}_{e,b}$ are singletons, so we often abbreviate the edge $e = (\mathcal{C}_{e,a}, \mathcal{C}_{e,b}; D_e)$ as $e = (a_e, b_e; D_e)$. For example, the edge $e = \{a = \{1, 2\}, b = \{2, 3\}\}$ has $A_a = \{1, 2\}$ and $A_b = \{2, 3\}$, and therefore $D_e = \{2\}$. It follows that $\mathcal{C}_{e,a} = \{1\}$ and $\mathcal{C}_{e,b} = \{3\}$, resulting in $e = (1, 3; 2)$.

We have already seen two common special cases in **Figure 4** for $d = 4$. In a canonical (C-)vine, all trees are stars: In every tree there is a single node, called the root, that connects all the others. A specific C-vine can be identified by the order of root nodes. Another example of a C-vine is shown in the middle panels of **Figure 5**, where the root nodes are 1, (1, 4), (6, 4; 1), (6, 2; 4, 1), and (5, 2; 6, 4, 1). Since all indices from previous root nodes are contained in the label of later root nodes, we can also specify this order by only referencing the index that enters in the next tree. For example, the root node sequence above can be written as 1, 4, 6, 2, 5, 3. In a drawable (D-)vine tree structure, all trees are paths. More formally, this means that all but two nodes have exactly two neighbors. The two nodes with only one neighbor are called leaves. For a D-vine tree sequence, the proximity condition implies that the specification of tree T_1 determines all other trees. To characterize a D-vine structure, we therefore only need to specify the path in the first tree, called the order of a D-vine. The right panels in **Figure 5** show a D-vine with order 3–5–6–2–1–4 (or, equivalently, 4–1–2–6–5–3). The tree structure of a D-vine can be drawn in a way that resembles a grapevine, which is why Bedford & Cooke (2001) called the linked tree sequence a vine.

After the building plan is defined, we can construct an R-vine distribution for a d-dimensional random vector $\boldsymbol{X} = (X_1, \ldots, X_d)$. This distribution is specified by the triplet $(\mathcal{F}, \mathcal{V}, \mathcal{B})$ with the following:

1. Marginal distributions: $\mathcal{F} = (F_1, \ldots, F_d)$ is a vector of continuous marginal distribution functions of the random variables $X_1, \ldots, X_d$.
2. R-vine tree sequence: $\mathcal{V}$ is an R-vine tree sequence on d elements.
3. Bivariate copulas: The set $\mathcal{B} = \{C_e | e \in E_i; i = 1, \ldots, d-1\}$, where C_e is a bivariate copula with density. Here, E_i is the edge set of tree T_i in the R-vine tree sequence $\mathcal{V}$.
4. Relation between R-vine tree sequence and the set of bivariate copulas: For each $e \in E_i, i = 1, \ldots, d-1, e = \{a, b\}$, $C_e(\cdot, \cdot)$ is the copula associated with the conditional distribution of $X_{\mathcal{C}_{e,a}}$ and $X_{\mathcal{C}_{e,b}}$ given $\boldsymbol{X}_{D_e}$.

The copula C_e corresponding to edge e is denoted by $C_{\mathcal{C}_{e,a}\mathcal{C}_{e,b};D_e}$ with corresponding density $c_{\mathcal{C}_{e,a}\mathcal{C}_{e,b};D_e}$. This copula is also called a pair copula. We already employed the simplifying assumption above: We assume that $C_e(\cdot, \cdot)$ does not depend on the specific value of $\boldsymbol{X}_{D_e}$. In nonsimplified models, we would need a separate set $\mathcal{B}$ of bivariate copulas in item 3 for every possible value of the random vector $\boldsymbol{X}$. This means that for every edge e, the pair copula C_e would depend on

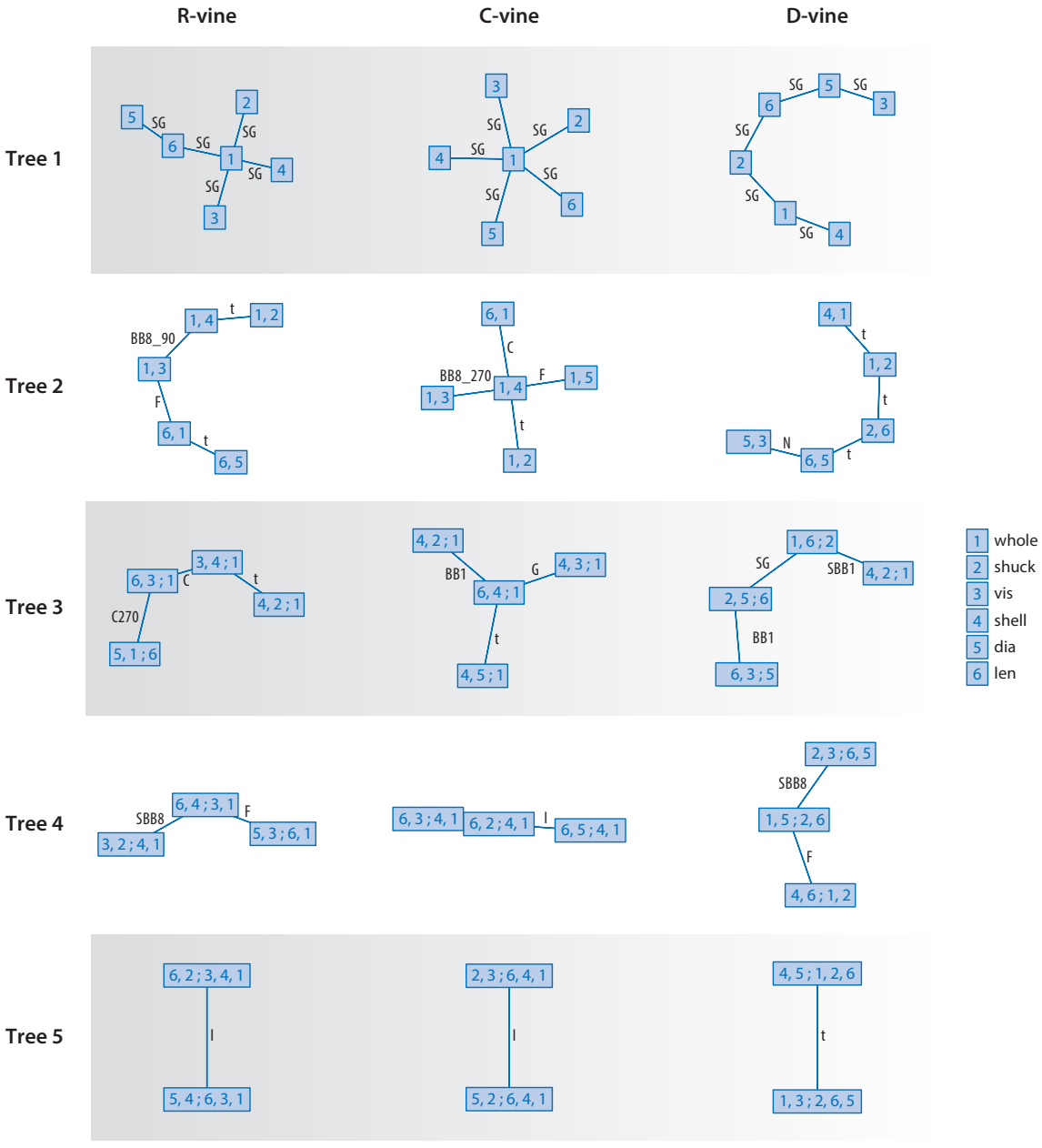

Figure 5

Vine tree structures for variables whole, shuck, viscera (vis), shell, diameter (dia), and length (len) from the female abalone data set. Abbreviations: BB, biparameter bivariate; BB8_90, reflected 90° degree BB8 copula; BB8_270, reflected 270° BB8 copula; C, Clayton copula; C270, reflected 270° Clayton copula; C-vine, canonical vine; D-vine, drawable vine; F, Frank copula; G, Gumbel copula; I, independence copula; N, Gaussian copula; R-vine, regular vine; SBB1, survival BB1 copula; SBB8, survival BB8 copula; SG, survival Gumbel copula; t, Student's *t*-copula.

the value of $\boldsymbol{X}_{D_e}$. For simplicity, we only consider the simplified case. For the existence of such distributions, Bedford & Cooke (2002) showed the following result.

Theorem 2 (existence of an R-vine distribution). Assume that $(\mathcal{F}, \mathcal{V}, \mathcal{B})$ satisfy the properties in items 1–3 above; then, there is a valid d-dimensional distribution F with density

$$f_{1,\dots d}(x_1,\dots x_d) = f_1(x_1) \times \cdots \times f_d(x_d) \times \prod_{i=1}^{d-1} \prod_{e \in E_i} c_{\mathcal{C}_{e,a}\mathcal{C}_{e,b};D_e}(F_{\mathcal{C}_{e,a}|D_e}(x_{\mathcal{C}_{e,a}}|\boldsymbol{x}_{D_e}), F_{\mathcal{C}_{e,b}|D_e}(x_{\mathcal{C}_{e,b}}|\boldsymbol{x}_{D_e})), \quad 18.$$

such that for each $e \in E_i, i = 1,\dots,d-1$, with $e = \{a, b\}$, we have for the distribution function of $X_{\mathcal{C}_{e,a}}$ and $X_{\mathcal{C}_{e,b}}$ given $\boldsymbol{X}_{D_e}$

$$F_{\mathcal{C}_{e,a}\mathcal{C}_{e,b}|D_e}\left(x_{\mathcal{C}_{e,a}}, x_{\mathcal{C}_{e,b}}|\boldsymbol{x}_{D_e}\right) = C_e\left(F_{\mathcal{C}_{e,a}|D_e}(x_{\mathcal{C}_{e,a}}|\boldsymbol{x}_{D_e}), F_{\mathcal{C}_{e,b}|D_e}(x_{\mathcal{C}_{e,b}}|\boldsymbol{x}_{D_e})\right).$$

Furthermore, the one-dimensional margins of F are given by $F_i(x_i), i = 1,\dots,d$.

If all margins are standard uniform, we call the resulting distribution an R-vine copula. Let D be a set of indices from $\{1,\dots,d\}$ not including i and j. The copula associated with the bivariate conditional distribution (X_i, X_j) given that $\boldsymbol{X}_D = \boldsymbol{x}_D$ is denoted by $C_{ij;D}(\cdot,\cdot)$. In contrast, the conditional distribution function of (U_i, U_j) given $\boldsymbol{U}_D = \boldsymbol{u}_D$ is expressed as $C_{ij|D}(\cdot,\cdot|\boldsymbol{u}_D)$. This is, in general, not a copula.

Equation 18 involves conditional distribution functions as arguments of the pair copula densities. These can be determined recursively from conditional distributions associated with the pair copulas in the model. In particular, Joe (1996) showed the following result: Let X be a random variable and $\boldsymbol{Y}$ be a random vector with absolutely continuous joint distribution. Let Y_j a component of $\boldsymbol{Y}$ and denote the subvector of $\boldsymbol{Y}$ with Y_j removed by $\boldsymbol{Y}_{-j}$. In this case the conditional distribution of X given $\boldsymbol{Y} = \boldsymbol{y}$ satisfies the following recursion:

$$F_{X|\boldsymbol{Y}}(\cdot|\boldsymbol{y}) = \frac{\partial C_{X,Y_j;\boldsymbol{Y}_{-j}}(F_{X|\boldsymbol{Y}_j}(x|\boldsymbol{y}_{-j}), F_{Y_j|\boldsymbol{Y}_j}(y|\boldsymbol{y}_{-j}))}{\partial F_{Y_j|\boldsymbol{Y}_{-j}}(y_j|\boldsymbol{y}_{-j})}, \quad 19.$$

where $C_{X,Y_j;\boldsymbol{Y}_{-j}}(\cdot,\cdot)$ is the copula corresponding to (X, Y_j) given $\boldsymbol{Y}_{-j}$.

For C- and D-vine tree sequences, we call the associated R-vine distributions C- and D-vine distributions, respectively. Omitting arguments, their densities are given by

$$f_{1,\dots,d}(x_1,\dots,x_d) = \left[\prod_{j=1}^{d-1}\prod_{i=1}^{d-j} c_{j,j+i;1,\dots,j-1}\right] \times \left[\prod_{k=1}^{d} f_k(x_k)\right] \text{ and}$$

$$f_{1,\dots,d}(x_1,\dots,x_d) = \left[\prod_{j=1}^{d-1}\prod_{i=1}^{d-j} c_{i,(i+j);(i+1)\dots,(i+j-1)}\right] \cdot \left[\prod_{k=1}^{d} f_k(x_k)\right].$$

Using the first column of **Figure 5**, we can express the associated R-vine distribution as

$$f^{rv}_{1,\dots,6} = c_{12}c_{13}c_{14}c_{16}c_{56}c_{15;6}c_{36;1}c_{34;1}c_{24;1}c_{23;14}c_{46;13}c_{35;16}$$
$$c_{45;136}c_{26;134}c_{24;1346}f_1f_2f_3f_4f_5f_6$$

and the C- and D-vine tree structures corresponding to the middle and right columns of **Figure 5**, respectively, by

$$f^{cv}_{1,\dots,6} = c_{12}c_{13}c_{14}c_{15}c_{16}c_{24;1}c_{34;1}c_{45;1}c_{46;1}c_{26;14}c_{36;14}c_{56;14} \\ c_{23;146}c_{25;146}c_{35;1246}f_1 f_2 f_3 f_4 f_5 f_6 \text{ and} \quad 20.$$

$$f^{dv}_{1,\dots,6} = c_{41}c_{12}c_{26}c_{56}c_{35}c_{24;1}c_{16;2}c_{25;6}c_{36;5}c_{23;65}c_{15;26}c_{46;12} \\ c_{45;126}c_{13;256}c_{34;1256}f_1 f_2 f_3 f_4 f_5 f_6. \quad 21.$$

If all or some of the variables are discrete, similar vine-based factorizations of the joint probability density/mass function can be derived. We refer readers to Panagiotelis et al. (2012), Stöber (2013), and Stöber et al. (2015) for more details.

4. ESTIMATION AND SELECTION OF VINE COPULA MODELS UNDER THE SIMPLIFYING ASSUMPTION

As mentioned in the previous section, a stepwise approach to estimation is more tractable in higher dimensions. Here, we first estimate the marginal distribution functions and use them to create pseudo copula data. These copula data are then used to estimate an appropriate vine copula. For this vine copula model we have to solve three problems of increasing complexity:

1. Given the vine tree sequence and pair copula families, estimate the copula parameters.
2. Given the vine tree sequence and a catalogue of pair copula families, select the best family and estimate the corresponding parameters for each edge in the vine.
3. Select the vine tree structure and the pair copula families and estimate the corresponding parameters for each edge.

We can solve problem 1 using the sequential estimation approach discussed in Section 2. For this we extend the construction of the pseudo data from tree T_2 given in Equation 15 to trees T_3 to T_{d-1} using estimated conditional copula distribution functions. This approach is very fast since it allows the estimation of the parameters of each pair copula term separately, starting from tree T_1 until T_{d-1}. The asymptotic properties of such parameter estimators, including standard errors, are studied by Haff et al. (2013), Stöber & Schepsmeier (2013), and Schepsmeier & Stöber (2014). For problem 2, we can also proceed sequentially and consider each pair copula separately. We fit the parameters for each family in the catalogue and choose the one that minimizes the AIC or BIC.

Problem 3 is the most challenging, since the number of vine tree structures grows as $d! \times 2^{(d-2)(d-3)/2-1}$ (Morales-Nápoles 2011). For example, the number of R-vine tree structures for $d = 10$ is approximately 5×10^{14}. Even if one wants to restrict to C- and D-vines, there are almost 2 million structures to be investigated. Dissmann et al. (2013) develop a greedy selection algorithm based on the idea of fitting the strongest dependencies first. This is natural since estimation errors are propagated in the sequential estimation approach and we may hope to find sparse models. To measure the strength of dependence, the empirical Kendall's τ is used. The Dissmann algorithm selects tree T_1 by using a maximal spanning tree algorithm with the absolute value of empirical Kendall's τ between any pair of variables as weights. Once tree T_1 is determined, all pair copula families and parameters are selected and estimated using the approaches outlined in problems 1 and 2. For tree T_2, all possible edges allowed by the proximity condition are considered. This identifies pairs of variables in the pseudo data as defined in Equation 15. The absolute empirical

Table 4 Sequential parameter estimates, selected copula families, and implied dependence measures for six variables of the female abalone data set

Tree	Edge[a]	Family	Parameter(s)	Kendall's τ	λ^{upper}	λ^{lower}
1	6,5	Survival Gumbel	6.39	0.84	0.00	0.89
1	6,1	Survival Gumbel	5.20	0.81	0.00	0.86
1	1,3	Survival Gumbel	4.92	0.80	0.00	0.85
1	1,4	Survival Gumbel	4.82	0.79	0.00	0.85
1	1,2	Survival Gumbel	5.64	0.82	0.00	0.87
2	5,1;6	Student's t	(0.39, 6.24)	0.25	0.11	0.11
2	6,3;1	Frank	0.53	0.06	0.00	0.00
2	3,4;1	BB8 90°	(−1.37, −0.97)	−0.15	0.00	0.00
2	4,2;1	Student's t	(−0.65, 5.46)	−0.45	0.00	0.00
3	5,3;6,1	Clayton 270°	−0.10	−0.05	0.00	0.00
3	6,4;3,1	Clayton	0.09	0.05	0.00	0.00
3	3,2;4,1	Student's t	(−0.41, 4.93)	−0.27	0.01	0.01
4	5,4;6,3,1	Frank	1.47	0.16	0.00	0.00
4	6,2;3,4,1	Survival BB8	(1.61, 0.85)	0.15	0.00	0.00
5	5,2;6,3,4,1	Independence	–	0.00	0.00	0.00

[a]Variables are whole (1), shuck (2), vis (3), shell (4), dia (5), and len (6).
Abbreviations: BB, biparameter bivariate; dia, diameter; len, length; vis, viscera.

Kendall's τ for these pairs is used as a weight for selecting the maximal spanning tree for T_2. We can again select pair copula families and estimate parameters as in problem 2. We continue that way until all trees, pair copula families, and parameters are selected and estimated. This approach can be adapted for C- and D-vine structures. For a C-vine, we choose the root node as the one maximizing the sum of absolute empirical Kendall's τ, as illustrated by Czado et al. (2012). For D-vines, we search for the path that maximizes the sum of absolute empirical Kendall's τ. The above procedures also work with nonparametric pair copulas (for a survey and comparison of available methods, see Nagler et al. 2017) and censored data (Barthel et al. 2017, 2018).

We illustrate the modeling process using the variables whole, shuck, vis, shell, diameter (dia) and length (len) of the abalone data set. In a preliminary step, nonparametric estimates of the marginal distribution functions are used to create pseudo copula data. Then we apply Dissmann's algorithm to fit R-, C-, and D-vine copula models to the pseudo data. The fitted vine tree structures are shown in **Figure 5**. **Table 4** contains the selected pair copula families and estimated parameters for the R-vine copula model. As expected, the fitted dependence strength decreases as we move from T_1 to T_5. As a benchmark, we also fit a Gaussian vine copula model. This is equivalent to a Gaussian copula but is more general than a multivariate Gaussian model, since we allow for non-Gaussian marginal distributions. **Table 5** compares the different vine copula models and shows that the R-vine copula model provides the best fit. In particular, a Gaussian copula model is clearly insufficient.

5. COMPUTATIONAL ASPECTS

The flexibility of R-vine copula models comes with computational challenges. First, the number of pair copulas grows quadratically in the dimension, which necessitates efficient algorithms for inference and simulation. Second, such algorithms must deal with the huge number of possible vine tree structures. Most of these issues have been addressed by the research community in

Table 5 Fit statistics for several vine copula models for variables whole, shuck, vis, shell, dia, and len from the female abalone data set

Copula model	Log-likelihood	Number of parameters	AIC	BIC
R-vine copula	9,097.57	17	−18,161.14	−18,073.17
Gaussian vine copula	8,257.21	14	−16,486.43	−16,413.98
C-vine copula	9,097.85	17	−18,159.70	−18,066.56
D-vine copula	9,073.32	22	−18,102.65	−17,988.80

Abbreviations: AIC, Akaike information criterion; BIC, Bayesian information criterion; C-vine, canonical vine; D-vine, drawable vine; dia, diameter; len, length; R-vine, regular vine; vis, viscera.

the past, but the algorithms are difficult to implement for nonexperts. Hence, the availability of user-friendly software libraries was key to the popularity of vine copula models. These libraries include the R packages `VineCopula` (Nagler et al. 2020b) and `rvinecopulib` (Nagler & Vatter 2020a), the MATLAB toolboxes `VineCopulaMatlab` (Kurz 2015) and `MATvines` (Coblenz 2021), the C++ library `vinecopulib` (Nagler & Vatter 2020c), and Python libraries `pyvinecopulib` (Vatter et al. 2020) and `pyvines` (Yuan & Hu 2019). These relieve applied researchers from algorithmic difficulties, but they still need to translate the model into a form the software can work with.

Morales-Nápoles (2011) developed a compact representation of the vine tree structure in the form of a triangular matrix. This representation was later used by Dissmann et al. (2013) to encode entire vine copula models. For example, the tree sequence in the left column of **Figure 5** can be translated into the array

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

The label of the jth edge in tree t is given by $(m_{j,d-j+1}, m_{t,j}; m_{t-1,j}, \ldots, m_{1,j})$. Less formally,

- start with the counter-diagonal element of column j (first conditioned variable),
- jump up to the element in row t of column j (second conditioned variable), and
- gather all entries further up than row t in column j (conditioning set).

For example, the first column encodes the edges (5, 2; 4, 3, 1, 6), (5, 4; 3, 1, 6), (5, 3; 1, 6), (5, 1; 6), and (5, 6). Dissmann et al. (2013) derived conditions that ensure the array corresponds to a valid R-vine tree sequence.

For parametric vine copula models, the pair copula families and parameters can be stored in similar arrays. For example, if $\theta_{t,j}$ is the parameter of the jth edge in the tth tree, we can store the model parameters in the array

$$\Theta = \begin{pmatrix} \theta_{1,1} & \theta_{1,2} & \theta_{1,3} & \theta_{1,4} & \theta_{1,5} \\ \theta_{2,1} & \theta_{2,2} & \theta_{2,3} & \theta_{2,4} & \\ \theta_{3,1} & \theta_{3,2} & \theta_{3,3} & & \\ \theta_{4,1} & \theta_{4,2} & & & \\ \theta_{5,1} & & & & \end{pmatrix}.$$

In software libraries, the arrays are usually written as square matrices, where empty entries in the matrix are filled with zeros by convention. For our example, this would mean using a

6×6 matrix for both M and Θ. The orientation of the matrices above is arbitrary, and different software libraries adopt different conventions. The upper-left triangular form above is used by the `vinecopulib` library and its interfaces. In this format, the (t, j) entry of the matrices is the pivotal element for the jth edge of tree t, which is intuitive and simplifies indexing in algorithms. In several other libraries, the rows of the matrix are reversed, making it lower-left triangular. The software provided by Joe (2014) reverses the column order, making it upper-right triangular.

Vine copula models are often used for simulation, which is achieved through the Rosenblatt transform (Rosenblatt 1952) and its inverse. The Rosenblatt transform turns a random vector $\boldsymbol{U} = (U_1, \ldots, U_d)$ with copula C into another vector $\boldsymbol{V} = (V_1, \ldots, V_d) = R(\boldsymbol{U})$ containing independent uniform variables. It is given by

$$V_j = C_{j|j-1,\ldots,1}(U_j|U_{j-1}, \ldots, U_1), \quad j = 1, \ldots, d,$$

where $C_{j|D}$ is the conditional distribution of U_j given $\boldsymbol{U}_D$. The corresponding inverse operation $\boldsymbol{U} = R^{-1}(\boldsymbol{V})$ turns independent uniform variables $\boldsymbol{V}$ into a vector $\boldsymbol{U}$ with copula C. It is given by

$$U_j = C^{-1}_{j|j-1,\ldots,1}(V_j|V_{j-1}, \ldots, V_1), \quad j = 1, \ldots, d.$$

In vine copula models, the conditional distributions and inverses appearing in the transformations can be computed efficiently using recursion over conditional distributions associated to the pair copulas in the model [see Equation 19 and Czado (2019, chapter 6)]. The same technique can be used to simulate conditionally from a vine copula model, provided the required conditional distributions can be expressed by pair copula terms without the need for integration (see, for example, Aas et al. 2021, section 4.2). A specialized implementation for D- and C-vines is given in the `CDVineCopulaConditional` library (Bevacqua 2017).

6. CURRENT AND FUTURE RESEARCH DIRECTIONS

In what follows, we summarize the literature of four particularly active areas of research and point to future directions, mainly focusing on methodological developments. Further topics and applications of vine copulas were reviewed recently by Czado (2019, chapter 11) and Aas (2016); readers are also directed to **vine-copula.org** for a broad collection of papers, talks, and videos.

6.1. Statistical Learning

In recent years, vine copula models have been used increasingly in statistical learning problems. One of the main tasks in statistical learning is regression. In the context of vine copulas, regression problems are solved by building conditional response distributions. This allows us to extract, among others, conditional means and conditional quantiles.

A first question is how to best set up the vine structure in such a context. One approach builds a vine distribution for the covariates alone and adds the response in a way that the resulting conditional distribution is available in closed form, i.e., without the need for integration over the vine copula density (Chang & Joe 2019, Chang et al. 2019, Cooke et al. 2019). These approaches focus on the vine structure of the covariates first, which may be unnatural in the regression context. An alternative is to start with the response as the first vine node and then add in such a way that the conditional distribution remains available in closed form. The selection procedure stops if the conditional (penalized) log-likelihood is no longer increasing. This idea was first developed for D-vine models by Kraus & Czado (2017a) and later extended to certain R-vine structures by Zhu et al. (2021). Tepegjozova et al. (2021) considered both D- and C-vines but generalized the procedure to look two steps ahead before adding a new variable. While the D- and C-vine methods are

feasible in large dimensions, the search for R-vines is restricted to smaller dimensions because of the huge number of possibilities. How to select the covariates in high-dimensional R-vine copula regression models therefore remains an open problem.

In the context of quantile regression, we need to adequately model tail dependence. The classical approach is the linear quantile regression model of Koenker & Bassett (1978). Bernard & Czado (2015) showed that for univariate Gaussian margins, the Gaussian copula is the only one complying with linear regression quantiles. Thus, linear quantile regression is of little use for data with tail dependence. A further problem is that linear quantile lines usually cross for different levels if the true multivariate distribution is not Gaussian. Noh et al. (2015) formulated a copula based quantile regression approach, which was later extended to censored response variables (De Backer et al. 2017) and other regression problems (Nagler & Vatter 2020b). In these papers, the conditional response distribution is not necessarily available in closed form but is found through copula based estimating equations. In contrast, Kraus & Czado (2017a) proposed a D-vine model where the conditional quantile function can be computed by inverting the conditional response distribution function, which is available in closed form. Extensions including ordinal discrete variables and nonparametric pair copulas were considered by Schallhorn et al. (2017) and Tepegjozova et al. (2021).

Another important task of statistical learning is clustering. For model-based clustering, finite mixtures of distributions are often assumed. Here, the random vector $\boldsymbol{X} = (X_1, \ldots, X_d)$ has a density given by

$$f(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{j=1}^{k} \pi_j f_j(\boldsymbol{x}; \boldsymbol{\theta}_j), \qquad 22.$$

where $\pi_j \geq 0$, $\sum_{j=1}^{k} \pi_j = 1$, and $f_j(\cdot; \boldsymbol{\theta}_j)$ is the density of the jth cluster with parameters $\boldsymbol{\theta}_j$. The task is to estimate the unknown mixing probabilities $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$ and the unknown parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k)$, as well as the number of clusters k. Given a fitted mixture model, observations are assigned to the cluster with the highest conditional probability given the data. Estimation in finite mixture models is facilitated with the expectation–maximization (EM) algorithm of Dempster et al. (1977). Mixtures of normals are the most prominent models (Fraley & Raftery 2002) but require cluster distributions to have elliptical shape. To achieve nonelliptical cluster shapes, Diday (2002) utilized copulas and adapted the EM algorithm. A restrictive d-dimensional Clayton copula was used by Cuvelier & Noirhomme-Fraiture (2005). Vrac et al. (2005) used the expectation/conditional maximization (ECM) algorithm of Meng & Rubin (1993) together with the Frank copula to jointly cluster bivariate atmospheric profiles.

There has been some work connecting finite mixture models to vine copulas. Markov tree models are equivalent to vine copulas truncated after the first tree (see, e.g., Brechmann et al. 2012) and have been considered by Kirshner (2008) and Silva & Gramacy (2009). Kim et al. (2013) considered parametric D-vine copulas with a single pair copula family and estimated parameters jointly in the maximization step (M-step), which is only tractable for small d. Roy & Parui (2014) use mixtures of D-vine distributions for the analysis of observed principal components, where the node order is determined by the magnitude of the eigenvalues. In contrast, Sun et al. (2017) considered C-vine copulas as mixture components and sequential estimation for the mixture components. In the M-step, the stepwise selection and estimation approach for C-vines outlined in Section 4 was used. However, their method neglects the fact that the choice of the C-vine copula families and parameters might depend on the weights resulting from the expectation step (E-step). Sahin & Czado (2021) extended these methods and allow selection and estimation of different R-vine distributions for each cluster based on the ECM algorithm. All methods above implicitly assume

that all variables in the data set are relevant for the clustering. In high dimensions, this may not be the case, and improvements can be expected when restricting to only relevant variables. How to select this set of variables is an open problem, however.

The mixture model in Equation 22 can also be used for classification problems. Here, we assign observations to the class with the highest posterior probability. Classification algorithms based on this rule are called Bayes classifiers. Copula based Bayes classifiers were considered by Elidan (2012) using Bayesian networks and Salinas-Gutiérrez et al. (2017) using chain graph models. Chen (2016), Carrera et al. (2016), and Carrera et al. (2019) employed parametric vine copulas for the mixture components. Nagler & Czado (2016) used nonparametric pair copulas, and Schellhase & Spanhel (2018) allowed for nonsimplified vines with a penalized spline approach. Tekumalla et al. (2017) used D-vine models but, in contrast to the previous methods, allowed for multiple classes and discrete variables through a computationally challenging Bayesian approach. Most of these works show good performance compared with other competitors but do not address important methodological issues. In particular, variable selection and hyperparameter tuning in vine copula based classifiers remain an open problem.

6.2. Structure Selection and High-Dimensional Models

In Section 3, we saw that there are many possible R-vine structures. In principle, classical likelihood-based criteria (like AIC or BIC) can be used to determine the best structure. However, because the number of possible vines grows superexponentially (see Section 4), such procedures become infeasible even in moderate dimensions. This explosion and the complex algebraic structure of vine tree sequences make their selection an extremely challenging problem.

Dissmann's heuristic, mentioned in Section 4, greedily maximizes the dependence in each tree based on the absolute value of Kendall's τ. Variants replacing Kendall's τ with AIC, BIC, or p-values of goodness-of-fit tests were investigated by Czado et al. (2013), variants with nonparametric pair copulas by Nagler et al. (2017), and a variant taking the simplifying assumption into account by Kraus & Czado (2017b). The empirical studies in these papers indicate small improvements in performance, but these gains appear negligible in view of the added computational demand. Nonheuristic methods based on Markov chain Monte Carlo (Gruber & Czado 2015, 2018) and neural networks (Sun et al. 2019) lead to more notable improvements but take orders of magnitude longer to compute and are therefore only feasible in rather small dimension. In view of the above, Dissmann's heuristic remains the gold standard, even a decade after it was initially developed in his thesis. A promising path for improvement is to find ways to efficiently explore the space of vine structures. A first step along these lines based on the concept of common sampling orders was taken by Cooke et al. (2015) and Zhu et al. (2020). Another approach, by Chang et al. (2019), limits the greediness of the algorithm by looking ahead a fixed number of trees using Monte Carlo tree search. Although these approaches improve significantly over the benchmark on various data sets, there remains much room for improvement—especially in higher dimensions.

In high-dimensional models, further challenges for model selection arise. The number of model parameters grows quadratically in the dimension, which calls for sparse vine copula models. A vine copula model is sparse when many pair copulas correspond to (conditional) independence. A natural subclass is truncated vine copulas, i.e., models where all pair copulas after a certain tree are set to independence. The question then becomes how to select the right truncation level, and several solutions have been proposed (Kurowicka 2011, Brechmann et al. 2012, Brechmann & Joe 2015, Joe 2018). An alternative model class is thresholded vine copulas, where all pairs with sufficiently weak dependence are set to independence. Traditionally, the thresholding is done by an independence test based on Kendall's τ (e.g., Dissmann et al. 2013), as implemented in the

VineCopula package. A modified BIC criterion specifically tailored to high-dimensional vines was proposed by Nagler et al. (2019) and is suitable to select both truncation level and thresholding level. This method is implemented in the vinecopulib library. A separate line of research (Müller & Czado 2018, 2019a,b) exploits connections between vine copulas and Gaussian directed acyclic graphs to find sparsity patterns. A completely different approach is to use dimension reduction techniques before employing a copula model (as in Tagasovska et al. 2019). In high-dimensional linear models, it is well known that theoretical guarantees for estimation and inference break when the number of parameters is large compared with the number of samples. There is no reason to believe that vine copula models are exempt from this phenomenon, but it is unclear where the breakpoint lies. The only known theoretical results in this regime were given in the model selection context by Nagler et al. (2019), but they rely on an unverified assumption of consistent parameter estimates. Thus, we are in dire need of asymptotic results for inference in high-dimensional vine copula models.

6.3. The Simplifying Assumption

As mentioned in Sections 2 and 3, vine copula models are usually built under the simplifying assumption. The assumption is much weaker than conditional independence. Conditional dependence of any strength and form is allowed, but only if it does not change with the conditioning value. It is worth emphasizing that the simplifying assumption is only required for explicit pairwise dependencies in the model, i.e., those corresponding to an edge in the vine. All other pairwise dependencies can (and often will) be nonsimplified. An obvious question, first raised by Hobæk Haff et al. (2010), is how constraining the simplifying assumption is. Some popular multivariate parametric copulas, most notably elliptical copulas and the Clayton copula, are known to satisfy the simplifying assumption (Stöber et al. 2013) in a very strong sense: Due to their closedness in conditioning and marginalization, they are simplified for every possible R-vine structure. Hobæk Haff et al. (2010) provided numerical examples suggesting that a simplified parametric model can be a very good approximation even if the true model is not simplified.

To check whether the simplifying assumption holds for a particular data set, several statistical tests have been developed. Most of them focus on a simpler setting, where there is only one bivariate copula and one or more covariates (e.g., Acar et al. 2013, Derumigny & Fermanian 2017, Gijbels et al. 2017). The problem is more difficult in the context of vine copulas because the assumption needs to be tested for every pair copula. A procedure tailored to parametric vine copulas was developed by Kurz & Spanhel (2018) and implemented in the pacotest R package (Kurz 2017). The authors performed an empirical study suggesting that the assumption cannot be rejected in several financial data sets but is already rejected in the second tree for some others. Even if the assumption is violated, Portier & Segers (2018) showed that a simplified empirical copula can be estimated nonparametrically at $\sqrt{n}$-rate, as suggested by simulations in the context of simplified vines by Haff & Segers (2015). In a similar vein, Nagler & Czado (2016) proved that simplified vine copula densities can be estimated at a rate equivalent to a two-dimensional problem. Their simulations suggest that, in a nonparametric setting, the simplifying assumption can be beneficial even if it is severely violated.

It is also possible to build nonsimplified models. Acar et al. (2012) developed a kernel method for three-dimensional semiparametric vines. A fully parametric model, where the conditional dependence is modeled similar to a generalized linear model, was proposed by Han et al. (2017). A semiparametric model allowing for more general, additive relationships was developed by Vatter & Nagler (2018) and is implemented in the gamCopula R package (Vatter & Nagler 2017). A fully nonparametric method based on splines and dimension reduction of the covariate space was

proposed by Schellhase & Spanhel (2018) and is available through the `pencopulaCond` R package (Schellhase 2017).

The fact that vine copula models contain multiple interrelated pair copulas complicates interpretation of the simplifying assumption. Killiches et al. (2017) used visualizations of three-dimensional density contours to provide some geometrical intuition. However, Spanhel et al. (2019) showed in several toy examples that the simplifying assumption can have several unintuitive implications. First, using the true pair copulas in the first tree is often suboptimal. Second, spurious dependence can appear in higher tree levels, which makes interpretation of those pair copulas difficult. Mroz et al. (2021) further proved that any copula can be approximated arbitrarily well by a simplified vine copula in the supremum metric, but not for more intuitive notions of distance (like Kullback–Leibler divergence or total variation metric) and the distance can be quite large. This alone should not discourage researchers from employing a simplified model. The exact validity of a model's assumptions is not the primary determinant of its usefulness, but some care is in order when interpreting the results.

6.4. Time Series Models

Time series models based on vine copulas have become quite popular, especially in financial econometrics. In the context of a multivariate time series $\boldsymbol{X}_1, \dots, \boldsymbol{X}_T \in \mathbb{R}^d$, there are two types of dependence: serial dependence (dependence across different points in time) and cross-sectional dependence (dependence in the vector $\boldsymbol{X}_t$ at a given point in time).

The majority of works separate the two types of dependence. Classical time series models are used to filter the univariate serial dependence in each margin and a vine copula model is employed for the cross-sectional dependence across residuals. A first instance of such a model with ARMA-GARCH filters was given in the seminal paper by Aas et al. (2009) and in higher dimensions by Brechmann & Czado (2013). Later, Min & Czado (2014) additionally allowed for nonparametric estimation of the marginal residual distribution and provided asymptotic results for the copula parameter estimates. Other marginal filters can be used as well—for example, in the presence of nonstationary trends (e.g., Jäger et al. 2019).

In most early works, the cross-sectional dependence was assumed to be fixed. Chollete et al. (2009) relaxed this assumption by allowing the parameters of a C-vine copula to switch between two regimes depending on a latent Markov process. This model was generalized by Stöber & Czado (2014) and Fink et al. (2017) to multiple regimes, each corresponding to a different R-vine copula. Other works modelled the change of dependence on a finer scale by explicitly specifying the dynamics of copula parameters. In So & Yeung (2014), each parameter follows observation-driven dynamics similar to the DCC-GARCH model (Engle 2002). Another line of research considers model-driven dynamics, where copula parameters are modeled as latent autoregressive models of order 1 [AR(1)] processes. Almeida et al. (2016) and Goel & Mehra (2019) developed frequentist estimation procedures for C- and D-vine models. Kreuzer & Czado (2019, 2021) proposed Bayesian approaches for general R-vine models and parsimonious factor copulas expressed as C-vines. A different approach was taken by Vatter & Nagler (2018) and Acar et al. (2019), where the model parameters are modeled as smooth functions of time and estimated by spline and kernel techniques, respectively.

The separation of serial and cross-sectional dependence is convenient because it allows us to rely on well-established models for univariate series. But serial dependence can be characterized by a copula, too, and conceptually there is no reason to treat the serial part differently from the cross-sectional one. Around the same time, three papers independently proposed vine copula models that capture both types of dependence simultaneously: the copula autoregressive (COPAR) model of

Brechmann & Czado (2015), the D-vine model of Smith (2015), and the M-vine model of Beare & Seo (2015). The three models differ in the vine structure, but their motivation is the same. The idea is to choose a structure that ensures stationarity of the model by imposing a simple restriction on the pair copulas: If we shift all indices of an edge in time, the associated pair copula must remain the same. This idea was formalized by Beare & Seo (2015) and called translation invariance. Nagler et al. (2020a) gave a complete characterization of the class of vine structures that ensure stationarity under translation invariance. Surprisingly, the COPAR model does not fall under this category and, hence, may not be stationary. Nagler et al. (2020a) also derived asymptotic properties of parameter estimates and simulation-based predictions from such models. They also pointed to an open problem regarding to the mixing properties of the resulting time series. Another interesting venue for future research is to adapt these models for nonstationary and long-memory time series.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

Claudia Czado acknowledges the support of the German Science Foundation (DFG grant CZ 86/6-1).

LITERATURE CITED

Aas K. 2016. Pair-copula constructions for financial applications: a review. *Econometrics* 4(4):43

Aas K, Czado C, Frigessi A, Bakken H. 2009. Pair-copula constructions of multiple dependence. *Insur. Math. Econ.* 44(2):182–98

Aas K, Nagler T, Jullum M, Løland A. 2021. Explaining predictive models using Shapley values and non-parametric vine copulas. arXiv:2102.06416 [stat.ME]

Acar EF, Craiu RV, Yao F, et al. 2013. Statistical testing of covariate effects in conditional copula models. *Electron. J. Stat.* 7:2822–50

Acar EF, Czado C, Lysy M. 2019. Flexible dynamic vine copula models for multivariate time series data. *Econom. Stat.* 12:181–97

Acar EF, Genest C, Nešlehová J. 2012. Beyond simplified pair-copula constructions. *J. Multivar. Anal.* 110:74–90

Almeida C, Czado C, Manner H. 2016. Modeling high-dimensional time-varying dependence using dynamic D-vine models. *Appl. Stoch. Models Bus. Ind.* 32(5):621–38

Barthel N, Geerdens C, Czado C, Janssen P. 2017. Modeling recurrent event times subject to right-censoring with D-vine copulas. arXiv:1712.05845 [stat.ME]

Barthel N, Geerdens C, Killiches M, Janssen P, Czado C. 2018. Vine copula based likelihood estimation of dependence patterns in multivariate event time data. *Comput. Stat. Data Anal.* 117:109–27

Beare BK, Seo J. 2015. Vine copula specifications for stationary multivariate Markov chains. *J. Time Ser. Anal.* 36(2):228–46

Bedford T, Cooke RM. 2001. Probability density decomposition for conditionally dependent random variables modeled by vines. *Ann. Math. Artif. Intell.* 32:245–68

Bedford T, Cooke RM. 2002. Vines: a new graphical model for dependent random variables. *Ann. Stat.* 30(4):1031–68

Bernard C, Czado C. 2015. Conditional quantiles and tail dependence. *J. Multivar. Anal.* 138:104–26

Bevacqua E. 2017. `CDVineCopulaConditional`: sampling from conditional C- and D-vine copulas. *R Package*, version 0.1.0. **https://CRAN.R-project.org/package=CDVineCopulaConditional**

Brechmann EC, Czado C. 2013. Risk management with high-dimensional vine copulas: an analysis of the Euro Stoxx 50. *Stat. Risk Model.* 30(4):307–42

Brechmann EC, Czado C. 2015. COPAR—multivariate time series modeling using the copula autoregressive model. *Appl. Stoch. Models Bus. Ind.* 31(4):495–514

Brechmann EC, Czado C, Aas K. 2012. Truncated regular vines and their applications. *Can. J. Stat.* 40(1):68–85

Brechmann EC, Joe H. 2015. Truncation of vine copulas using fit indices. *J. Multivar. Anal.* 138:19–33

Carrera D, Bandeira L, Santana R, Lozano JA. 2019. Detection of sand dunes on mars using a regular vine-based classification approach. *Knowl. Based Syst.* 163:858–74

Carrera D, Santana R, Lozano JA. 2016. Vine copula classifiers for the mind reading problem. *Prog. Artif. Intell.* 5(4):289–305

Chang B, Joe H. 2019. Prediction based on conditional distributions of vine copulas. *Comput. Stat. Data Anal.* 139:45–63

Chang B, Pan S, Joe H. 2019. Vine copula structure learning via Monte Carlo tree search. *Proc. Mach. Learn. Res.* 89:353–61

Chen Y. 2016. A copula-based supervised learning classification for continuous and discrete data. *J. Data Sci.* 14(4):769–82

Chollete L, Heinen A, Valdesogo A. 2009. Modeling international financial returns with a multivariate regime-switching copula. *J. Financ. Econom.* 7(4):437–80

Coblenz M. 2021. MATVines: a vine copula package for MATLAB. *SoftwareX* 14:100700

Cooke RM, Joe H, Chang B. 2019. Vine copula regression for observational studies. *AStA Adv. Stat. Anal.* 104:141–67

Cooke RM, Kurowicka D, Wilson K. 2015. Sampling, conditionalizing, counting, merging, searching regular vines. *J. Multivar. Anal.* 138:4–18

Cuvelier E, Noirhomme-Fraiture M. 2005. Clayton copula and mixture decomposition. In *Applied Stochastic Models and Data Analysis (ASMDA 2005)*, ed. J Janssen, P Lenca, pp. 699–708. Bretagne, Fr.: ENST

Czado C. 2019. *Analyzing Dependent Data with Vine Copulas: A Practical Guide with R*. New York: Springer

Czado C, Brechmann EC, Gruber L. 2013. Selection of vine copulas. In *Copulae in Mathematical and Quantitative Finance*, ed. P Jaworski, F Durante, WK Härdle, pp. 17–37. New York: Springer

Czado C, Schepsmeier U, Min A. 2012. Maximum likelihood estimation of mixed C-vines with application to exchange rates. *Stat. Model.* 12(3):229–55

De Backer M, El Ghouch A, Van Keilegom I. 2017. Semiparametric copula quantile regression for complete or censored data. *Electron. J. Stat.* 11(1):1660–98

Dempster AP, Laird NM, Rubin DB. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* 39(1):1–22

Derumigny A, Fermanian JD. 2017. About tests of the "simplifying" assumption for conditional copulas. *Depend. Model.* 5(1):154–97

Diday E. 2002. Mixture decomposition of distributions by copulas. In *Classification, Clustering, and Data Analysis: Recent Advances and Applications*, ed. K Jajuga, A Sokołowski, H-H Bock, pp. 297–310. New York: Springer

Dissmann J, Brechmann EC, Czado C, Kurowicka D. 2013. Selecting and estimating regular vine copulae and application to financial returns. *Comput. Stat. Data Anal.* 59:52–69

Elidan G. 2012. Copula network classifiers (CNCs). *Proc. Mach. Learn. Res.* 22:346–54

Engle R. 2002. Dynamic conditional correlation: a simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *J. Bus. Econ. Stat.* 20(3):339–50

Fink H, Klimova Y, Czado C, Stöber J. 2017. Regime switching vine copula models for global equity and volatility indices. *Econometrics* 5(1):3

Fraley C, Raftery AE. 2002. Model-based clustering, discriminant analysis, and density estimation. *J. Am. Stat. Assoc.* 97(458):611–31

Genest C, Favre AC. 2007. Everything you always wanted to know about copula modeling but were afraid to ask. *J. Hydrol. Eng.* 12(4):347–68

Genest C, Ghoudi K, Rivest L. 1995. A semi-parametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika* 82:543–52

Genest C, Nešlehová J, Ben Ghorbal N. 2011. Estimators based on Kendall's tau in multivariate copula models. *Aust. N. Z. J. Stat.* 53(2):157–77
Gijbels I, Omelka M, Veraverbeke N. 2017. Nonparametric testing for no covariate effects in conditional copulas. *Statistics* 51(3):475–509
Goel A, Mehra A. 2019. Analyzing contagion effect in markets during financial crisis using stochastic autoregressive canonical vine model. *Comput. Econ.* 53(3):921–50
Gruber L, Czado C. 2015. Sequential Bayesian model selection of regular vine copulas. *Bayesian Anal.* 10(4):937–63
Gruber L, Czado C. 2018. Bayesian model selection of regular vine copulas. *Bayesian Anal.* 13(4):1111–35
Haff IH. 2013. Parameter estimation for pair-copula constructions. *Bernoulli* 19(2):462–91
Haff IH, Segers J. 2015. Nonparametric estimation of pair-copula constructions with the empirical pair-copula. *Comput. Stat. Data Anal.* 84:1–13
Han D, Tan KS, Weng C. 2017. Vine copula models with GLM and sparsity. *Commun. Stat. Theory Methods* 46(13):6358–81
Hobæk Haff I, Aas K, Frigessi A. 2010. On the simplified pair-copula construction—simply useful or too simplistic? *J. Multivar. Anal.* 101(5):1296–310
Jäger WS, Nagler T, Czado C, McCall RT. 2019. A statistical simulation method for joint time series of non-stationary hourly wave parameters. *Coastal Eng.* 146:14–31
Joe H. 1996. Families of m-variate distributions with given margins and $m(m\text{-}1)/2$ bivariate dependence parameters. In *Distributions with Fixed Marginals and Related Topics*, ed. L Rüschendorf, B Schweizer, MD Taylor, pp. 120–41. N.p.: Inst. Math. Stat.
Joe H. 1997. *Multivariate Models and Dependence Concepts*. London: Chapman and Hall
Joe H. 2005. Asymptotic efficiency of the two stage estimation method for copula-based models. *J. Multivar. Anal.* 94:401–19
Joe H. 2014. *Dependence Modeling with Copulas*. Boca Raton, FL: Chapman and Hall/CRC
Joe H. 2018. Parsimonious graphical dependence models constructed from vines. *Can. J. Stat.* 46(4):532–55
Killiches M, Kraus D, Czado C. 2017. Examination and visualisation of the simplifying assumption for vine copulas in three dimensions. *Aust. N. Z. J. Stat.* 59(1):95–117
Kim D, Kim JM, Liao SM, Jung YS. 2013. Mixture of D-vine copulas for modeling dependence. *Comput. Stat. Data Anal.* 64:1–19
Kirshner S. 2008. Learning with tree-averaged densities and distributions. In *NIPS'07: Proceedings of the 20th International Conference on Neural Information Processing Systems*, ed. JC Platt, D Koller, Y Singer, ST Roweis, pp. 761–68. Red Hook, NY: Curran
Koenker R, Bassett G. 1978. Regression quantiles. *J. Econom. Soc.* 46:33–50
Kraus D, Czado C. 2017a. D-vine copula based quantile regression. *Comput. Stat. Data Anal.* 110:1–18
Kraus D, Czado C. 2017b. Growing simplified vine copula trees: improving Dißmann's algorithm. arXiv:1703.05203 [stat.ME]
Kreuzer A, Czado C. 2019. Bayesian inference for dynamic vine copulas in higher dimensions. arXiv:1911.00702 [stat.ME]
Kreuzer A, Czado C. 2021. Bayesian inference for a single factor copula stochastic volatility model using Hamiltonian Monte Carlo. *Econom. Stat.* 19:130–50
Kurowicka D. 2011. Optimal truncation of vines. In *Dependence Modeling: Vine Copula Handbook*, ed. D Kurowicka, H Joe, pp. 233–47. Singapore: World Sci.
Kurowicka D, Cooke R. 2006. *Uncertainty Analysis with High Dimensional Dependence Modelling*. New York: Wiley
Kurz M. 2015. `VineCopulaMATLAB:` a MATLAB toolbox for vine copulas based on C++. *MATLAB Toolbox*. **https://maltekurz.github.io/VineCopulaMatlab/**
Kurz MS. 2017. `pacotest:` Testing for partial copulas and the simplifying assumption in vine copulas. *R package*, version 0.2.2. **https://CRAN.R-project.org/package=pacotest**
Kurz MS, Spanhel F. 2018. Testing the simplifying assumption in high-dimensional vine copulas. arXiv:1706.02338 [stat.ME]
Meng XL, Rubin DB. 1993. Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika* 80(2):267–78

Min A, Czado C. 2014. SCOMDY models based on pair-copula constructions with application to exchange rates. *Comput. Stat. Data Anal.* 76:523–35

Morales-Nápoles O. 2011. Counting vines. In *Dependence Modeling: Vine Copula Handbook*, ed. D Kurowicka, H Joe, pp. 189–218. Singapore: World Sci.

Mroz T, Fuchs S, Trutschnig W. 2021. How simplifying and flexible is the simplifying assumption in pair-copula constructions—analytic answers in dimension three and a glimpse beyond. *Electron. J. Stat.* 15(1):1951–92

Müller D, Czado C. 2018. Representing sparse Gaussian DAGs as sparse R-vines allowing for non-Gaussian dependence. *J. Comput. Graph. Stat.* 27(2):334–44

Müller D, Czado C. 2019a. Dependence modelling in ultra high dimensions with vine copulas and the graphical lasso. *Comput. Stat. Data Anal.* 137:211–32

Müller D, Czado C. 2019b. Selection of sparse vine copulas in high dimensions with the lasso. *Stat. Comput.* 29(2):269–87

Nagler T. 2018. kdecopula: an R package for the kernel estimation of bivariate copula densities. *J. Stat. Softw.* 84(7):1–22

Nagler T, Bumann C, Czado C. 2019. Model selection in sparse high-dimensional vine copula models with an application to portfolio risk. *J. Multivar. Anal.* 172:180–92

Nagler T, Czado C. 2016. Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas. *J. Multivar. Anal.* 151:69–89

Nagler T, Krüger D, Min A. 2020a. Stationary vine copula models for multivariate time series. arXiv:2008.05990 [stat.ME]

Nagler T, Schellhase C, Czado C. 2017. Nonparametric estimation of simplified vine copula models: comparison of methods. *Depend. Model.* 5:99–120

Nagler T, Schepsmeier U, Stöber J, Brechmann EC, Graeler B, Erhardt T. 2020b. `VineCopula`: statistical inference of vine copulas. *R package*, version 2.4.1. **https://CRAN.R-project.org/package=VineCopula**

Nagler T, Vatter T. 2020a. rvinecopulib: high performance algorithms for vine copula modeling. *R package*, version 0.5.5.1.1. **https://CRAN.R-project.org/package=rvinecopulib**

Nagler T, Vatter T. 2020b. Solving estimating equations with copulas. arXiv:1801.10576 [stat.ME]

Nagler T, Vatter T. 2020c. `vinecopulib`: high performance algorithms for vine copula modeling. *C++ library*, version 0.5.5

Nelsen RB. 2007. *An Introduction to Copulas*. New York: Springer

Noh H, El Ghouch A, Van Keilegom I. 2015. Semiparametric conditional quantile estimation through copula-based multivariate models. *J. Bus. Econ. Stat.* 33(2):167–78

Panagiotelis A, Czado C, Joe H. 2012. Pair copula constructions for multivariate discrete data. *J. Am. Stat. Assoc.* 107(499):1063–72

Portier F, Segers J. 2018. On the weak convergence of the empirical conditional copula under a simplifying assumption. *J. Multivar. Anal.* 166:160–81

Rosenblatt M. 1952. Remarks on a multivariate transformation. *Ann. Math. Stat.* 23(3):470–72

Roy A, Parui SK. 2014. Pair-copula based mixture models and their application in clustering. *Pattern Recognit.* 47(4):1689–97

Sahin Ö, Czado C. 2021. Vine copula mixture models and clustering for non-Gaussian data. arXiv:2102.03257 [stat.ME]

Salinas-Gutiérrez R, Hernández-Quintero A, Dalmau-Cedeño O, Pérez-Daz ÁP. 2017. Modeling dependencies in supervised classification. In *Mexican Conference on Pattern Recognition*, ed. JA Carrasco-Ochoa, JF Martínez-Trinidad, JA Olvera-López, pp. 117–26. New York: Springer

Schallhorn N, Kraus D, Nagler T, Czado C. 2017. D-vine quantile regression with discrete variables. arXiv:1705.08310 [stat.ME]

Schellhase C. 2017. pencopulaCond: estimating non-simplified vine copulas using penalized splines. *R package*, version 0.2

Schellhase C, Spanhel F. 2018. Estimating non-simplified vine copulas using penalized splines. *Stat. Comput.* 28(2):387–409

Schepsmeier U, Stöber J. 2014. Derivatives and Fisher information of bivariate copulas. *Stat. Pap.* 55(2):525–42

Silva R, Gramacy R. 2009. MCMC methods for Bayesian mixtures of copulas. *Proc. Mach. Learn. Res.* 5:512–19
Sklar A. 1959. Fonctions de répartition à *n* dimensions et leurs marges. *Publ. Inst. Stat. Univ. Paris* 8:229–31
Smith MS. 2015. Copula modelling of dependence in multivariate time series. *Int. J. Forecast.* 31(3):815–33
So MK, Yeung CY. 2014. Vine-copula GARCH model with dynamic conditional dependence. *Comput. Stat. Data Anal.* 76:655–71
Spanhel F, Kurz MS, et al. 2019. Simplified vine copula models: approximations based on the simplifying assumption. *Electron. J. Stat.* 13(1):1254–91
Stöber J. 2013. *Regular vine copulas with the simplifying assumption, time-variation, and mixed discrete and continuous margins*. Ph.D. Thesis, Dep. Math., Tech. Univ. Munich, Munich, Ger.
Stöber J, Czado C. 2014. Regime switches in the dependence structure of multidimensional financial data. *Comput. Stat. Data Anal.* 76:672–86
Stöber J, Hong HG, Czado C, Ghosh P. 2015. Comorbidity of chronic diseases in the elderly: patterns identified by a copula design for mixed responses. *Comput. Stat. Data Anal.* 88:28–39
Stöber J, Joe H, Czado C. 2013. Simplified pair copula constructions—limitations and extensions. *J. Multivar. Anal.* 119:101–18
Stöber J, Schepsmeier U. 2013. Estimating standard errors in regular vine copula models. *Comput. Stat.* 28(6):2679–707
Sun M, Konstantelos I, Strbac G. 2017. C-vine copula mixture model for clustering of residential electrical load pattern data. *IEEE Trans. Power Syst.* 32(3):2382–93
Sun Y, Cuesta-Infante A, Veeramachaneni K. 2019. Learning vine copula models for synthetic data generation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pp. 5049–57. Palo Alto, CA: AAAI
Tagasovska N, Ackerer D, Vatter T. 2019. Copulas as high-dimensional generative models: vine copula autoencoders. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, ed. H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox, R Garnett. Red Hook, NY: Curran
Tekumalla LS, Rajan V, Bhattacharyya C. 2017. Vine copulas for mixed data: multi-view clustering for mixed data beyond meta-Gaussian dependencies. *Mach. Learn.* 106(9):1331–57
Tepegjozova M, Zhou J, Claeskens G, Czado C. 2021. Nonparametric C-and D-vine based quantile regression. arXiv:2102.04873 [stat.ME]
Vatter T, Nagler T. 2017. `gamCopula:` generalized additive models for bivariate conditional dependence structures and vine copulas. *R package*, version 0.0-4. **https://CRAN.R-project.org/package=gamCopula**
Vatter T, Nagler T. 2018. Generalized additive models for pair-copula constructions. *J. Comput. Graph. Stat.* 27(4):715–27
Vatter T, Nagler T, Arabas S. 2020. `pyvinecopulib`. *Python library*, version 0.5.5. **https://pypi.org/project/pyvinecopulib**
Vrac M, Chédin A, Diday E. 2005. Clustering a global field of atmospheric profiles by mixture decomposition of copulas. *J. Atmos. Ocean. Technol.* 22(10):1445–59
Yuan Z, Hu T. 2019. `pyvine:` the Python package for regular vine copula modeling, sampling and testing. *Commun. Math. Stat.* 9:53–86
Zhu K, Kurowicka D, Nane GF. 2020. Common sampling orders of regular vines with application to model selection. *Comput. Stat. Data Anal.* 142:106811
Zhu K, Kurowicka D, Nane GF. 2021. Simplified R-vine based forward regression. *Comput. Stat. Data Anal.* 155:107091