Stat Comput (2018) 28:323–341
https://doi.org/10.1007/s11222-017-9733-y

Model distances for vine copulas in high dimensions

Matthias Killiches[1] · **Daniel Kraus**[1] · **Claudia Czado**[1]

Received: 20 April 2016 / Accepted: 30 January 2017 / Published online: 11 February 2017

Abstract Vine copulas are a flexible class of dependence models consisting of bivariate building blocks and have proven to be particularly useful in high dimensions. Classical model distance measures require multivariate integration and thus suffer from the curse of dimensionality. In this paper, we provide numerically tractable methods to measure the distance between two vine copulas even in high dimensions. For this purpose, we consecutively develop three new distance measures based on the Kullback–Leibler distance, using the result that it can be expressed as the sum over expectations of KL distances between univariate conditional densities, which can be easily obtained for vine copulas. To reduce numerical calculations, we approximate these expectations on adequately designed grids, outperforming Monte Carlo integration with respect to computational time. For the sake of interpretability, we provide a baseline calibration for the proposed distance measures. We further develop similar substitutes for the Jeffreys distance, a symmetrized version of the Kullback–Leibler distance. In numerous examples and applications, we illustrate the strengths and weaknesses of the developed distance measures.

Keywords Vine copulas · Model distances · Kullback–Leibler · Jeffreys distance · Monte Carlo integration

1 Introduction

In the course of growing data sets and increasing computing power, statistical data analysis has considerably developed within the last decade. The necessity of proper dependence modeling has become evident at least since the financial crisis of 2007. Using vine copulas is a popular option to approach this task. Bedford and Cooke (2002) described how multivariate distributions can be sequentially decomposed into bivariate building blocks via conditioning. Since the seminal paper of Aas et al. (2009), which developed statistical inference for this method, many aspects of vines have been studied: Dißmann et al. (2013) provide a sequential estimation algorithm for vines, Panagiotelis et al. (2012) treat vine copulas for discrete data and Nagler and Czado (2016) examine nonparametric vine copulas. Further, there have been various applications to data from many fields such as finance (Maya et al. 2015; Kraus and Czado 2017), sociology (Cooke et al. 2015) or hydrology (Killiches and Czado 2015). The advantage of these models is that they are flexible and numerically tractable even in high dimensions.

Since it is interesting in many cases to determine how much two models differ, some authors like Stöber et al. (2013) and Schepsmeier (2015) use the *Kullback–Leibler (KL) distance* (Kullback and Leibler 1951) as a model distance between vines. A symmetrized version of the KL distance is given by the *Jeffreys distance (JD)* (Jeffreys 1946). However, all popular distance measures require multivariate integration, which is why they can only deal with up to three- or four-dimensional models in a reasonable amount of time.

In this paper, we will address the question of how to measure the distance between two vine copulas even for high dimensions. For this purpose, we develop methods based on the Kullback–Leibler distance, where we use the fact that it can be expressed as the sum over expectations of KL distances between univariate conditional densities. By cleverly approximating these expectations in different ways, we introduce three new distance measures with varying focuses. The *approximate Kullback–Leibler distance* (aKL)

✉ Matthias Killiches
matthias.killiches@tum.de

[1] Zentrum Mathematik, Technische Universität München, Boltzmannstraße 3, 85748 Garching, Germany

aims to approximate the true Kullback–Leibler distance via structured Monte Carlo integration and is a computationally tractable distance measure in up to five dimensions. The *diagonal Kullback–Leibler distance* (dKL) focuses on the distance between two vine copulas on specific conditioning vectors, namely those lying on certain diagonals in the space. We show that even though the resulting distance measure does not approximate the KL distance in a classical sense, it still reproduces its qualitative behavior quite well. While this way of measuring distances between vines is fast in up to ten dimensions, we still have to reduce the number of evaluation points in order to get a numerically tractable distance measure for dimensions 30 and higher. By concentrating on only one specific diagonal we achieve this, defining the *single diagonal Kullback–Leibler distance* (sdKL). The lack of symmetry of the KL distance and its substitutes is overcome by developing similar approximations to the Jeffreys distance. In numerous examples and applications, we illustrate that the proposed methods are valid distance measures and outperform benchmark approaches like Monte Carlo integration regarding computational time. Finally, in order to enable the assessment of the size of our developed distance measures we provide a baseline calibration based on the comparison of specific Gaussian copulas to the independence copula.

The paper is organized as follows: Sect. 2 introduces vine copulas and basic properties. In Sect. 3, we develop the above-mentioned model distances for vines and compare their performances in various settings. Section 4 concludes with a summary and an outlook to ongoing research.

2 Vine copulas

A *copula* $C\colon [0,1]^d \to [0,1]$ is a d-dimensional distribution function on $[0,1]^d$ with uniformly distributed margins. Since the publication of Sklar (1959), copulas have gained more and more interest and have been a frequent subject in many areas of probabilistic and statistical research. Sklar's Theorem states that for every joint distribution function $F\colon \mathbb{R}^d \to [0,1]$ of a d-dimensional random variable $(X_1,\dots,X_d)'$ with univariate marginal distribution functions F_j, $j=1,\dots,d$, there exists a copula C such that

$$F(x_1,\dots,x_d) = C\left(F_1(x_1),\dots,F_d(x_d)\right). \tag{2.1}$$

This copula C is unique if all X_j are continuous random variables. Further, if the so-called *copula density*

$$c(u_1,\dots,u_d) := \frac{\partial^d}{\partial u_1 \dots \partial u_d} C(u_1,\dots,u_d)$$

exists, one has

$$f(x_1,\dots,x_d) = c\left(F_1(x_1),\dots,F_d(x_d)\right) f_1(x_1)\dots f_d(x_d),$$

where f_j are the marginal densities. In the following, we will always assume absolute continuity of C and the existence of c. Equation (2.1) can also be used to define a multivariate distribution by combining a copula C and marginal distribution functions F_j. Thus, marginals and dependence structure can be modeled separately, as we can specify the copula C independently of the marginal distributions. A thorough overview over copulas can be found in Joe (1997) and Nelsen (2006).

There are several multivariate parametric copula families, for example, Gaussian, t, Gumbel, Clayton and Joe copulas. Being specified by a small number of parameters (usually 1 or 2), these models are rather inflexible in high dimensions. Therefore, Bedford and Cooke (2002) suggested a method for constructing copula densities based on the combination of bivariate building blocks: *vines*. The concept of vine copulas, also referred to as *pair-copula constructions* (PCCs), was used by Aas et al. (2009) to develop statistical inference methods.

As an example, a three-dimensional copula density c of a random vector $(U_1,U_2,U_3)'$ with $U_j \sim \text{uniform}(0,1)$ can be decomposed by conditioning on $U_2 = u_2$ and using $c_j(u_j) = 1$:

$$\begin{aligned} &c(u_1,u_2,u_3) \\ &\quad = c_{1,3|2}(u_1,u_3|u_2)\, c_2(u_2) \\ &\quad \overset{Sklar}{=} c_{1,3;2}\left(C_{1|2}(u_1|u_2),\right. \\ &\qquad \left. C_{3|2}(u_3|u_2);\, u_2\right) c_{1|2}(u_1|u_2)\, c_{2|3}(u_2|u_3) \\ &\quad = c_{1,3;2}\left(C_{1|2}(u_1|u_2),\right. \\ &\qquad \left. C_{3|2}(u_3|u_2);\, u_2\right) c_{1,2}(u_1,u_2)\, c_{2,3}(u_2,u_3), \end{aligned} \tag{2.2}$$

where $c_{1,3|2}(\cdot,\cdot|u_2)$ denotes the density of the conditional distribution of $(U_1,U_3)|U_2 = u_2$, while $c_{1,3;2}(\cdot,\cdot;u_2)$ is the associated copula density. The distribution function of the conditional distribution of U_j given $U_2 = u_2$ is denoted by $C_{j|2}(\cdot|u_2)$, $j=1,3$. Hence, we have expressed the three-dimensional copula density as the product over three bivariate pair-copulas.

Of course, there are alternative decompositions since the choice of U_2 as conditioning variable was arbitrary. For example, we also could have conditioned on U_1 or U_3 such that

$$\begin{aligned} c(u_1,u_2,u_3) &= c_{2,3;1}\left(C_{2|1}(u_2|u_1),\ C_{3|1}(u_3|u_1);\, u_1\right) \\ &\quad \times c_{1,2}(u_1,u_2)\, c_{1,3}(u_1,u_3), \\ c(u_1,u_2,u_3) &= c_{1,2;3}\left(C_{1|3}(u_1|u_3),\ C_{2|3}(u_2|u_3);\, u_3\right) \\ &\quad \times c_{1,3}(u_1,u_3)\, c_{2,3}(u_2,u_3). \end{aligned}$$

This way of decomposing copula densities into bivariate building blocks can be extended to arbitrary dimensions. Morales-Nápoles (2011) show that in d dimensions there are $\frac{d!}{2}\cdot 2^{\binom{d-2}{2}}$ possible vine decompositions. This flexibility and

variety of choice can be of great advantage when it comes to modeling.

Dißmann et al. (2013) and Stöber and Czado (2012) provide a method of how to store the structure of a vine copula decomposition in a lower-triangular matrix $M = (m_{i,j})_{i,j=1}^{d}$ with $m_{i,j} = 0$ for $i < j$, a so-called *vine structure matrix*.

Definition 1 (*Vine structure matrix*) A lower-triangular matrix $M = (m_{i,j})_{i,j=1}^{d}$ is called a *vine structure matrix* if it has the following three properties:

1. The entries of a selected column appear in every column to the left of that column, i.e., $\{m_{j,j}, \dots, m_{d,j}\} \subseteq \{m_{i,i}, \dots, m_{d,i}\}$ for $1 \leq i < j \leq d$.
2. The diagonal entry of a column does not appear in any column further to the right, i.e., $m_{i,i} \notin \{m_{i+1,i+1}, \dots, m_{d,i+1}\}$ for $i = 1, \dots, d-1$.
3. For $i = 1, \dots, d-2$ and $k = i+1, \dots, d$ there exists a $j > i$ such that the set $\{m_{k,i}, \{m_{k+1,i}, \dots, m_{d,i}\}\}$ is equal to

$$\{m_{j,j}, \{m_{k+1,j}, m_{k+2,j}, \dots, m_{d,j}\}\} \text{ or}$$
$$\{m_{k+1,j}, \{m_{j,j}, m_{k+2,j}, \dots, m_{d,j}\}\}.$$

The structure of the vine is encoded in the matrix as subsequently described: A pair-copula is determined by the two conditioned variables and a (possibly empty) set of conditioning variables (e.g., $c_{1,3;2}$ has conditioned variables U_1 and U_3 and conditioning variable U_2). For each entry in the structure matrix, the entry $m_{i,j}$ itself and the diagonal entry $m_{j,j}$ in the corresponding column form the indices of the two conditioned variables, while the indices of the conditioning variables are given by the entries $m_{i+1,j}, \dots, m_{d,j}$ in the corresponding column below the considered entry. The bivariate pair-copulas are evaluated at the conditional distribution functions of the distributions of each of the conditioned variables given the conditioning variables.

Expressed in formulas this means: In d dimensions, for $i > j$ the entry $m_{i,j}$ together with $m_{j,j}$ and $m_{i+1}, \dots, m_{d,j}$ stands for the copula density of the (conditional) distribution of $U_{m_{i,j}}$ and $U_{m_{j,j}}$ given $(U_{m_{i+1,j}}, \dots, U_{m_{d,j}})' = (u_{m_{i+1,j}}, \dots, u_{m_{d,j}})'$ evaluated at $C_{m_{i,j}|m_{i+1,j},\dots,m_{d,j}}(u_{m_{i,j}}| u_{m_{i+1,j}}, \dots, u_{m_{d,j}})$ and $C_{m_{j,j}|m_{i+1,j},\dots,m_{d,j}}(u_{m_{j,j}}|u_{m_{i+1,j}}, \dots, u_{m_{d,j}})$, i.e.,

$$c_{m_{i,j},m_{j,j};m_{i+1,j},\dots,m_{d,j}} \Big(C_{m_{i,j}|m_{i+1,j},\dots,m_{d,j}}(u_{m_{i,j}}|u_{m_{i+1,j}}, \dots, u_{m_{d,j}}), C_{m_{j,j}|m_{i+1,j},\dots,m_{d,j}}(u_{m_{j,j}}|u_{m_{i+1,j}}, \dots, u_{m_{d,j}}); u_{m_{i+1,j}}, \dots, u_{m_{d,j}}\Big).$$

Taking the product over all $d(d-1)/2$ pair-copula expressions implied by the vine structure matrix yields the copula density c (see Dißmann et al. 2013):

$$\begin{aligned} &c(u_1, \dots, u_d) \\ &= \prod_{j=1}^{d-1} \prod_{k=j+1}^{d} c_{m_{k,j},m_{j,j};m_{k+1,j},\dots,m_{d,j}} \\ &\quad \Big(C_{m_{k,j}|m_{k+1,j},\dots,m_{d,j}}(u_{m_{k,j}}|u_{m_{k+1,j}}, \dots, u_{m_{d,j}}), \\ &\quad C_{m_{j,j}|m_{k+1,j},\dots,m_{d,j}}(u_{m_{j,j}}|u_{m_{k+1,j}}, \dots, u_{m_{d,j}}); \\ &\quad u_{m_{k+1,j}}, \dots, u_{m_{d,j}}\Big). \end{aligned} \tag{2.3}$$

In our three-dimensional example (Eq. (2.2)), the structure matrix looks as follows:

$$M = \begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,1} & m_{3,2} & m_{3,3} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 2 & 0 \\ 2 & 3 & 3 \end{pmatrix}.$$

The entries $m_{3,1} = 2$ (together with $m_{1,1} = 1$) and $m_{3,2} = 3$ (together with $m_{2,2} = 2$) in the last row represent $c_{1,2}(u_1, u_2)$ and $c_{2,3}(u_2, u_3)$, respectively. In both cases, the conditioning set is empty because the considered entries are the last ones in their columns. The entry $m_{2,1}$ (together with $m_{1,1}$ and $m_{3,1}$) encodes the expression $c_{1,3;2}\left(C_{1|2}(u_1|u_2), C_{3|2}(u_3|u_2); u_2\right)$ since the indices of the conditioned variables are given by $m_{2,1} = 3$ and $m_{1,1} = 1$ and the conditioning variable is $m_{3,1} = 2$. Multiplying these three factors leads to the expression from Eq. (2.2). Note that there is not a unique way of encoding a given vine decomposition into a structure matrix. For instance, exchanging $m_{2,2}$ and $m_{3,2}$ in the above example yields the same vine decomposition.

Property 2 from Definition 1 implies that the diagonal of any vine structure matrix is a permutation of $1\!:\!d$, where we use the notation $r\!:\!s$ to describe the vector $(r, r+1, \dots, s)'$ for $r \leq s$. In order to simplify notation, for the remainder of the paper we assume that the diagonal of a d-dimensional structure matrix is $1\!:\!d$. This assumption comes without any loss of generality since relabeling of the variables suffices to obtain the desired property.

The following Proposition 1 states that for a vine copula with structure matrix M the (univariate) conditional density $c_{j|(j+1):d}$ of $U_j \mid (U_{j+1}, \dots, U_d)' = (u_{j+1}, \dots, u_d)'$ can be calculated by taking the product over all pair-copula expressions corresponding to the entries in the jth column of M. A proof is found in "Appendix 1".

Proposition 1 *Let* $\mathbf{U} = (U_1, \dots, U_d)'$ *be a random vector with vine copula density* c *and corresponding structure matrix* $M = (m_{i,j})_{i,j=1}^{d}$. *Then, for* $j < d$

$$
\begin{aligned}
&c_{j|(j+1):d}(u_j|u_{j+1},\dots,u_d)\\
&\quad=\prod_{k=j+1}^{d} c_{m_{k,j},m_{j,j};m_{k+1,j},\dots,m_{d,j}}\\
&\qquad\Big(C_{m_{k,j}|m_{k+1,j},\dots,m_{d,j}}\big(u_{m_{k,j}}|u_{m_{k+1,j}},\dots,u_{m_{d,j}}\big),\\
&\qquad C_{m_{j,j}|m_{k+1,j},\dots,m_{d,j}}\big(u_{m_{j,j}}|u_{m_{k+1,j}},\dots,u_{m_{d,j}}\big);\\
&\qquad u_{m_{k+1,j}},\dots,u_{m_{d,j}}\Big).
\end{aligned}
\tag{2.4}
$$

This proposition will prove itself to be crucial for the development of the distance measures from Sect. 3. For simulation and Monte Carlo integration, it is important that we can sample from vine copula distributions. Stöber and Czado (2012) and Joe (2014) provide sampling algorithms for arbitrary vine copulas. They are based on the inverse Rosenblatt transformation (Rosenblatt 1952): First, sample $w_j \sim \text{uniform}(0,1)$ for $j = 1,\dots,d$. Then, apply an inverse Rosenblatt transform T_c to the uniform sample, i.e., $\mathbf{u} = (u_1,\dots,u_d)' = T_c(\mathbf{w})$, where $\mathbf{w} = (w_1,\dots,w_d)'$ is mapped from the (uniform) w-scale to the (warped) u-scale in the following way:

- $u_d := w_d$,
- $u_{d-1} := C^{-1}_{d-1|d}(w_{d-1}|u_d)$,

  $\vdots$
- $u_1 := C^{-1}_{1|2:d}(w_1|u_2,\dots,u_d)$.

Note that the appearing inverse conditional distribution functions can be obtained easily for vine copulas. When it comes to modeling, for tractability reasons most authors assume that for pair-copulas with a non-empty conditioning set the copula itself does not depend on the conditioning variables (e.g., $c_{1,3;2}(\cdot,\cdot;u_2) = c_{1,3;2}(\cdot,\cdot)$ for any $u_2 \in [0,1]$). This assumption is referred to as the *simplifying assumption*. Among others, Haff et al. (2010), Acar et al. (2012), Stöber et al. (2013) and Killiches et al. (2016) discuss when this assumption is justified. Since *simplified vines*, i.e., vine copulas satisfying the simplifying assumption, are in practice the most relevant class of vine copulas for high dimensions, all examples in this paper consider simplified vines. Nevertheless, the presented concepts are also applicable to non-simplified vines (see Sect. 4).

We typically work in a (simplified) parametric framework, where we specify each pair-copula of the vine decomposition as a parametric bivariate copula with up to two parameters. For the sake of notation, we borrow the concept of the vine structure matrix to introduce a lower-triangular family matrix $B = \big(b_{i,j}\big)_{i,j=1}^{d}$ and two lower-triangular parameter matrices $P^{(k)} = (p^{(k)}_{i,j})_{i,j=1}^{d}$, $k = 1,2$, containing the pair-copula families and associated parameters of $c_{m_{i,j},m_{j,j}|m_{i+1,j},\dots,m_{d,j}}$, respectively. Since we only use one- and two-parametric copula families, two parameter matrices are sufficient. The entries of the family and parameter matrices, $b_{i,j}$, $p^{(1)}_{i,j}$ and $p^{(2)}_{i,j}$, specify the pair-copula corresponding to the entry $m_{i,j}$. For one-parametric families, we set the corresponding entry in the second parameter matrix to zero. For the family matrix, we use the following copula families (with corresponding abbreviations): Gaussian ($\mathscr{N}$), Student t (t), Clayton ($\mathscr{C}$), Gumbel ($\mathscr{G}$), Frank ($\mathscr{F}$) and Joe ($\mathscr{J}$). In order to compare the strengths of dependence of different copula families, we also compute the Kendall's τ values $k_{i,j}$ corresponding to pair-copulas with family $b_{i,j}$ and parameters $p^{(1)}_{i,j}$ and $p^{(2)}_{i,j}$ and store them in a lower-triangular matrix $K = \big(k_{i,j}\big)_{i,j=1}^{d}$. A (simplified) vine copula can then be written as the quadruple $\mathscr{R} = \big(M, B, P^{(1)}, P^{(2)}\big)$.

Dißmann et al. (2013) developed a sequential estimation method that fits a simplified vine, i.e., the structure matrix as well the corresponding family and parameter matrices, to a given data set. This algorithm is also implemented in R (R Core Team 2017) as the function `RVineStructureSelect` in the package `VineCopula` (Schepsmeier et al. 2017), which we use frequently throughout this paper.

Finally, for a simplified vine we define the associated *matched Gaussian vine*, i.e., the vine with the same structure matrix and Kendall's τ values but only Gaussian pair-copulas.

Definition 2 (*Matched Gaussian vine*) For a simplified vine copula $\mathscr{R} = (M, B, P^{(1)}, P^{(2)})$ let $K = \big(k_{i,j}\big)_{i,j=1}^{d}$ denote the lower-triangular matrix containing the corresponding Kendall's τ values. Then, the *matched Gaussian vine* of $\mathscr{R}$ is given by $\tilde{\mathscr{R}} = (M, \tilde{B}, \tilde{P}^{(1)}, \tilde{P}^{(2)})$, where $\tilde{B}$ is a family matrix where all entries are Gaussian, $\tilde{P}^{(1)} = (\tilde{p}^{(1)}_{i,j})_{i,j=1}^{d}$ with $\tilde{p}^{(1)}_{i,j} = \sin\big(\frac{\pi}{2}k_{i,j}\big)$ and $\tilde{P}^{(2)}$ is a zero-matrix.

3 Model distances for vines

There are many motivations to measure the model distance between different vines. For example, Stöber et al. (2013) try to find the simplified vine with the smallest distance to a given non-simplified vine. Further, it might be of interest to measure the distance between a vine copula and a Gaussian copula, both fitted to the same data set, in order to assess the need for the more complicated model. Common methods to measure such distance are the Kullback–Leibler distance and the Jeffreys distance.

3.1 Kullback–Leibler distance

Kullback and Leibler (1951) introduced a measure that indicates the distance between two d-dimensional statistical models with densities $f, g\colon \mathbb{R}^d \to [0,\infty)$. The so-called

Kullback–Leibler distance between f and g is defined as

$$\mathrm{KL}(f, g) := \int_{\mathbf{x}\in\mathbb{R}^d} \ln\left(\frac{f(\mathbf{x})}{g(\mathbf{x})}\right) f(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{3.1}$$

The KL distance between f and g can also be expressed as an expectation with respect to f:

$$\mathrm{KL}(f, g) = \mathbb{E}_f\left[\ln\left(\frac{f(\mathbf{X})}{g(\mathbf{X})}\right)\right], \tag{3.2}$$

where $\mathbf{X} \sim f$. Note that the KL distance is non-negative and equal to zero if and only if $f = g$. It is not symmetric, i.e., in general $\mathrm{KL}(f, g) \neq \mathrm{KL}(g, f)$ for arbitrary densities f and g. To clarify the order of the arguments, in the following we denote f as the *reference* density. Further, since symmetry is one of the properties of a distance, the Kullback–Leibler distance is not a distance in the classical sense and thus is often referred to as *Kullback–Leibler divergence*. A symmetrized version of the KL distance is given by the *Jeffreys distance* (Jeffreys 1946), which is defined as

$$\mathrm{JD}(f, g) = \mathrm{KL}(f, g) + \mathrm{KL}(g, f). \tag{3.3}$$

Since the Jeffreys distance is just a sum of two Kullback–Leibler distances, we will in the following sections concentrate on the KL distance and apply our results to the Jeffreys distance in Sect. 3.6.

Under the assumption that f and g have identical marginals, i.e., $f_j = g_j$, $j = 1, \ldots, d$, the KL distance between f and g is equal to the KL distance between their corresponding copula densities. This is due to the fact that the KL distance is invariant under one-to-one transformations of the marginals (Cover and Thomas 2012). Hence, if we let c^f and c^g be the copula densities corresponding to f and g, respectively, and assume that f and g have the same marginal densities, we obtain

$$\mathrm{KL}(f, g) = \mathrm{KL}\left(c^f, c^g\right). \tag{3.4}$$

In this paper, we are mainly interested in comparing different models that are obtained by fitting a data set. Since we usually first estimate the margins and afterward the dependence structure, the assumption of identical margins is always fulfilled. Hence, we will in the following concentrate on calculating the Kullback–Leibler distance between copula densities.

Having a closer look at the definition of the KL distance, we see that for its calculation a d-dimensional integral has to be evaluated. In general, this cannot be done analytically and, further, is numerically infeasible in high dimensions. For example, Schepsmeier (2015) stresses the difficulty of numerical integration in dimensions 8 and higher. In this section, we propose modifications of the Kullback–Leibler distance designed to be computationally tractable and still measure model distances adequately. These modifications are all based on the following proposition that shows that the KL distance between d-dimensional copula densities c^f and c^g can be expressed as the sum over expectations of KL distances between univariate conditional densities.

Proposition 2 *For two copula densities c^f and c^g it holds:*

$$\mathrm{KL}\left(c^f, c^g\right) = \sum_{j=1}^{d} \mathbb{E}_{c^f_{(j+1):d}}\Big[\mathrm{KL}\Big(c^f_{j|(j+1):d}\left(\cdot\,|\mathbf{U}_{(j+1):d}\right), c^g_{j|(j+1):d}\left(\cdot\,|\mathbf{U}_{(j+1):d}\right)\Big)\Big], \tag{3.5}$$

where $\mathbf{U}_{(j+1):d} \sim c^f_{(j+1):d}$ and $(d+1):d := \emptyset$.

"Appendix 2" contains the proof of an even more general version of this statement for arbitrary densities.
Proposition 2 is especially useful if c^f and c^g are vine copula densities since the appearing conditional densities can easily be obtained (see Proposition 1). As an example, for four-dimensional copula densities c^f and c^g, we can write:

$$\begin{aligned}\mathrm{KL}\left(c^f, c^g\right) = {} & \mathbb{E}_{c^f_{2:4}}\left[\mathrm{KL}\left(c^f_{1|2:4}(\cdot\,|\mathbf{U}_{2:4}), c^g_{1|2:4}(\cdot\,|\mathbf{U}_{2:4})\right)\right] \\ & + \mathbb{E}_{c^f_{3,4}}\left[\mathrm{KL}\left(c^f_{2|3,4}(\cdot\,|\mathbf{U}_{3,4}), c^g_{2|3,4}(\cdot\,|\mathbf{U}_{3,4})\right)\right] \\ & + \mathbb{E}_{c^f_4}\left[\mathrm{KL}\left(c^f_{3|4}(\cdot\,|U_4), c^g_{3|4}(\cdot\,|U_4)\right)\right] \\ & + 0,\end{aligned} \tag{3.6}$$

where for instance

$$\begin{aligned}c^f_{1|2:4}(u_1|u_2, u_3, u_4) = {} & c^f_{12}(u_1, u_2) \\ & \times c^f_{1,3;2}\left(C^f_{1|2}(u_1|u_2), C^f_{3|2}(u_3|u_2); u_2\right) \\ & \times c^f_{1,4;23}\left(C^f_{1|23}(u_1|u_2, u_3), C^f_{4|23}(u_4|u_2, u_3); u_2, u_3\right).\end{aligned}$$

The zero in the last line of Eq. (3.6) results from the fact that $c^f_4(u_4) = c^g_4(u_4) = 1$ for all $u_4 \in [0, 1]$. This is generally the case for the dth summand in Eq. (3.5), which will therefore be omitted in the following. Further note that the last nonzero term of Eq. (3.6) can also be written as $\mathrm{KL}\left(c^f_{3,4}(\cdot\,,\cdot), c^g_{3,4}(\cdot\,,\cdot)\right)$.

Of course the evaluation of the KL distance with this formula still implicitly requires the calculation of a d-dimensional integral since the expectation in the first summand of Eq. (3.5) demands a $(d-1)$-dimensional integral of the KL distance between univariate densities. A commonly used method to approximate expectations is *Monte Carlo (MC) integration* (see for example Caflisch 1998): For a random vector $\mathbf{X} \in \mathbb{R}^d$ with density $f\colon \mathbb{R}^d \to [0, \infty)$ and a scalar-valued function $h\colon \mathbb{R}^d \to \mathbb{R}$, the expectation $\mathbb{E}_f[h(\mathbf{X})] = \int_{\mathbb{R}^d} h(\mathbf{x}) f(\mathbf{x})\,\mathrm{d}\mathbf{x}$ can be approximated by

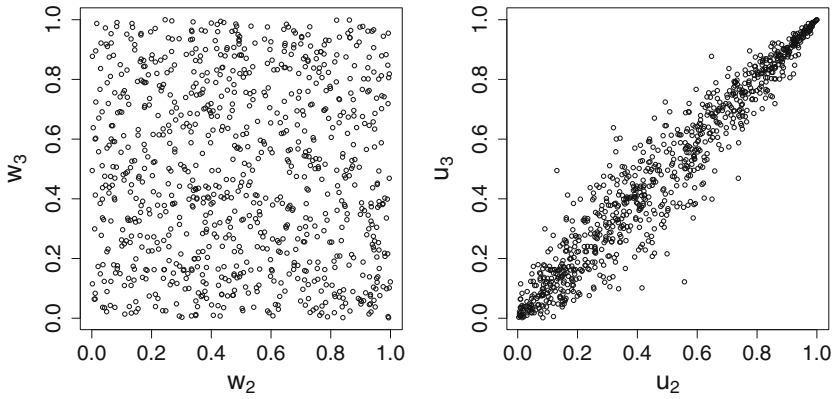

Fig. 1 Sample of size 900 from the uniform distribution (*left*) and corresponding warped sample under transformation $T_{c^f_{2,3}}$, which is a sample from a Gumbel copula with $\theta = 6$ (*right*)

$$\mathbb{E}_f[h(\mathbf{X})] \approx \frac{1}{N_{\mathrm{MC}}} \sum_{i=1}^{N_{\mathrm{MC}}} h(\mathbf{x}_i), \tag{3.7}$$

where $\{\mathbf{x}_i\}_{i=1}^{N_{\mathrm{MC}}}$ is an i.i.d. sample of size N_{MC} distributed according to the density f. However, the slow convergence rate of this method has been subject to criticism. Moreover, Do (2003) argues that when approximating the KL distance via Monte Carlo integration the random nature of the method is an unwanted property. Additionally, MC integration might produce negative approximations of KL distances even though it can be shown theoretically that the KL distance is non-negative.

As an alternative to Monte Carlo integration, in the next sections we propose several ways to approximate the expectation in Eq. (3.5) by replacing it with the average over a $(d-j)$-dimensional non-random grid $\mathscr{U}_j$, such that

$$\mathrm{KL}\left(c^f, c^g\right) \approx \sum_{j=1}^{d-1} \frac{1}{|\mathscr{U}_j|} \sum_{\mathbf{u}_{(j+1):d} \in \mathscr{U}_j} \mathrm{KL}\left(c^f_{j|(j+1):d}(\cdot\,|\mathbf{u}_{(j+1):d}), c^g_{j|(j+1):d}(\cdot\,|\mathbf{u}_{(j+1):d})\right). \tag{3.8}$$

Note that, being a sum over univariate KL distances, this approximation produces non-negative results, regardless of the grids $\mathscr{U}_j$, $j = 1, \ldots, d$. Now, the question remains how to choose the grids $\mathscr{U}_j$, such that the approximation is on the one hand fast to calculate and on the other hand still maintains the main properties of the KL distance. We will provide three possible answers to this question yielding different distance measures and investigate their performances.

Throughout the subsequent sections, we assume the following setting: Let $\mathscr{R}^f$ and $\mathscr{R}^g$ be two d-dimensional vines with copula densities c^f and c^g, respectively. We assume that their vine structure matrices have the same entries on the diagonals, i.e., $\mathrm{diag}(M^f) = \mathrm{diag}(M^g)$. Note that, although this assumption is a restriction, there are still $2^{\binom{d-2}{2}+d-2}$ different vine decompositions with equal diagonals of the structure matrix (cf. Proposition 4).[1] As before, without loss of generality we set the diagonals equal to $1:d$.

3.2 Approximate Kullback–Leibler distance

We illustrate the idea of the approximate Kullback–Leibler distance at the example of two three-dimensional vines $\mathscr{R}^f$ and $\mathscr{R}^g$. For the first summand ($j = 1$) of Eq. (3.8), the KL distance between $c^f_{1|2,3}(\cdot\,|u_2, u_3)$ and $c^g_{1|2,3}(\cdot\,|u_2, u_3)$ is calculated for all pairs $(u_2, u_3)'$ contained in the grid $\mathscr{U}_1$. In this example, we assume that the pair-copula $c^f_{2,3}$ is a Gumbel copula with parameter $\theta = 6$ (implying a Kendall's τ value of 0.83). Regarding the choice of the grid, if we used the Monte Carlo method, $\mathscr{U}_1$ would contain a random sample of $c^f_{2,3}$. Recall from Sect. 2 that such a sample can be generated by simulating from a uniform distribution on $[0, 1]^2$ and applying the inverse Rosenblatt transformation $T_{c^f_{2,3}}$. Figure 1 displays a sample of size 900 on the (uniform) w-scale and its transformation via $T_{c^f_{2,3}}$ to the (warped) u-scale.

As mentioned before we do not want our distance measure to be random. This motivates us to introduce the concept of *structured Monte Carlo integration*: Instead of sampling from the uniform distribution on the w-scale, we use a *structured grid* $\mathscr{W}$, which is an equidistant lattice on the two-dimensional unit cube,[2] and transform it to the warped u-scale by applying the inverse Rosenblatt transformation $T_{c^f_{2,3}}$. Figure 2 shows an exemplary structured grid with 30 grid points per margin.

Applying this procedure for all grids $\mathscr{U}_j$, $j = 1, \ldots, d-1$, yields the *approximate Kullback–Leibler distance*.

Definition 3 (*Approximate Kullback–Leibler distance*) Let $\mathscr{R}^f$ and $\mathscr{R}^g$ be as described above. Further, let $n \in \mathbb{N}$ be

[1] This includes, for example, C- and D-vines (Aas et al. 2009) having the same diagonal.

[2] Since most copulas have an infinite value at the boundary of the unit cube, we usually restrict ourselves to $[\varepsilon, 1-\varepsilon]^d$ for a small $\varepsilon > 0$.

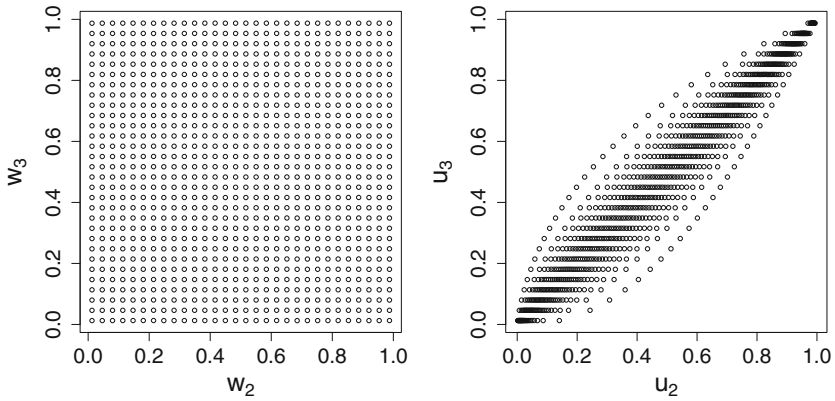

Fig. 2 Structured grid with 30 grid points per margin (*left*) and corresponding warped grid under transformation $T_{c^f_{2,3}}$ (*right*)

the number of grid points per margin and $\varepsilon > 0$. Then, the *approximate Kullback–Leibler distance* (aKL) between $\mathscr{R}^f$ (*reference* vine) and $\mathscr{R}^g$ is defined as

$$\begin{aligned}&\mathrm{aKL}\left(\mathscr{R}^f, \mathscr{R}^g\right)\\ &:= \sum_{j=1}^{d-1} \frac{1}{\left|\mathscr{G}_j\right|} \sum_{\mathbf{u}_{(j+1):d} \in \mathscr{G}_j} \mathrm{KL}\left(c^f_{j|(j+1):d}\left(\cdot\,|\mathbf{u}_{(j+1):d}\right),\right.\\ &\qquad \left. c^g_{j|(j+1):d}\left(\cdot\,|\mathbf{u}_{(j+1):d}\right)\right),\end{aligned}$$

where the warped grid $\mathscr{G}_j \subseteq [0,1]^{d-j}$ is constructed as follows:

1. Define the structured grid $\mathscr{W}_j := \{\varepsilon, \varepsilon + \Delta, \ldots, 1-\varepsilon\}^{d-j}$ to be an equidistant discretization of $[0,1]^{d-j}$ with n grid points per margin, where $\Delta := \frac{1-2\varepsilon}{n-1}$.
2. The warped grid $\mathscr{G}_j := T_{c^f_{(j+1):d}}(\mathscr{W}_j)$ is defined as the image of $\mathscr{W}_j$ under the inverse Rosenblatt transform $T_{c^f_{(j+1):d}}$ associated with the copula density $c^f_{(j+1):d}$.

Note that by construction $\left|\mathscr{G}_j\right| = n^{d-j}$.

Proposition 3 shows that the approximate KL distance in fact approximates the true KL distance in the sense that the aKL converges to the KL for $\varepsilon \to 0$ and $n \to \infty$. A proof is found in "Appendix 4".

Proposition 3 *Let $\mathscr{R}^f$ and $\mathscr{R}^g$ be as described above. Then,*

$$\lim_{\varepsilon \to 0} \lim_{n \to \infty} \mathrm{aKL}\left(\mathscr{R}^f, \mathscr{R}^g\right) = \mathrm{KL}\left(c^f, c^g\right).$$

In the following applications, we use the function `integrate` for the calculation of the one-dimensional KL. Further, we choose ε such that the convex hull of the structured grid contains volume $\beta \in (0,1)$, so $\varepsilon := \frac{1}{2}\left(1 - \beta^{\frac{1}{d-j}}\right)$. Unless otherwise specified we set β to be 95 %.

Example 1 (*Four-dimensional aKL-example*) We consider a data set from the Euro Stoxx 50, already used in Brechmann and Czado (2013). It covers a 4-year period (May 22, 2006 to April 29, 2010) containing 985 daily observations. The Euro Stoxx 50 is a major index consisting of the stocks of 50 large European companies. In this example, we consider the following four national indices: the Dutch AEX (U_1), the Italian FTSE MIB (U_2), the German DAX (U_3) and the Spanish IBEX 35 (U_4). Fitting a simplified vine to the data yields:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 4 & 2 & 0 & 0 \\ 2 & 4 & 3 & 0 \\ 3 & 3 & 4 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \mathscr{F} & 0 & 0 & 0 \\ t & t & 0 & 0 \\ t & t & t & 0 \end{pmatrix},$$

$$P^{(1)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1.01 & 0 & 0 & 0 \\ 0.36 & 0.36 & 0 & 0 \\ 0.91 & 0.89 & 0.88 & 0 \end{pmatrix}, \quad P^{(2)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 6.34 & 10.77 & 0 & 0 \\ 6.23 & 4.96 & 6.80 & 0 \end{pmatrix}.$$

As usual for financial data, most of the pair-copulas selected by the fitting algorithm are t copulas with rather high dependence; only $c_{14;23}$ is modeled as a Frank copula. Now we compute the approximate KL distance between this (reference) vine and its matched Gaussian vine (see Definition 2) and compare it to the numerically integrated KL distance. The latter limits our example to low dimensions because numerical integration becomes very slow in higher dimensions. Even in four dimensions we have to set the tolerance level of the integration routine `adaptIntegrate` of the package `cubature` from its default level of 10^{-5} to 10^{-4} to obtain results within less than 10 days. Throughout the paper, we will also consider examples for $d \geq 5$, where numerical integration becomes (almost) infeasible. As a substitute benchmark for the numerically integrated KL distance, we compare our approximated KL values to the corresponding Monte Carlo Kullback Leibler (MCKL) values, where the expectation in Eq. (3.2) is approximated by Monte Carlo integration, i.e.,

$$\mathrm{MCKL}\left(c^f, c^g\right) := \frac{1}{N_{\mathrm{MC}}} \sum_{i=1}^{N_{\mathrm{MC}}} \ln\left(\frac{c^f(\mathbf{u}^i)}{c^g(\mathbf{u}^i)}\right), \tag{3.9}$$

Table 1 Approximate, numerically integrated and Monte Carlo integrated KL distances for different parameter settings with corresponding computational times (in h)

β	aKL			Numeric	MCKL	
	$n = 10$	$n = 20$	$n = 50$	tol=10^{-4}	$N_{\text{MC}} = 10^5$	$N_{\text{MC}} = 10^6$
95 %						
Value	0.135	0.095	0.076	0.077	0.076	0.079
Time (h)	0.004	0.030	0.582	20.3	0.005	0.061
99 %						
Value	0.311	0.170	0.107	0.082	0.085	0.081
Time (h)	0.006	0.034	0.609	33.4	0.006	0.063
100 %						
Value				0.084	0.084	0.084
Time (h)				99.4	0.005	0.058

where $\mathbf{u}^1, \ldots, \mathbf{u}^{N_{\text{MC}}}$ are sampled from c^f. We choose the sample size N_{MC} to be very large in order to get acceptable low-variance results (cf. Do 2003).

Table 1 displays the approximate Kullback–Leibler distance between the fitted (reference) vine and its matched Gaussian vine for different values of β and n together with the corresponding computational time (in h).[3] We further present the numerically and Monte Carlo integrated KL distances. In order to facilitate comparability, for each value of β we compute the integrals on the corresponding domain of integration with volume β.

We see that for an increasing number of marginal grid points n, the value of the approximate KL distance gets closer to the value obtained by numerical integration. We further observe that in this example the value of the numerically integrated KL distance does not change considerably when the integral is computed on the constrained domain of integration with volume β. We expect the computational time of the aKL to increase cubically since the number of univariate KL evaluations is $\sum_{j=1}^{3} |\mathscr{G}_j| = n^3 + n^2 + n$. This is empirically validated by the observed computational times. Further, we see that even for larger values of n the aKL is still considerably faster than classical numerical integration. Concerning the Monte Carlo integrated KL distances in this example, we observe that the values still vary notably between $N_{\text{MC}} = 10^5$ and $N_{\text{MC}} = 10^6$. Thus, for the remainder of the paper we will use $N_{\text{MC}} = 10^6$ in order to get rather reliable results.

Remark 1 An anonymous referee suggested to compare our approach of the (structured) warped grid to using the Latin Hypercube sampling (LHS) method, which is a quasi-random sampler guaranteeing that the sample points are more evenly spread across the unit hypercube compared to standard Monte Carlo methods (cf. McKay et al. 1979). Section 3.6 contains a simulation study assessing the performance of the introduced distance measures. There, we also implemented LHS and compared its performance. The results showed no improvement and still had the disadvantage of being random with a rather high volatility. This is a property we wanted to avoid with our approach. The weaker performance may result from the fact that tail behavior cannot be captured sufficiently by LHS with a small sample size; for larger sample sizes, LHS loses its competitiveness due to very long computational times. For these reasons, we omit a thorough discussion of the LHS in this paper.

We can conclude that the approximate KL distance is a valid tool to estimate the Kullback–Leibler distance. However, similar to numerical integration it suffers from the curse of dimensionality, causing computational times to increase sharply when a certain precision is required or dimension increases. The number of evaluation points $|\mathscr{G}_j|$ increases exponentially in d, making calculations infeasible for higher dimensions. This motivates us to thin out the grids $\mathscr{G}_j$ in a way that considerably reduces the number of grid points, while still producing sound results. We have found that the restriction to diagonals in the unit cube fulfills these requirements reasonably well. Of course, with this modification we cannot hope for the resulting distance measure to still approximate the KL distance but we will see that in applications it reproduces the behavior of the original KL distance remarkably well.

3.3 Diagonal Kullback–Leibler distance

In order to illustrate the idea behind the *diagonal Kullback–Leibler distance*, we continue our example from Sect. 3.2. Figure 3 shows the structured and warped grids used for the aKL (gray circles). Additionally, the diagonal grid points are highlighted by filled diamonds.

The idea is now to reduce the evaluation grids $\mathscr{U}_j$ to the diagonal grids in order to define a distance measure related to the original KL distance with the advantage of reduced

[3] All numerical calculations in this paper were performed on a Linux computer (8-way Opteron) with 32 cores (each with 2.6 GHz and 3.9 GB of memory).

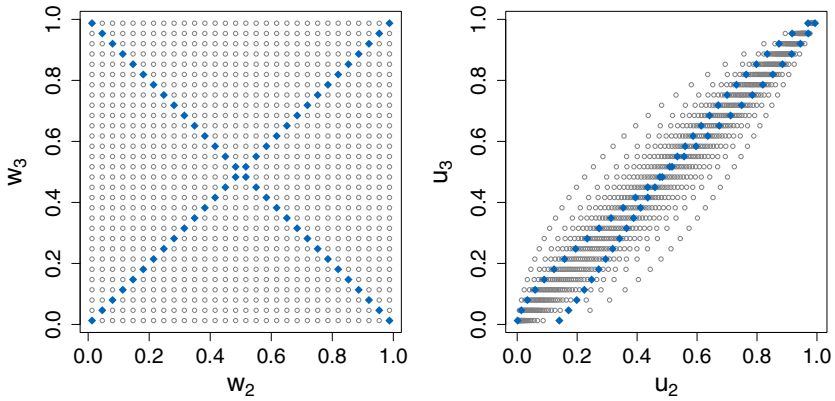

Fig. 3 Structured grid with highlighted diagonals consisting of 30 evaluation points (*left*) and corresponding warped grid under transformation $T_{c^f_{2,3}}$ (*right*)

computational costs. For this, we formally define the sets of diagonals and warped discretized diagonals.

Definition 4 (*Diagonals and warped discretized diagonals*) For $j = 1, \dots, d-1$, we define the *set of diagonals* in $[0,1]^{d-j}$:

$$D_j := \left\{ \{\mathbf{r} + t\mathbf{v}(\mathbf{r}) \mid t \in [0,1]\} \,\middle|\, \mathbf{r} \in \{0,1\}^{d-j} \right\},$$

where $\mathbf{v}(\mathbf{r}) = \big(v_1(\mathbf{r}), \dots, v_{d-j}(\mathbf{r})\big)'$ with

$$v_i(\mathbf{r}) := \begin{cases} 1 & \text{if } r_i = 0 \\ -1 & \text{if } r_i = 1 \end{cases}, \quad i = 1, \dots, d-j,$$

is the direction vector corresponding to the corner point $\mathbf{r}$. Note that the set of diagonals D_j only contains 2^{d-j-1} elements since every diagonal is implied by two corner points (e.g., the diagonals $\{(0,\dots,0)' + t(1,\dots,1)' \mid t \in [0,1]\}$ and $\{(1,\dots,1)' + t(-1,\dots,-1)' \mid t \in [0,1]\}$ coincide). Further, let $D_{j,1}, \dots, D_{j,2^{d-j-1}}$ be an arbitrary ordering of the 2^{d-j-1} diagonals. We define the kth *discretized diagonal* on the w-scale as $\mathscr{D}^w_{j,k} := D_{j,k} \cap \mathscr{W}_j$, where $\mathscr{W}_j$ is the structured grid in $[0,1]^{d-j}$ defined in Definition 3, such that it contains n grid points (cf. left panel of Fig. 3). Finally, the kth *warped discretized diagonal* on the u-scale is defined as $\mathscr{D}^u_{j,k} := T_{c^f_{(j+1):d}}\big(\mathscr{D}^w_{j,k}\big)$, where $T_{c^f_{(j+1):d}}$ is defined as in Definition 3 (cf. right panel of Fig. 3).

Now, we can define the diagonal Kullback–Leibler distance by using the set of warped discretized diagonals

$$\mathscr{D}^u_j := \bigcup_{k=1}^{2^{d-j-1}} \mathscr{D}^u_{j,k}$$

as evaluation grid $\mathscr{U}_j$.

Definition 5 (*Diagonal Kullback–Leibler distance*) Let $\mathscr{R}^f$, $\mathscr{R}^g$ and $\mathscr{D}^u_j$ be as described above. Then, the *diagonal Kullback–Leibler distance* dKL between $\mathscr{R}^f$ (*reference* vine) and $\mathscr{R}^g$ is defined as

$$\mathrm{dKL}\left(\mathscr{R}^f, \mathscr{R}^g\right) := \sum_{j=1}^{d-1} \frac{1}{\left|\mathscr{D}^u_j\right|} \sum_{\mathbf{u}_{(j+1):d} \in \mathscr{D}^u_j} \mathrm{KL}\left(c^f_{j|(j+1):d}\left(\cdot \,|\mathbf{u}_{(j+1):d}\right), c^g_{j|(j+1):d}\left(\cdot \,|\mathbf{u}_{(j+1):d}\right)\right),$$

where $\left|\mathscr{D}^u_j\right| = n \cdot 2^{d-j-1}$.

Remark 2 Similar to Proposition 3 one can show (see Appendix "Limit of the dKL") that for each of the 2^{d-j-1} diagonals $\mathscr{D}^u_{j,k}$ it holds

$$\begin{aligned} &\lim_{\varepsilon \to 0} \lim_{n \to \infty} \frac{1}{n} \sum_{\mathbf{u}_{(j+1):d} \in \mathscr{D}^u_{j,k}} \mathrm{KL}\left(c^f_{j|(j+1):d}\left(\cdot \,|\mathbf{u}_{(j+1):d}\right), \right. \\ &\qquad \left. c^g_{j|(j+1):d}\left(\cdot \,|\mathbf{u}_{(j+1):d}\right)\right) \\ &= \frac{1}{\sqrt{d-j}} \int_{\mathbf{u}_{(j+1):d} \in D^u_{j,k}} \mathrm{KL}\left(c^f_{j|(j+1):d}\left(\cdot \,|\mathbf{u}_{(j+1):d}\right), \right. \\ &\qquad \left. c^g_{j|(j+1):d}\left(\cdot \,|\mathbf{u}_{(j+1):d}\right)\right) c^f_{(j+1):d}\left(\mathbf{u}_{(j+1):d}\right) \mathrm{d}\mathbf{u}_{(j+1):d}, \end{aligned}$$

where $D^u_{j,k} := T_{c^f_{(j+1):d}}(D_{j,k})$. Hence, the diagonal Kullback–Leibler distance can be interpreted as a sum of scaled approximated line integrals over weighted univariate KL distances between conditional densities, which is exactly the integrand appearing in Proposition 2. Having the infeasibility of the aKL in higher dimensions in mind, the reduction to points on lines seems to be a good choice in order to reduce the approximation of multivariate integrals to one-dimensional ones. We choose the warped diagonals as these lines since they on the one hand contain points with high density values due to the warping and on the other hand each let all components of the conditioning vector take values on the whole range from 0 to 1. Since in practice vine models tend to differ most in the tails of the distributions, we increase the concentration of evaluation points in the tails by transforming the discretized diagonal with the method described at the end of Appendix "Tail transformation".

Table 2 dKL values and corresponding computational times (in s) for the four-dimensional Euro Stoxx 50 example

n	10	20	50	100	1000	10,000
Value	0.123	0.117	0.115	0.114	0.113	0.113
Time (s)	0.9	1.7	3.9	7.7	89	964

The following examples are supposed to illustrate that the dKL is a reasonable distance measure between vine copulas.

Applications of the dKL

Example 2 (Example 1 continued) We continue Example 1 and apply the dKL to measure the distance between the fitted four-dimensional (reference) vine and its matched Gaussian vine using different numbers of grid points n per diagonal and $\beta = 95\,\%$ as usual. The results and computational times (in s) are displayed in Table 2.

We observe that the dKL values seem to converge quite fast and are already quite close to their limit for small n. Of course, we cannot expect them to converge to the original KL distance, but we see that the order of magnitude is the same as the values in Example 1 that were calculated by numerical integration. Concerning computational times, the dKL merely needs a couple of seconds to produce reasonable results, which is a vast improvement to the computational times of the aKL and MCKL, which were in the order of hours. As expected, the computational times of the dKL grow linearly in n.

In order to assess the aptitude of the dKL for measuring the distance between vine copulas, we conduct several plausibility checks in the following examples. Since numerical integration is not practicable for these examples ($d \geq 5$), we compare our dKL values to the corresponding MCKL values.

Example 3 (*Plausibility checks*) For the first plausibility check, we consider five-dimensional t copulas with ν degrees of freedom (ranging from 3 to 30). Those can be specified as vine copulas with structure matrices $M = D_5$, family matrices B containing only t copulas, Kendall's τ matrices $K = K_5(0.5)$ and degrees of freedom matrix $P^{(2)} = V_5(\nu)$, where

$$D_d := \begin{pmatrix} 1 & & & & & \\ d & 2 & & & & \\ d-1 & d & \ddots & & & \\ \vdots & d-1 & \ddots & \ddots & & \\ 4 & \vdots & \ddots & \ddots & d-2 & \\ 3 & 4 & & \ddots & d & d-1 \\ 2 & 3 & 4 & \cdots & d-1 & d & d \end{pmatrix}, \tag{3.10}$$

$$K_d(\tau) = \left(k_{i,j}(\tau)\right)_{i,j=1}^{d}, \tag{3.11}$$

$$V_d(\nu) = \left(v_{i,j}(\nu)\right)_{i,j=1}^{d}.$$

Here, $k_{i,j}(\tau) := \left(\frac{1}{2}\right)^{d-i} \cdot \tau \cdot 1_{\{i>j\}}$ and $v_{i,j}(\nu) := (\nu + d - i) \cdot 1_{\{i>j\}}$, where $1_{\{\cdot\}}$ denotes the indicator function. Note that the structure encoded in D_d is also known as a D-vine structure (see Aas et al. 2009) and the parameter matrix $P^{(1)}$ is uniquely determined by the Kendall's τ matrix since all pairs are bivariate t copulas. Table 3 (left table) displays the diagonal ($n = 10$) and Monte Carlo ($N_{\mathrm{MC}} = 10^6$) KL distances between these (reference) t copulas and their matched Gaussian vines.

We see that the diagonal KL distance decreases as the degrees of freedom increase. This is very plausible since the t copula converges to the Gaussian copula as $\nu \to \infty$. Further, while their values are not on the same scale, we observe that the dKL and MCKL values behave similarly. This can be seen by the fact that in this example the ratio between both values ranges only between 2.20 and 2.32, where some of the fluctuation can be explained by the randomness of the MCKL. Further, the fact that the scale of the dKL differs from the one of the KL is no real drawback since the scale of the KL distance itself is not particularly meaningful. Regarding computational times, we note that in this five-dimensional example the average time for computing a MCKL value was 125 s, while the average computation of the dKL only took 3 s.

In the second plausibility check, we also deal with five-dimensional t copulas decomposed as above. However, in this scenario the degrees of freedom are fixed to be equal to 3 and the value for τ in $K_5(\tau)$ is ranging between -0.7 and 0.7. All these vines are compared to a t copula with Kendall's τ matrix $K_5(0)$ and degrees of freedom $\nu = 3$. The dKL and MCKL distances between the resulting eight (reference) t copulas and the t copula with Kendall's τ matrix $K_5(0)$ is shown in Table 3 (middle table).

Both dKL and MCKL values grow with increasing absolute value of τ as we would expect from the true KL distance. As before, the rank correlation between dKL and MCKL values is equal to 1, the ratio is nearly constant and the dKL is computed 40 times faster than the MCKL.

In the third plausibility check, we consider five-dimensional Gumbel vines (i.e., every pair-copula is a bivariate Gumbel copula having upper tail dependence) with the same structure matrix D_d and Kendall's τ matrix $K_5(0.5)$. In Table 3 (right table) we compare this reference vine to its matched Gaussian vine and other vines constructed similarly using one copula family only but retaining the same dependence in terms of the Kendall's τ matrix. As other copula families we choose the Clayton copula ($\mathscr{C}$) exhibiting lower tail dependence, the survival Clayton copula ($s\mathscr{C}$) with upper

Table 3 Left table: dKL ($n = 10$) and MCKL ($N_{\mathrm{MC}} = 10^6$) values between five-dimensional t copulas with $K = K_5(0.5)$ and $P^{(2)} = V_5(\nu)$ and their matched Gaussian vines. Middle table: dKL and MCKL values between five-dimensional t copulas with $K = K_5(\tau)$ and $P^{(2)} = V_5(3)$ and their matched Gaussian vines. Right table: dKL and MCKL values between a five-dimensional Gumbel D-vine and D-vines with the same Kendall's τ matrix constructed using one copula family only. The third column contains the ratio between dKL and MCKL

ν	dKL	MCKL	ratio
3	0.857	0.374	2.29
5	0.376	0.162	2.32
7	0.209	0.091	2.30
10	0.109	0.047	2.31
15	0.051	0.023	2.26
20	0.029	0.013	2.20
25	0.019	0.008	2.25
30	0.013	0.006	2.23

τ	dKL	MCKL	ratio
−0.7	4.702	3.226	1.46
−0.5	2.106	1.431	1.47
−0.3	0.740	0.473	1.56
−0.1	0.077	0.050	1.54
0.1	0.067	0.048	1.41
0.3	0.561	0.423	1.33
0.5	1.740	1.262	1.38
0.7	4.590	2.982	1.54

family	dKL	MCKL	ratio
$\mathscr{N}$	0.369	0.205	1.80
$\mathscr{C}$	2.987	1.780	1.68
$s\mathscr{C}$	0.322	0.158	2.04
$\mathscr{J}$	0.483	0.249	1.94

Table 4 dKL values ($n = 10$) and MCKL values ($N_{\mathrm{MC}} = 10^6$) with corresponding computational times (in s) for vines with different dimensions based on the stock exchange data. The fifth row contains the ratio between dKL and MCKL

d	3	4	5	6	7	8	9	10	11	12
dKL	0.076	0.109	0.178	0.249	0.297	0.459	0.529	0.657	0.670	0.839
Time (s)	0.4	1	3	8	22	61	145	342	788	1846
MCKL	0.048	0.075	0.098	0.140	0.172	0.211	0.240	0.287	0.322	0.354
Time (s)	25	43	87	129	158	174	235	279	408	448
Ratio	1.57	1.45	1.81	1.78	1.73	2.17	2.2	2.29	2.08	2.37

tail dependence and the Joe copula ($\mathscr{J}$) having lower tail dependence. As the difference between upper and lower tail dependent pair-copulas is large, we expect the highest distance value for the Clayton vine. Conversely, the distance to the survival Clayton vine should be the lowest. The diagonal KL distance also passes this plausibility check assigning the largest distance to the Clayton vine, a small distance to the survival Clayton vine and medium distances to the Joe and Gaussian vines. Again, the ratio between dKL and MCKL values varies only little and the dKL is still roughly 40 times faster than the MCKL regarding computational time.

The previous plausibility checks have shown that the diagonal Kullback–Leibler distance is a reasonable and fast distance measure for five-dimensional vines. Since the main motivation of the reduction to diagonals was reduced computational complexity, we now turn to higher dimensional examples.

Example 4 (*Performance in different dimensions*) In order to assess the performance of the diagonal KL distance regarding computational time in different dimensions, we again make use of the Euro Stoxx 50 data. We take the 12 German stocks (with ticker symbols ALV, BAS, BAYN, DAI, DB1, DBK, DTE, EOAN, MUV2, RWE, SIE, SAP, corresponding to $U_1, \ldots, U_{12}$) and fit vines to the first d variables ($d = 3, \ldots, 12$). We display the dKL distance ($n = 10$) between these (reference) vines and their matched Gaussian vines with corresponding computational times (in s) in Table 4. Again, we also present the approximated KL values using Monte Carlo integration ($N_{\mathrm{MC}} = 10^6$) and the ratio between dKL and MCKL values.

While we observe that the dKL is exceptionally fast in low dimensions, we note that computational times more than double when moving up one dimension. This is reasonable since the total number of diagonals in all evaluation grids is equal to

$$\sum_{j=1}^{d-1} |\mathscr{D}_j^u| = \sum_{j=1}^{d-1} 2^{d-j-1} = 2^{d-1} - 1 \tag{3.12}$$

and thus grows exponentially in d. Further, the evaluations of the conditional copula densities become more costly in higher dimensions, which can also be seen by the fact that the computational times for the MCKL increase even though the number of evaluations N_{MC} stays constant. Comparing the computational times of the dKL and MCKL one notices that the dKL is considerably faster than the MCKL in lower dimensions. Only in dimensions 10 and higher, it looses this competitive advantage. Considering the ratio between dKL and MCKL values it seems that the dKL values increase slightly faster with the dimension d than the MCKL values.

The preceding examples suggest that with the dKL distance we have found a valid distance measure between vines with reasonable computational times for up to ten dimensions. However, we are still interested in finding a distance measure computable in dimensions of order 30–50. To achieve this, the number of grid points should not depend on the dimension of the evaluation grid, implying a constant number of grid points. Hence, we choose only one of the 2^{d-j-1}

Table 5 Left table: sdKL ($n = 10$) and MCKL ($N_{\text{MC}} = 10^6$) values between five-dimensional t copulas with $P^{(2)} = V_5(\nu)$ and their matched Gaussian vines. Middle table: sdKL and MCKL values between five-dimensional t copulas with $K = K_5(\tau)$ and their matched Gaussian vines. Right table: sdKL and MCKL values between a five-dimensional Gumbel vine and vines constructed using one copula family only. The third column contains the ratio between sdKL and MCKL

ν	sdKL	MCKL	ratio
3	0.754	0.374	2.02
5	0.330	0.162	2.04
7	0.184	0.091	2.03
10	0.097	0.047	2.06
15	0.046	0.023	2.04
20	0.026	0.013	1.98
25	0.017	0.008	2.04
30	0.012	0.006	2.03

τ	sdKL	MCKL	ratio
−0.7	7.534	3.226	2.34
−0.5	3.100	1.431	2.17
−0.3	0.773	0.473	1.63
−0.1	0.053	0.050	1.05
0.1	0.157	0.048	3.30
0.3	1.322	0.423	3.12
0.5	3.226	1.262	2.56
0.7	6.193	2.982	2.08

family	sdKL	MCKL	ratio
$\mathscr{N}$	0.394	0.205	1.92
$\mathscr{C}$	3.557	1.780	2.00
$\mathscr{G}$	0.421	0.158	2.66
$\mathscr{J}$	0.576	0.249	2.32

warped discrete diagonals in $\mathscr{D}^u_j$ to be the evaluation grid. While this may seem like a very severe restriction (with the curse of dimensionality in mind), two heuristic observations justify this approach. On the one hand we observe that most of the 2^{d-j-1} diagonals contain many grid points with density values close to zero while there is always one diagonal whose points have very large density values. On the other hand we will see that the properties of the distance measure using only this single diagonal for the evaluation grid still pass the plausibility checks with values behaving closely to those of the dKL and the MCKL.

3.4 Single diagonal Kullback–Leibler distance

In order to find the one diagonal whose grid points have the highest density values, we introduce a *weighting measure* that assigns a positive real number to a diagonal depending on how the density behaves on it. The higher the density values are the more weight the corresponding diagonal obtains.

Definition 6 (*Diagonal weighting measure*) Assume we can parameterize a diagonal $D \subseteq [0,1]^d$ (on the u-scale) by the mapping $\boldsymbol{\gamma}\colon [0,1] \to [0,1]^d$. Let $c\colon [0,1]^d \to [0,\infty)$ be a copula density. Then, we define

$$\lambda_c(D) := \int_{\boldsymbol{\xi}\in D} c(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} = \int_{t\in[0,1]} c(\boldsymbol{\gamma}(t))\,\|\dot{\boldsymbol{\gamma}}(t)\|\,\mathrm{d}t \quad (3.13)$$

to be the *weight of D under c*, where $\dot{\boldsymbol{\gamma}}$ is the vector of componentwise derivatives of $\boldsymbol{\gamma}$.

We now define the *single diagonal Kullback–Leibler distance*, which is a version of the diagonal Kullback–Leibler distance that only evaluates the diagonal with the highest weight.

Definition 7 (*Single diagonal Kullback–Leibler distance*) Let $\mathscr{R}^f$, $\mathscr{R}^g$ be as before and $\mathscr{D}^u_{j,k^*_j}$ with $|\mathscr{D}^u_{j,k^*_j}| = n$ and $k^*_j := \operatorname{argmax}_k \lambda_{c^f_{(j+1):d}}\left(D^u_{j,k}\right)$ be a discretization of the corresponding diagonal with the highest weight according to $\lambda_{c^f_{(j+1):d}}$, $j = 1,\dots,d-1$. Then, the *single diagonal Kullback–Leibler distance* (sdKL) between $\mathscr{R}^f$ (*reference* vine) and $\mathscr{R}^g$ is defined as

$$\mathrm{sdKL}\left(\mathscr{R}^f, \mathscr{R}^g\right) := \sum_{j=1}^{d-1} \frac{1}{\left|\mathscr{D}^u_{j,k^*_j}\right|} \sum_{\mathbf{u}_{(j+1):d}\in\mathscr{D}^u_{j,k^*_j}} \mathrm{KL}\left(c^f_{j|(j+1):d}\left(\cdot\,|\mathbf{u}_{(j+1):d}\right), c^g_{j|(j+1):d}\left(\cdot\,|\mathbf{u}_{(j+1):d}\right)\right).$$

Remark 3 From Remark 2, we know that the single diagonal Kullback–Leibler distance approximates a scaled line integral over weighted univariate KL distances between conditional densities along the diagonal with the highest weight.

To find this diagonal, we actually would have to calculate the integral of $c^f_{(j+1):d}$ over each of the 2^{d-j-1} diagonals. In practice, this may be infeasible for high dimensions. Therefore, we propose a more sophisticated method to find a candidate for the diagonal with the highest weight. Similar to the hill-climbing algorithm used to find optimal graph structures in Bayesian networks (see Tsamardinos et al. 2006), we choose a starting value in form of a certain diagonal implied by the vine's unconditional dependencies and look in the "neighborhood" of this diagonal for another diagonal with higher weight. This procedure is repeated until a (local) maximum is found. The two procedures of finding a suitable starting diagonal and locally searching for better candidates are described in "Appendix 6".

In the following, we continue the plausibility checks from Example 3 in order to demonstrate that the restriction to a single diagonal is still enough for the resulting distance measure to generate reasonable results.

Example 5 (Plausibility checks (Example 3)) Table 5 repeats the three plausibility checks of Example 3. The resulting sdKL values obviously also pass these tests resembling the behavior of the MCKL values quite closely with relatively steady sdKL/MCKL ratios. Evaluating at only one diagonal in each grid reduces computational times even more such that the sdKL is roughly 180 times faster than the MCKL.

These five-dimensional plausibility checks empirically show that the reduction from all to one diagonal still yields viable results for our modified version of the KL distance. As a final application, we want to compare all distance measures introduced in this paper.

3.5 Comparison of all introduced KL approximations

In the following example, we will investigate the behavior of the KL, aKL, dKL, sdKL and MCKL in dimensions $d = 3, 4, 5, 7, 10, 15, 20, 30$. We make use of the fact that the Kullback–Leibler distance between Gaussian copulas can be expressed analytically (Hershey and Olsen 2007). For two Gaussian copulas c^f and c^g with correlation matrices Σ^f and Σ^g, respectively, one has

$$\mathrm{KL}\left(c^f, c^g\right) = \frac{1}{2}\left\{\ln\left(\frac{\det\left(\Sigma^g\right)}{\det\left(\Sigma^f\right)}\right) + \mathrm{tr}\left(\left(\Sigma^g\right)^{-1}\Sigma^f\right) - d\right\},$$

where $\det(\cdot)$ denotes the determinant and $\mathrm{tr}(\cdot)$ the trace of a matrix. For each dimension d we use a reference Gaussian vine $\mathscr{R}^0$ (which is also a Gaussian copula) with the (D-vine) structure matrix $M = D_d$ and Kendall's τ matrix $K = K_d(0.5)$ (cf. Eqs. (3.10) and (3.11), respectively).
We generate another $m = 50$ Gaussian vines $\mathscr{R}^r$, $r = 1, \ldots, m$, with the same structure matrix $M = D_d$ and a parameter matrix $P^{(1)}$, where the $d(d-1)/2$ partial correlations are simulated such that the corresponding correlation matrix is uniform over the space of valid correlation matrices. For this purpose, we follow Joe (2006): For $i = 2, \ldots, d$ and $j = 1, \ldots, i-1$ we draw $p_{i,j}^{(1)}$ from a Beta$(i/2, i/2)$ distribution and transform it linearly to $[-1, 1]$.
We compare the reference vine $\mathscr{R}^0$ to each $\mathscr{R}^r$ using the model distances KL, MCKL, aKL, dKL and sdKL. Since the Kullback–Leibler distance is exact in these cases, we can assess the performance of the remaining distance measures by comparing their $m = 50$ distance values to the ones of the true KL. As the scale of the KL and related distance measures cannot be interpreted in a sensible way, we are only interested in how well the ordering suggested by the KL is reproduced by aKL ($n = 20$), dKL ($n = 10$), sdKL ($n = 10$) and MCKL ($N_{\mathrm{MC}} = 10^6$), respectively. Hence, we consider the respective rank correlations to the KL values in order to assess their performances. The results and average computation times are displayed in Table 6 and 7, respectively.

Table 6 Rank correlations (in %) between the true Kullback–Leibler distance and the results of aKL, dKL, sdKL and MCKL, respectively, for $d = 3, 4, 5, 7, 10, 15, 20, 30$

d	3	4	5	7	10	15	20	30
aKL	98.4	98.5	98.4	–	–	–	–	–
dKL	96.7	97.4	98.7	98.7	97.2	–	–	–
sdKL	93.3	90.2	91.5	89.7	82.9	84.8	84.5	80.4
MCKL	99.8	99.5	99.5	99.7	98.5	97.2	92.3	91.7

With a rank correlation of more than 98 %, the approximate KL performs extremely well for $d = 3, 4, 5$. However, computational times increase drastically with the dimension such that it cannot be computed in higher dimensions in a reasonable amount of time. As the plausibility checks from the previous sections suggested, the diagonal KL also produces very good results. In lower dimensions, the dKL is competitive regarding computational times. Only for dimensions 10 and higher it becomes slower due to the exponentially increasing number of diagonals. Therefore, calculations have not been performed for $d = 15, 20, 30$. As expected, restricting to only one diagonal reduces computational times considerably such that even in very high dimensions they are kept to a minimum. Of course, this restriction comes along with slight loss of performance, still achieving a rank correlation of over 80 % in 30 dimensions. Being a consistent estimator of the KL distance (for $N_{\mathrm{MC}} \to \infty$), the Monte Carlo KL has the best performance of the considered model distances. However, the performance decreases for high dimensions due to the curse of dimensionality ($N_{\mathrm{MC}} = 10^6$ for all d). Further, the price of the slightly better performance (compared to sdKL) is a considerably higher computational time, e.g., in 10 and 30 dimensions the sdKL is roughly 20 and 9 times faster than the MCKL, respectively.

Altogether, we can say that in order to have good performance and low computational times one should use the dKL in lower dimensions and then switch to the sdKL in higher dimensions in order to obtain a usable proxy for the KL distance at (relatively) low computational costs.

Table 7 Average computational times (in s) of aKL, dKL, sdKL and MCKL for $d = 3, 4, 5, 7, 10, 15, 20, 30$

d	3	4	5	7	10	15	20	30
aKL	3.46	117.67	4357.91	–	–	–	–	–
dKL	0.23	0.82	2.49	19.63	338.32	–	–	–
sdKL	0.18	0.38	0.69	1.80	4.86	16.04	36.91	114.12
MCKL	7.35	14.50	24.12	46.05	97.54	239.17	473.41	961.12

Table 8 Rank correlations (in %) between the true Jeffreys distance and the results of aJD, dJD, sdJD and MCJD, respectively, for $d = 3, 4, 5, 7, 10, 15, 20, 30$

d	3	4	5	7	10	15	20	30
aJD	97.2	96.1	97.7	–	–	–	–	–
dJD	95.6	95.1	98.0	98.0	96.2	–	–	–
sdJD	86.9	86.1	85.5	85.5	82.7	83.5	84.5	83.8
MCJD	99.7	100	99.8	99.8	99.7	98.9	95.6	92.7

3.6 Comparison of the resulting JD approximations

All approximations of the KL distance discussed in this paper can be easily used to define corresponding approximations of the Jeffreys distance [see Eq. (3.3)]. We will call these aJD, dJD, sdJD and MCJD, where, for example, $\text{sdJD}(f, g) = \text{sdKL}(f, g) + \text{sdKL}(g, f)$. We repeat the simulation study from Sect. 3.5 for the comparison of the approximated JD values to the true one. Table 8 displays the results.

We see that the results are similar to the ones where we just considered the KL distance. Of course, being sums of two approximated KL distances the approximations of the Jeffreys distance are more volatile and therefore perform slightly worse than their KL counterparts. However, we still have aJD and dJD values close to 100 % and sdJD values around 85 %. As one would have expected, computational times of the JD substitutes are simply (approximately) twice as long as the ones of their KL counterparts (cf. Table 7). So we can conclude that in low dimensions suitable substitutes for the Jeffreys Distance are given by the aJD and dJD with better computational times than those of the MCJD. In dimensions 10 and higher, the sdJD would be the measure of choice with low computational times and high correlations to the true Jeffreys distance. In practice, users can decide whether they want to apply substitutes of the Jeffreys distance or the Kullback–Leibler distance, depending on whether the focus is on symmetry of the distance measure or computational times.

3.7 Calibration

The results in Sect. 3.5 showed that the sdKL is a valid substitute for the Kullback–Leibler distance since it ranks the differences between the considered models very similarly. However, what still remains is the drawback that any divergence or distance measure shares: One distance value alone cannot be interpreted properly. Therefore, we will provide a baseline comparison in this section. Based on these results, one can assess whether a certain sdKL-value is small or large. Of course, this procedure can similarly be used to calibrate any of the other distance measures. As reference vines we take exchangeable d-dimensional Gaussian copulas with correlation matrix $\Sigma(\rho) = \big(\sigma_{i,j}(\rho)\big)_{i,j=1,\dots,d}$ with $\sigma_{i,j}(\rho) = 1$ for $i = j$ and $\sigma_{i,j}(\rho) = \rho$ for $i \neq j$, i.e., every pair of variables has the same correlation coefficient ρ [all copulas are written as D-vines with structure matrix D_d, cf. Eq. (3.10)]. These (reference) copulas are compared to the d-dimensional independence copula using the sdKL. Figure 4 shows the sdKL values for $d = 4$ and $d = 20$ for values of ρ between 0 and 0.90.

Of course, both graphs start at 0 as for $\rho = 0$ the exchangeable Gaussian copula is simply the independence copula such that the compared models are the same. As one would expect, the distance increases as ρ increases, regardless of the dimension. However, we see that the scale for $d = 4$ is very different from that of $d = 20$. Whereas a sdKL-value of 1 corresponds to a ρ of roughly 0.4 in 4 dimensions, for $d = 20$ it corresponds to a ρ of approximately 0.02. Plots like the ones in Fig. 4 can now be used as a baseline comparison: If we obtain an sdKL-value of 1 between two four-dimensional vine copulas, we know that this comparable to how much an exchangeable Gaussian copula with $\rho = 0.4$ differs from the independence copula, which is in fact considerable. If we get the same sdKL-value for $d = 20$, this corresponds to the difference between an exchangeable Gaussian copula with $\rho = 0.02$ and the independence copula, which is not too extreme.

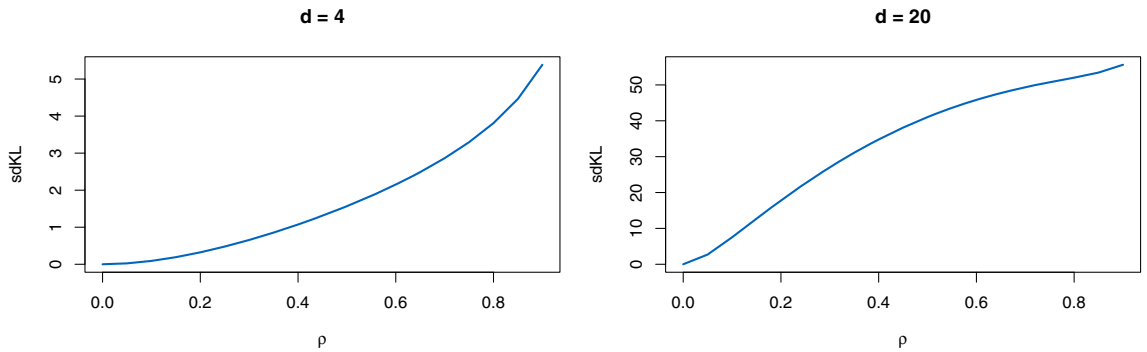

Fig. 4 Plots of the sdKL between the exchangeable (reference) Gaussian copula with joint correlation ρ and the independence copula against ρ for $d = 4$ (*left*) and $d = 20$ (*right*)

Of course this calibration procedure can be easily extended to a calibration of approximated Jeffreys distances. To interpret a given sdJD value, one can either use the corresponding baseline comparison of the sdKL multiplied by 2 (since the sdJD is the sum of two sdKL values) or consider a similar plot of the sdJD between exchangeable Gaussian copulas and the independence copula to classify the sdJD value as small or large.

4 Conclusion

In this paper, we have developed new methods for measuring model distances between vine copulas. Since vines are frequently used for high-dimensional dependence modeling, the focus was to propose concepts that can in particular be applied to higher dimensional models. With the approximate Kullback–Leibler distance, we introduced a measure which converges to the original Kullback–Leibler distance and therefore produces good approximations. Although being considerably faster than the calculation of the KL by numerical integration, the aKL suffers from the curse of dimensionality and therefore is not computationally tractable in dimensions $d \geq 6$. Being a more crude approximation, the diagonal Kullback–Leibler distance, which highlights the difference between vines conditioned on points on the diagonals, has proven itself to be a reliable and computationally parsimonious model distance measure for comparing vines in up to 10 dimensions. In higher dimensions, the number of diagonals becomes intractable, which is why we suggested to reduce calculations to only one diagonal with large density values, introducing the single diagonal Kullback–Leibler distance. With the sdKL, we have found a possibility to overcome the shortfalls of alternative methods like Monte Carlo (low speed and randomness) and at the same time maintain the desired properties of the Kullback–Leibler distance relatively well. For the sake of interpretability, we provided a baseline calibration answering the question whether a distance value is small or large. The above distance measures can be used to substitute the Jeffreys distance. In a simulation study, we have seen that the performance is very similar, whereas computational times double.

In ongoing research, we address ourselves to applying our distance measures in many possible fields: applications to real data sets, comparing C- and D-vines, determining appropriate truncation levels for vines and comparing non-simplified vines with simplified ones in order to develop a test to decide whether the simplifying assumption is satisfied for given data.

Acknowledgements The authors would like to thank the editor and an anonymous referee for their constructive comments and suggestions, which helped to improve the quality of the paper. The first author acknowledges financial support by a research stipend of the Technische Universität München. The third author is supported by the German Research Foundation (DFG Grant CZ 86/4-1). Numerical calculations were performed on a Linux cluster supported by DFG Grant INST 95/919-1 FUGG.

Appendix 1: Proof of Proposition 1

From Eq. (2.3) we know that the vine copula density can be written as a product over the pair-copula expressions corresponding to the matrix entries. In Property 2.8 (ii), Dißmann et al. (2013) state that deleting the first row and column from a d-dimensional structure matrix yields a $(d-1)$-dimensional trimmed structure matrix. Due to Property 2 from Definition 1 the entry $m_{1,1} = 1$ does not appear in the remaining matrix. Hence, we obtain the density $c_{2:d}$ by taking the product over all pair-copula expressions corresponding to the entries in the trimmed matrix. Iterating this argument yields that the entries of matrix $M_k := (m_{i,j})_{i,j=k+1}^{d}$ resulting from cutting the first k rows and columns from M represent the density $c_{(k+1):d}$. In general, we have

$$c_{j|(j+1):d}\left(u_j|u_{j+1},\ldots,u_d\right) = \frac{c_{j:d}\left(u_j,\ldots,u_d\right)}{c_{(j+1):d}\left(u_{j+1},\ldots,u_d\right)}.$$

The numerator and denominator can be obtained as the product over all pair-copula expressions corresponding to the entries of M_{j-1} and M_j. Thus, $c_{j|(j+1):d}$ is simply the product over the expressions corresponding to the entries from the first column of M_{j-1}. This proves Eq. (2.4).

Appendix 2: Proof of Proposition 2

We will prove an even more general version of Proposition 2 that holds for arbitrary densities f and g:

$$\mathrm{KL}\left(f,g\right) = \sum_{j=1}^{d} \mathbb{E}_{f_{(j+1):d}}\Big[\mathrm{KL}\left(f_{j|(j+1):d}\left(\cdot|\mathbf{X}_{(j+1):d}\right), g_{j|(j+1):d}\left(\cdot|\mathbf{X}_{(j+1):d}\right)\right)\Big],$$

where $\mathbf{X}_{(j+1):d} \sim f_{(j+1):d}$ and $(d+1):d := \emptyset$. Proposition 2 then follows directly from this statement.

Recall that using recursive conditioning we can obtain for density f

$$f\left(x_1,\ldots,x_d\right) = \prod_{j=1}^{d} f_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right).$$

Thus, the Kullback–Leibler distance between f and g can be written in the following way:

$$
\begin{aligned}
\mathrm{KL}\left(f,g\right) &= \int\limits_{\mathbf{x}\in\mathbb{R}^d} \ln\left(\frac{f(\mathbf{x})}{g(\mathbf{x})}\right) f(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \int\limits_{\mathbf{x}\in\mathbb{R}^d} \sum_{j=1}^{d} \ln\left(\frac{f_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right)}{g_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right)}\right) f(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \sum_{j=1}^{d} \int\limits_{x_d\in\mathbb{R}} \cdots \int\limits_{x_1\in\mathbb{R}} \ln\left(\frac{f_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right)}{g_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right)}\right) \\
&\quad \times f(x_1,\ldots,x_d)\,\mathrm{d}x_1\cdots\,\mathrm{d}x_d \\
&= \sum_{j=1}^{d} \int\limits_{x_d\in\mathbb{R}} \cdots \int\limits_{x_j\in\mathbb{R}} \ln\left(\frac{f_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right)}{g_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right)}\right) \\
&\quad \times \left\{ \int\limits_{x_{j-1}\in\mathbb{R}} \cdots \int\limits_{x_1\in\mathbb{R}} f(x_1,\ldots,x_d)\,\mathrm{d}x_1\cdots\mathrm{d}x_{j-1} \right\} \mathrm{d}x_j\cdots\,\mathrm{d}x_d \\
&= \sum_{j=1}^{d} \int\limits_{x_d\in\mathbb{R}} \cdots \int\limits_{x_j\in\mathbb{R}} \ln\left(\frac{f_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right)}{g_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right)}\right) \\
&\quad \times f_{j,\ldots,d}(x_j,\ldots,x_d)\,\mathrm{d}x_j\cdots\,\mathrm{d}x_d \\
&= \sum_{j=1}^{d} \int\limits_{x_d\in\mathbb{R}} \cdots \int\limits_{x_{j+1}\in\mathbb{R}} \left\{ \int\limits_{x_j\in\mathbb{R}} \ln\left(\frac{f_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right)}{g_{j|(j+1):d}\left(x_j|\mathbf{x}_{(j+1):d}\right)}\right) \right. \\
&\quad \left. \times f_{j|(j+1):d}(x_j|\mathbf{x}_{(j+1):d})\,\mathrm{d}x_j \right\} f_{(j+1):d}(\mathbf{x}_{(j+1):d})\,\mathrm{d}x_{j+1}\cdots\mathrm{d}x_d \\
&= \sum_{j=1}^{d} \mathbb{E}_{f_{(j+1):d}} \\
&\quad \left[\mathrm{KL}\left(f_{j|(j+1):d}(\cdot|\mathbf{X}_{(j+1):d}), g_{j|(j+1):d}(\cdot|\mathbf{X}_{(j+1):d})\right)\right].
\end{aligned}
$$

Appendix 3: Number of vines with the same diagonal

Proposition 4 *Let $\boldsymbol{\sigma} = (\sigma_1,\ldots,\sigma_d)'$ be a permutation of $1{:}d$. Then, there exist $2^{\binom{d-2}{2}+d-2}$ different vine decompositions whose structure matrix has the diagonal $\boldsymbol{\sigma}$.*

Proof The number of vine decompositions whose structure matrix has the same diagonal σ can be calculated as the quotient of the number of valid structure matrices and the number of possible diagonals. Morales-Nápoles (2011) show that there are $\frac{d!}{2}\cdot 2^{\binom{d-2}{d}}$ different vine decompositions. In each of the $d-1$ steps of the algorithm for encoding a vine decomposition in a structure matrix (see Stöber and Czado 2012) we have two possible choices such that there are 2^{d-1} structure matrices representing the same vine decomposition. Hence, there are in total $\frac{d!}{2}\cdot 2^{\binom{d-2}{d}}\cdot 2^{d-1}$ valid structure matrices. Further, there are $d!$ different diagonals. Thus, for a fixed diagonal $\boldsymbol{\sigma}$ there exist

$$
\frac{\frac{d!}{2}\cdot 2^{\binom{d-2}{d}}\cdot 2^{d-1}}{d!} = 2^{\binom{d-2}{2}+d-2} \quad \text{different vine decompositions.}
$$

□

Appendix 4: Proof of Proposition 3

Let $\varepsilon > 0$ and $n \in \mathbb{N}$. To simplify notation, for $j = 1,\ldots,d-1$ we define

$$
\begin{aligned}
&\kappa_j\left(\mathbf{u}_{(j+1):d}\right) \\
&\quad := \mathrm{KL}\left(c^f_{j|(j+1):d}\left(\cdot|\mathbf{u}_{(j+1):d}\right), c^g_{j|(j+1):d}\left(\cdot|\mathbf{u}_{(j+1):d}\right)\right).
\end{aligned}
$$

Then, by definition

$$
\begin{aligned}
\mathrm{aKL}\left(\mathscr{R}^f,\mathscr{R}^g\right) &= \sum_{j=1}^{d-1} \frac{1}{n^{d-j}} \sum_{\mathbf{u}_{(j+1):d}\in\mathscr{G}_j} \kappa_j\left(\mathbf{u}_{(j+1):d}\right) \\
&= \sum_{j=1}^{d-1} \frac{1}{n^{d-j}} \sum_{\mathbf{w}_{(j+1):d}\in\mathscr{W}_j} \kappa_j\left(T_{c^f_{(j+1):d}}(\mathbf{w}_{(j+1):d})\right).
\end{aligned}
$$

Since W_j is a discretization of $[\varepsilon,1-\varepsilon]^{d-j}$ with mesh going to zero for $n\to\infty$, we have

$$
\begin{aligned}
&\frac{1}{n^{d-j}} \sum_{\mathbf{w}_{(j+1):d}\in\mathscr{W}_j} \kappa_j\left(T_{c^f_{(j+1):d}}(\mathbf{w}_{(j+1):d})\right) \\
&\xrightarrow{n\to\infty} \int_{[\varepsilon,1-\varepsilon]^{d-j}} \kappa_j\left(T_{c^f_{(j+1):d}}(\mathbf{w}_{(j+1):d})\right)\mathrm{d}\mathbf{w}_{(j+1):d}.
\end{aligned}
$$

Substituting $\mathbf{w}_{(j+1):d} = T^{-1}_{c^f_{(j+1):d}}\left(\mathbf{u}_{(j+1):d}\right)$ yields

$$
\begin{aligned}
&\int_{[\varepsilon,1-\varepsilon]^{d-j}} \kappa_j\left(T_{c^f_{(j+1):d}}(\mathbf{w}_{(j+1):d})\right)\mathrm{d}\mathbf{w}_{(j+1):d} \\
&\quad = \int_{T_{c^f_{(j+1):d}}([\varepsilon,1-\varepsilon]^{d-j})} \kappa_j\left(\mathbf{u}_{(j+1):d}\right) c^f_{(j+1):d} \\
&\qquad \left(\mathbf{u}_{(j+1):d}\right)\mathrm{d}\mathbf{u}_{(j+1):d}
\end{aligned}
$$

since

$$
\begin{aligned}
&T^{-1}_{c^f_{(j+1):d}}\left(\mathbf{u}_{(j+1):d}\right) \\
&\quad = \left(C^f_{j+1|(j+2):d}(u_{j+1}|\mathbf{u}_{(j+2):d}),\ldots,C^f_{d-1|d}(u_{d-1}|\mathbf{u}_d),u_d\right)'
\end{aligned}
$$

with (upper triangular) Jacobian matrix

$$
\begin{aligned}
J &= J_{T^{-1}_{c^f_{(j+1):d}}}\left(\mathbf{u}_{(j+1):d}\right) \\
&= \begin{pmatrix} c^f_{j+1|(j+2):d}(u_{j+1}|\mathbf{u}_{(j+2):d}) & & \\ & \ddots & * \\ 0 & & c^f_{d-1|d}(u_{d-1}|\mathbf{u}_d) \\ & & 1 \end{pmatrix}
\end{aligned}
$$

such that $\mathrm{d}\mathbf{w}_{(j+1):d} = \det(J)\,\mathrm{d}\mathbf{u}_{(j+1):d} = c^f_{(j+1):d}\left(\mathbf{u}_{(j+1):d}\right)\mathrm{d}\mathbf{u}_{(j+1):d}$. Since we are only interested in the determinant of J, whose lower-triangular matrix contains only zeros, the values in the upper triangular matrix (denoted by $*$) are irrelevant here. Finally, using the fact that

$$\begin{aligned}\lim_{\varepsilon\to 0} T_{c^f_{(j+1):d}}\left([\varepsilon, 1-\varepsilon]^{d-j}\right) &= T_{c^f_{(j+1):d}}\left([0,1]^{d-j}\right)\\ &= [0,1]^{d-j},\end{aligned}$$

we obtain

$$\begin{aligned}&\lim_{\varepsilon\to 0}\lim_{n\to\infty} \mathrm{aKL}\left(\mathscr{R}^f, \mathscr{R}^g\right)\\ &= \sum_{j=1}^{d-1} \int_{[0,1]^{d-j}} \kappa_j\left(\mathbf{u}_{(j+1):d}\right) c^f_{(j+1):d}\left(\mathbf{u}_{(j+1):d}\right)\mathrm{d}\mathbf{u}_{(j+1):d}\\ &\overset{\text{Prop. 2}}{=} \mathrm{KL}\left(c^f, c^g\right).\end{aligned}$$

Appendix 5: Regarding Remark 2

Limit of the dKL

Let $\varepsilon > 0$ and $n \in \mathbb{N}$. Again, for $j = 1, \ldots, d-1$ we define

$$\begin{aligned}&\kappa_j\left(\mathbf{u}_{(j+1):d}\right)\\ &:= \mathrm{KL}\left(c^f_{j|(j+1):d}\left(\cdot\,|\mathbf{u}_{(j+1):d}\right), c^g_{j|(j+1):d}\left(\cdot\,|\mathbf{u}_{(j+1):d}\right)\right).\end{aligned}$$

The contribution of $\mathscr{D}^u_{j,k}$, $j = 1, \ldots, d-1, k=1, \ldots, 2^{d-j-1}$, to the dKL is given by

$$\begin{aligned}&\frac{1}{n}\sum_{\mathbf{u}_{(j+1):d}\in\mathscr{D}^u_{j,k}} \kappa_j\left(\mathbf{u}_{(j+1):d}\right)\\ &= \frac{1}{n}\sum_{\mathbf{w}_{(j+1):d}\in\mathscr{D}^w_{j,k}} \kappa_j\left(T_{c^f_{(j+1):d}}(\mathbf{w}_{(j+1):d})\right)\\ &= \frac{1}{n}\sum_{i=1}^{n} \kappa_j\left(T_{c^f_{(j+1):d}}(\boldsymbol{\omega}(t_i))\right),\end{aligned}$$

where $\boldsymbol{\omega}(t) = \mathbf{r} + t\mathbf{v}(\mathbf{r})$ with $\mathbf{v}(\cdot)$ as defined in Definition 4, $\mathbf{r} \in \{0,1\}^{d-j}$ being a corner point of $D^w_{j,k}$ and $t_i = \varepsilon + (i-1)\frac{1-2\varepsilon}{n-1}$ for $i = 1, \ldots, n$. Letting $n \to \infty$ yields

$$\frac{1}{n}\sum_{i=1}^{n} \kappa_j\left(T_{c^f_{(j+1):d}}(\boldsymbol{\omega}(t_i))\right) \overset{n\to\infty}{\longrightarrow} \int_{t\in[\varepsilon,1-\varepsilon]} \kappa_j\left(T_{c^f_{(j+1):d}}(\boldsymbol{\omega}(t))\right)\mathrm{d}t. \tag{4.1}$$

Now, we further let $\varepsilon \to 0$ and use the fact that $\|\dot{\boldsymbol{\omega}}(t)\| = \sqrt{d-j}$ to obtain

$$\begin{aligned}&\int_{t\in[0,1]} \kappa_j\left(T_{c^f_{(j+1):d}}(\boldsymbol{\omega}(t))\right)\mathrm{d}t\\ &= \frac{1}{\sqrt{d-j}}\int_{t\in[0,1]} \kappa_j\left(T_{c^f_{(j+1):d}}(\boldsymbol{\omega}(t))\right)\|\dot{\boldsymbol{\omega}}(t)\|\,\mathrm{d}t\\ &= \frac{1}{\sqrt{d-j}}\int_{\mathbf{w}_{(j+1):d}\in D^w_{j,k}} \kappa_j\left(T_{c^f_{(j+1):d}}(\mathbf{w}_{(j+1):d})\right)\mathrm{d}\mathbf{w}_{(j+1):d}\\ &= \frac{1}{\sqrt{d-j}}\int_{\mathbf{u}_{(j+1):d}\in D^u_{j,k}} \kappa_j\left(\mathbf{u}_{(j+1):d}\right) c_{(j+1):d}\\ &\quad\left(\mathbf{u}_{(j+1):d}\right)\mathrm{d}\mathbf{u}_{(j+1):d},\end{aligned}$$

where we used the substitution $\mathbf{u}_{(j+1):d} := T^{-1}_{c^f_{(j+1):d}}(\mathbf{w}_{(j+1):d})$, $\mathrm{d}\mathbf{w}_{(j+1):d} = c^f_{(j+1):d}(\mathbf{u}_{(j+1):d})\,\mathrm{d}\mathbf{u}_{(j+1):d}$ (cf. Appendix D) in the last line.

Tail transformation

In our empirical applications of the dKL, we have noticed that different vines tend to differ most in the tails of the distribution. Therefore, we increase the concentration of evaluation points in the tails of the diagonal by transforming the points t_i, $i = 1, \ldots, n$, via a suited function Ψ. Hence, by substituting $t = \Psi(s)$ in Eq. (4.1) we obtain

$$\int_{s\in\Psi^{-1}([\varepsilon,1-\varepsilon])} \kappa_j\left(T_{c^f_{(j+1):d}}\left(\eta\left(\Psi(s)\right)\right)\right)\Psi'(s)\,\mathrm{d}s.$$

We use its discrete pendant

$$\frac{1}{n}\sum_{i=1}^{n} \kappa_j\left(T_{c^f_{(j+1):d}}\left(\eta\left(\Psi(s_i)\right)\right)\right)\Psi'(s_i),$$

where $s_i = \Psi^{-1}(\varepsilon) + (i-1)\frac{\Psi^{-1}(1-\varepsilon)-\Psi^{-1}(\varepsilon)}{n-1}$ for $i = 1, \ldots, n$. Regarding the choice of Ψ, all results in this paper are obtained using

$$\Psi_a\colon [0,1] \to [0,1], \quad \Psi_a(t) := \frac{\Phi(2a(t-0.5)) - \Phi(-a)}{2\Phi(a) - 1}$$

with shape parameter $a > 0$, where Φ is the standard normal distribution function. Figure 5 shows the graph of Ψ_a for different values of a. We see that larger values of a imply more points being transformed into the tails. Having tested different values for a, we found that $a = 4$ yields the best overall results. Therefore, we consistently use $a = 4$.

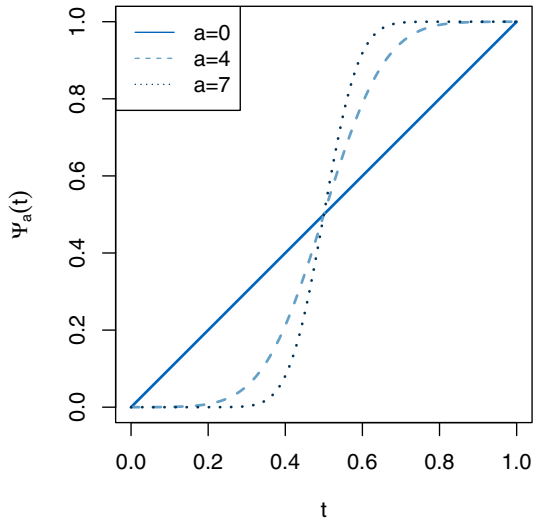

Fig. 5 Plot of Ψ_a for $a = 0, 4, 7$

Appendix 6: Finding the diagonal with the highest weight

Procedure 1: Finding a starting value

The idea behind the following heuristic is that a diagonal has a higher weight if its points have high probability implied by the copula density. Hence, the diagonal should reflect the dependence structure of the variables. The unconditional dependence in a vine captures most of the total dependence and is easy to interpret. For example, if U_i and U_j are positively dependent (i.e., $\tau_{i,j} > 0$) and U_j and U_k are negatively dependent (i.e., $\tau_{j,k} < 0$), then it seems plausible that U_i and U_k are negatively dependent. This concept can be extended to arbitrary dimensions.

1. Take each variable to be a node in an empty graph.
2. Consider the last row of the structure matrix, encoding the unconditional pair-copulas. Connect two nodes by an edge if the dependence of the corresponding variables is described by one of those copulas.
3. Assign a "+" to node 1.
4. As long as not all nodes have been assigned a sign, repeat:
   (a) For each node that has been assigned a sign in the previous step, consider its neighborhood.
   (b) If the root node has a "+," then assign to the neighbor node the sign of the Kendall's τ of the pair-copula connecting the root and neighbor node, else the opposite sign.
5. The resulting direction vector $\mathbf{v} \in \{-1, 1\}^d$ has entries v_i which are 1 or -1 if node i is has been assigned a "+" or a "−," respectively.

Table 9 Specification of the pair-copulas with empty conditioning set

pair-copula	$c_{1,2}$	$c_{1,3}$	$c_{3,4}$	$c_{3,5}$	$c_{2,6}$	$c_{6,7}$	$c_{7,8}$	$c_{7,9}$
Kendall's τ	−0.3	0.5	0.2	−0.4	0.5	0.5	−0.4	0.6

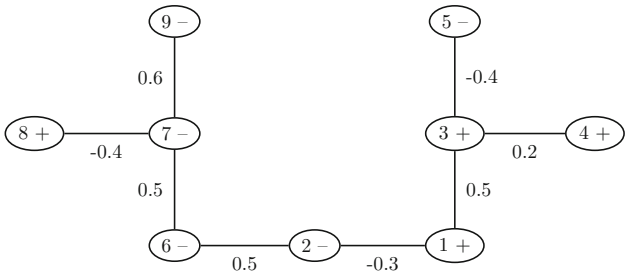

Fig. 6 Example for finding the candidate vector

Note that if we had assigned a "−" to node 1 in step 3, we would have ended up with $-v$ instead of v, implying the same diagonal.

To illustrate the procedure from above, we consider a nine-dimensional example: Let $\mathscr{R}$ be a vine copula with density c, where the following (unconditional) pair-copulas are specified:

Now, we take an empty graph with node 1 to 9 and add edges (i, j) if $c_{i,j}$ is specified in Table 9. The result is a tree on the nodes 1 to 9 (see Fig. 6). We assign a "+" to node 1 and consider its neighborhood $\{2, 3\}$ as there are still nodes without a sign. Since $\tau_{1,2} < 0$ and the root node 1 has been assigned a "+", node 2 gets a "−". Node 3 is assigned a "+". Next, we repeat this procedure for the neighborhoods of nodes 2 and 3. Iterating in this way until all nodes have been assigned a "+" or a "−" we obtain what is shown in Fig. 6. The resulting direction vector is given by $\mathbf{v} = (1, -1, 1, 1, -1, -1, -1, 1, -1)'$.

Procedure 2: Local search for better candidates

Having found a diagonal through Procedure 1 (Appendix "Procedure 1: Finding a starting value"), we additionally perform the following steps in order to look whether there is a diagonal with even higher weight in the "neighborhood" of $\mathbf{v}$.

1. Consider a candidate diagonal vector $\mathbf{v} \in \{1, -1\}^d$ with corresponding weight $\lambda_c^{(0)}$.
2. For $j = 1, \ldots, d$, calculate the weight $\lambda_c^{(j)}$ corresponding to $\mathbf{v}_j \in \{1, -1\}^d$, where $\mathbf{v}_j$ is equal to $\mathbf{v}$ with the sign of the jth entry being reversed.
3. If $\max_i \lambda_c^{(i)} > \lambda_c^{(0)}$, take $\mathbf{v} := \mathbf{v}_k$ with $k = \text{argmax}_i \lambda_c^{(i)}$ to be the new candidate for the (local) maximum.
4. Repeat the steps 1–3 until a (local) maximum is found, i.e., $\max_i \lambda_c^{(i)} \leq \lambda_c^{(0)}$.

Although there is no guarantee that we really find the global maximum of the diagonal weights, this procedure in any case finds a local maximum. Starting with a very plausible choice of $\mathbf{v}$, it is highly likely that we end up with the "right" diagonal.

In step 2, the weight of numerous diagonals has to be calculated. For a fast determination of these weights, it is reasonable to approximate the integral in Eq. (3.13) by

$$\lambda_c(D) \approx \frac{1}{n} \sum_{i=1}^{n} c(\boldsymbol{\gamma}(t_i)) \, \|\dot{\boldsymbol{\gamma}}(t_i)\| \,,$$

where $0 < t_1 < t_2 < \ldots < t_n < 1$ is an equidistant discretization of $[0, 1]$.

References

Aas, K., Czado, C., Frigessi, A., Bakken, H.: Pair-copula constructions of multiple dependence. Insur. Math. Econ. **44**, 182–198 (2009)

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

Bedford, T., Cooke, R.M.: Vines: a new graphical model for dependent random variables. Ann. Stat. **30**(4), 1031–1068 (2002)

Brechmann, E.C., Czado, C.: Risk management with high-dimensional vine copulas: an analysis of the Euro Stoxx 50. Stat. Risk Model. **30**(4), 307–342 (2013)

Caflisch, R.E.: Monte carlo and quasi-monte carlo methods. Acta Numer. **7**, 1–49 (1998)

Cooke, R.M., Joe, H., Chang, B.: Vine regression. Resources for the Future Discussion Paper, pp. 15–52 (2015)

Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, Hoboken (2012)

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

Do, M.N.: Fast approximation of Kullback–Leibler distance for dependence trees and hidden Markov models. IEEE Signal Process. Lett. **10**(4), 115–118 (2003)

Haff, I.H., Aas, K., Frigessi, A.: On the simplified pair-copula construction—simply useful or too simplistic? J. Multivar. Anal. **101**(5), 1296–1310 (2010)

Hershey, J.R., Olsen, P.A.: Approximating the Kullback Leibler divergence between Gaussian mixture models. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007, vol. 4, pp. IV–317. IEEE (2007)

Jeffreys, H.: An invariant form for the prior probability in estimation problems. In: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 186, pp. 453–461. The Royal Society (1946)

Joe, H.: Multivariate Models and Multivariate Dependence Concepts. CRC Press, Boca Raton (1997)

Joe, H.: Generating random correlation matrices based on partial correlations. J. Multivar. Anal. **97**(10), 2177–2189 (2006)

Joe, H.: Dependence Modeling with Copulas. CRC Press, Boca Raton (2014)

Killiches, M., Czado, C.: Block-maxima of vines. In: Dey, D., Yan, J. (eds.) Extreme Value Modelling and Risk Analysis: Methods and Applications, pp. 109–130. CRC Press, Boca Raton (2015)

Killiches, M., Kraus, D., Czado, C.: Examination and visualisation of the simplifying assumption for vine copulas in three dimensions. Aust. N. Z. J. Stat. (2016). doi:10.1111/anzs.12182

Kraus, D., Czado, C.: D-vine copula based quantile regression. Comput. Stat. Data Anal. **110C**, 1–18 (2017)

Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**(1), 79–86 (1951)

Maya, L., Albeiro, R., Gomez-Gonzalez, J.E., Melo Velandia, L.F.: Latin american exchange rate dependencies: a regular vine copula approach. Contemp. Econ. Policy **33**(3), 535–549 (2015)

McKay, M.D., Beckman, R.J., Conover, W.J.: Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**, 239–245 (1979)

Morales-Nápoles, O.: Counting vines. In: Kurowicka, D., Joe, H. (eds.) Dependence Modeling: Vine Copula Handbook. World Scientific Publishing Co, Singapore (2011)

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

Nelsen, R.: An Introduction to Copulas, 2nd edn. Springer-Science Business Media, New York (2006)

Panagiotelis, A., Czado, C., Joe, H.: Pair copula constructions for multivariate discrete data. J. Am. Stat. Assoc. **107**(499), 1063–1072 (2012)

R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna (2017)

Rosenblatt, M.: Remarks on a Multivariate Transformation. Ann. Math. Stat. **23**(3), 470–472 (1952)

Schepsmeier, U.: Efficient information based goodness-of-fit tests for vine copula models with fixed margins. J. Multivar. Anal. **138**, 34–52 (2015)

Schepsmeier, U., Stoeber, J., Brechmann, E.C., Graeler, B., Nagler, T., Erhardt, T.: VineCopula: statistical inference of vine copulas. R Package Version **2**(1), 1 (2017)

Sklar, A.: Fonctions dé Repartition á n Dimensions et leurs Marges. Publ. Inst. Stat. Univ. Paris **8**, 229–231 (1959)

Stöber, J., Czado, C.: Pair copula constructions. In: Mai, J.-F., Scherer, M. (eds.) Simulating Copulas: Stochastic Models, Sampling Algorithms, and Applications. World Scientific, Singapore (2012)

Stöber, J., Joe, H., Czado, C.: Simplified pair copula constructions—limitations and extensions. J. Multivar. Anal. **119**, 101–118 (2013)

Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max–min hill-climbing Bayesian network structure learning algorithm. Mach. Learn. **65**(1), 31–78 (2006)