Stat Comput (2018) 28:219–237
https://doi.org/10.1007/s11222-017-9727-9

Vine copula approximation: a generic method for coping with conditional dependence

Mimi Zhang[1] · **Tim Bedford**[1]

Received: 2 April 2016 / Accepted: 13 January 2017 / Published online: 31 January 2017

Abstract Pair-copula constructions (or vine copulas) are structured, in the layout of vines, with bivariate copulas and conditional bivariate copulas. The main contribution of the current work is an approach to the long-standing problem: how to cope with the dependence structure between the two conditioned variables indicated by an edge, acknowledging that the dependence structure changes with the values of the conditioning variables. The changeable dependence problem, though recognized as crucial in the field of multivariate modelling, remains widely unexplored due to its inherent complication and hence is the motivation of the current work. Rather than resorting to traditional parametric or nonparametric methods, we proceed from an innovative viewpoint: approximating a conditional copula, to any required degree of approximation, by utilizing a family of basis functions. We fully incorporate the impact of the conditioning variables on the functional form of a conditional copula by employing local learning methods. The attractions and dilemmas of the pair-copula approximating technique are revealed via simulated data, and its practical importance is evidenced via a real data set.

Keywords Compact set · Cross-validation · k-means clustering · Kullback–Leibler divergence · Weighted average · Locally weighted regression

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s11222-017-9727-9) contains supplementary material, which is available to authorized users.

✉ Mimi Zhang
mimi.zhang@strath.ac.uk

[1] Department of Management Science, University of Strathclyde, Glasgow G1 1XQ, UK

1 Introduction

Pair-copula constructions (or vine copulas), introduced by Joe (1996) and further developed by Bedford and Cooke (2001), Bedford and Cooke (2002) and Aas et al. (2009), provide an adaptable and manageable way of modelling the dependence structure within a random vector. While a multivariate copula is superior to a multivariate joint distribution (in that the former divides the problem of specifying a full joint distribution into two: the problem of modelling marginal distributions and the problem of modelling multivariate dependence structure), a vine copula is more preferable than a multivariate copula (in that, compared with bivariate copulas, multivariate copulas developed in the literature are quite few and are incapable of capturing all the possible dependence structures within a random vector). A vine copula owes its flexibility and competence in modelling multivariate dependence to its vine hierarchy—a graphical tool for stacking (conditional) bivariate copulas. Over the past decade, vine copulas have been used in a variety of applied work, including finance, hydrology, meteorology, biostatistics, machine learning, geology and wind energy; see, e.g., Soto et al. (2012), Fan and Patton (2014), Hao and Singh (2015) and Valizadeh et al. (2015). Pircalabelu et al. (2015) incorporated vine copulas into Bayesian network to deal with continuous variables, while Panagiotelis et al. (2012) studied the problem of applying vine copulas to discrete multivariate data. We refer the reader to Joe (2014) for a comprehensive review on vine copulas and related topics.

A vine copula is a hierarchy of bivariate copulas and conditional bivariate copulas. For a conditional bivariate copula, the dependence structure between the two conditioned variables (i.e., the functional form of the conditional copula) can be highly influenced by the conditioning variables. Though

theoretical and applied literature on vine copulas is quite large, the vast majority of the documented work adopted the simplifying assumption that the functional form of a conditional bivariate copula does not change with the conditioning variables (Acar et al. 2012). To name a few, Haff (2013) extended the work of Aas et al. (2009) to develop a stepwise semi-parametric estimator for parameter estimation of vine copulas; both Aas et al. (2009) and Haff (2013) assumed that the parameters of conditional bivariate copulas are all fixed. Later, Haff and Segers (2015) developed a method for nonparametric estimation of vine copulas. Again, they employed the simplifying assumption. Likewise, by adopting the simplifying assumption, Kauermann and Schellhase (2014) approximated conditional bivariate copulas by tensor product of a family of basis functions. So and Yeung (2014) assumed that certain dependence measures, e.g., rank correlation, change with time yet not with conditioning variables. See Stöber et al. (2013) for a discussion on limitations of simplified pair-copula constructions.

Apparently, ignoring the role of the conditioning variables in a conditional bivariate copula will contaminate the whole performance of the fitted multivariate copula. A natural practice to model a changeable conditional bivariate copula is to employ a parametric copula of which the involved parameter is a function of the conditioning variables; see, e.g., Gijbels et al. (2011), Veraverbeke et al. (2011) and Dißmann et al. (2013). Acar et al. (2011) approximated the function by local polynomials. Lopez-Paz et al. (2013) employed a type of parametric copulas that can be fully determined by Kendall's τ rank correlation coefficient; they related Kendall's τ rank correlation coefficient to conditioning variables by the standard normal distribution. We shall point out that, on the one hand, the choice of a parametric copula is always preconceived, usually from the existing parametric copulas in the literature. On the other hand, the functional form, relating the copula parameter and the conditioning variables, is always subjectively determined. The main contribution of the current work is a generic approach to the changeable dependence problem. One distinguishing feature of our approach is that we do not impose any structural assumption on the true (conditional) bivariate copula, except that (1) the copula is continuous w.r.t. its two arguments and its parameters and (2) the parameters are continuous functions of the conditioning variables. We approximate the true copula by a family of basis functions (to any required degree of approximation). The feasibility of the pair-copula approximating approach is guaranteed by the theoretical work developed in Bedford et al. (2016).

The remainder of the paper is organized as follows. In Sect. 2, we give a brief summary of vine copula and relative information. In Sect. 3, we present the general procedure for approximating bivariate copulas and conditional bivariate copulas. Section 4 is devoted to dealing with some technical issues when approximating a conditional copula. In Sect. 5, the attractions and dilemmas of the pair-copula approximating technique are revealed via simulated data, and its practical importance is evidenced via a real data set.

2 Vine copula and relative information

2.1 Vine copula

Vine is a graphical tool for helping construct multivariate distributions in a flexible and explicit manner. A vine on n variables is a nested set of connected trees: $\{T_1, \ldots, T_{n-1}\}$ in which the edges of tree T_i $(i = 1, \ldots, n-2)$ are the nodes of tree T_{i+1}, and each tree has the maximum number of edges. A *regular vine* on n variables is a particular vine in which two edges in tree T_i $(i = 1, \ldots, n-2)$ are joined by an edge in tree T_{i+1} if and only if the two edges share a common node. Formally, a vine is defined as follows (Kurowicka 2011, Chapter 3).

Definition 1 $\mathcal{V}$ is a vine on n variables with $\mathcal{E}(\mathcal{V}) = \mathcal{E}_1 \cup \cdots \cup \mathcal{E}_{n-1}$ denoting the set of edges if

1. $\mathcal{V} = \{T_1, \ldots, T_{n-1}\}$;
2. T_1 is a tree with nodes $\mathcal{N}_1 = \{1, \ldots, n\}$ and a set of $(n-1)$ edges denoted by $\mathcal{E}_1$;
3. for $i = 2, \ldots, n-1$, T_i is a tree with nodes $\mathcal{N}_i = \mathcal{E}_{i-1}$. $\mathcal{V}$ is a regular vine on n variables if, additionally,
4. $\forall\ e = \{e_1, e_2\} \in \mathcal{E}_i$ $(i = 2, \ldots, n-1)$, we have $\#\{e_1 \triangle e_2\} = 2$.

Here, $\triangle$ is the symmetric difference operator, and # is the cardinality operator. A regular vine (called D-vine) on 4 variables is exemplified in Fig. 1.

In Fig. 1, T_1 is a tree with nodes $\mathcal{N}_1 = \{1, 2, 3, 4\}$ and edges $\mathcal{E}_1 = \{\{1, 2\}, \{2, 3\}, \{3, 4\}\}$, and T_2 is a tree with nodes $\mathcal{N}_2 = \mathcal{E}_1$ and edges $\mathcal{E}_2 = \{\{1, 3|2\}, \{2, 4|3\}\}$. For an edge,

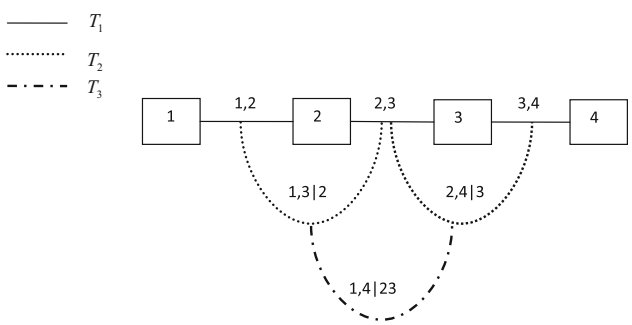

Fig. 1 A regular vine on four variables

the set of variables to the right of the vertical slash is called a conditioning set, and the set of variables to the left of the vertical slash is called a conditioned set. The constraint set, conditioning set and conditioned set of an edge are defined as follows.

Definition 2 $\forall\, e = \{e_1,\ e_2\} \in \mathcal{E}_i$ $(i = 2, \ldots, n-1)$, the *constraint set* related to edge e is the subset of $\{1, \ldots, n\}$ reachable from e. Write U_e^* for the constraint set of e. The *conditioning set* of e is $D_e = U_{e_1}^* \cap U_{e_2}^*$, and the conditioned set of e is $\{U_{e_1}^* \setminus D_e,\ U_{e_2}^* \setminus D_e\}$.

Here, $U_{e_1}^* \setminus D_e$ represents the relative complement of D_e in $U_{e_1}^*$. We might write $\dot{e}_1$ for $U_{e_1}^* \setminus D_e$ and $\dot{e}_2$ for $U_{e_2}^* \setminus D_e$. Hence the conditioned set of e is $\{\dot{e}_1, \dot{e}_2\}$. Throughout the work, we represent edge e by $\{\dot{e}_1, \dot{e}_2 | D_e\}$. Referring to Fig. 1, the set of edges for tree T_3 contains only one element: $\mathcal{E}_3 = \{\{1, 4|2, 3\}\}$; the constraint set of the edge is $\{1, 2, 3, 4\}$, the conditioned set of the edge is $\{1, 4\}$, and the conditioning set of the edge is $\{2, 3\}$. If $e = \{e_1,\ e_2\} \in \mathcal{E}_1$, we have $U_e^* = \{e_1,\ e_2\}$ and D_e is empty.

An n-variate copula is an n-variate probability distribution defined on the unit hypercube $[0, 1]^n$ with uniform marginal distributions. There is a one-to-one correspondence between the set of n-variate copulas and the set of n-variate distributions, as was stated in a theorem by Sklar (1959).

Theorem 1 *Given random variables $X_1, \ldots, X_n$ having continuous distribution functions $F_1(x_1)$, $\cdots$, $F_n(x_n)$ and a joint distribution function $F(x_1, \ldots, x_n)$, there exists a unique n-variate copula $C(\cdot)$ such that*

$$F(x_1, \ldots, x_n) = C(F_1(x_1), \ldots, F_n(x_n)), \quad \forall\, (x_1, \ldots, x_n) \in \mathbb{R}^n. \tag{1}$$

And conversely, given continuous distribution functions $F_1(x_1)$, $\cdots$, $F_n(x_n)$ and an n-variate copula $C(\cdot)$, $F(x_1, \cdots, x_n)$ defined through Eq. (1) is an n-variate distribution with marginal distribution functions $F_1(x_1)$, …, $F_n(x_n)$.

The coupling of regular vines and bivariate copulas produces a particularly versatile tool, called vine copula or pair-copula construction, for modelling multivariate data. The backbone of vine copula is re-forming, according to the structure of a regular vine, a multivariate copula into a hierarchy of (conditional) bivariate copulas. Given a regular vine $\mathcal{V}$, for any $e \in \mathcal{E}(\mathcal{V})$ with the conditioned set $\{\dot{e}_1,\ \dot{e}_2\}$ and the conditioning set D_e, let $\boldsymbol{X}_e = (X_v : v \in D_e)$ denote the vector of random variables indicated by the conditioning set D_e. Throughout the work, all vectors are defined to be row vectors. Define $C_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$ (resp. $c_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$) to be the bivariate copula (resp. copula density) for the edge e. $C_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$ and $c_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$ are conditioned on $\boldsymbol{X}_e$. Let $x_{\dot{e}_1}$, $x_{\dot{e}_2}$ and $\mathbf{x}_e$, respectively, denote, from the generic point of view, the value of $X_{\dot{e}_1}$, $X_{\dot{e}_2}$ and $\boldsymbol{X}_e$. We have the following theorem (Bedford and Cooke 2001).

Theorem 2 *Let $\mathcal{V} = \{T_1, \ldots, T_{n-1}\}$ be a regular vine on the random variables $\{X_1, \ldots, X_n\}$, and let the marginal distribution function $F_i(x_i)$ and density function $f_i(x_i)$ $(i = 1, \ldots, n)$ be given. Then the vine-dependent n-variate distribution is uniquely determined with density function*

$$f(x_1, \ldots, x_n) = \prod_{i=1}^{n} f_i(x_i) \times \prod_{e \in \mathcal{E}(\mathcal{V})} c_{\dot{e}_1\dot{e}_2|D_e}\left(u_{\mathbf{x}_e},\ w_{\mathbf{x}_e} | \boldsymbol{X}_e = \mathbf{x}_e\right).$$

Here, $u_{\mathbf{x}_e} = F_{\dot{e}_1|D_e}(x_{\dot{e}_1} | \boldsymbol{X}_e = \mathbf{x}_e)$ and $w_{\mathbf{x}_e} = F_{\dot{e}_2|D_e}(x_{\dot{e}_2} | \boldsymbol{X}_e = \mathbf{x}_e)$ are two conditional marginal distributions, both conditioned on $\boldsymbol{X}_e$. All the involved conditional marginal distributions can be derived from the marginal distribution functions and copula densities. (See Section 2.2 of the supplementary material for more discussion on deriving conditional marginal distributions.) Theorem 2 claims that we are able to derive the n-variate density function, once we are given the n marginal distribution functions and all the bivariate copulas originated from the regular vine. The n marginal distribution functions can be readily estimated from collected data, either parametrically or empirically, by using standard univariate methods. The estimation of the involved bivariate copulas is, however, non-trivial and still remains an open problem. Note that the form of the copula density $c_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$ (namely the dependence structure between $F_{\dot{e}_1|D_e}(X_{\dot{e}_1} | \boldsymbol{X}_e = \mathbf{x}_e)$ and $F_{\dot{e}_2|D_e}(X_{\dot{e}_2} | \boldsymbol{X}_e = \mathbf{x}_e)$) can be highly influenced by the value of $\boldsymbol{X}_e$. The dependence of the form of $c_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$ on $\boldsymbol{X}_e$ is always intentionally ignored in the community of vine copula, due to certain practical concerns such as computational load and the curse of dimensionality (see, e.g., Kauermann and Schellhase 2014).

In Sect. 3, we will introduce a family of minimally informative copulas that can cope with the dependence of $c_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$ on $\boldsymbol{X}_e$. Deriving a minimally informative copula involves the notion of relative information (Kullback–Leibler divergence).

2.2 Relative information

Definition 3 The relative information of Q from P is a non-symmetric measure of the information lost when a probability measure P is approximated by another probability measure Q (over a set Ω). P should be absolutely continuous w.r.t. Q. The relative information of Q from P, denoted by $I(P|Q)$, is defined by

$$I(P|Q) = \int_{\Omega} \log\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right) \mathrm{d}P.$$

Here, $\frac{\mathrm{d}P}{\mathrm{d}Q}$ is the Radon–Nikodym derivative of P w.r.t. Q. If μ is any measure on Ω for which $\frac{\mathrm{d}P}{\mathrm{d}\mu}$ and $\frac{\mathrm{d}Q}{\mathrm{d}\mu}$ exist, then the relative information of Q from P can be written into

$$I(P|Q) = \int_{\Omega} p \log\left(\frac{p}{q}\right) \mathrm{d}\mu,$$

where $p = \frac{\mathrm{d}P}{\mathrm{d}\mu}$ and $q = \frac{\mathrm{d}Q}{\mathrm{d}\mu}$.

Relative information is always nonnegative and is minimized to 0 when $P = Q$ almost everywhere.

Relative information is a popular "metric" for measuring probability distance. There are two elegant properties of relative information, making it a natural criterion for copula selection. Firstly, relative information is invariant under monotonic transformation. For example, let n-variate distributions $f(x_1, \ldots, x_n)$ and $g(x_1, \ldots, x_n)$ have identical marginal distributions: $f_i(x_i)$, $i = 1, \ldots, n$. Write $c_f(\cdot)$ for the copula density of $f(x_1, \ldots, x_n)$, and $c_g(\cdot)$ for the copula density of $g(x_1, \ldots, x_n)$. If we want to approximate

$$f(x_1, \ldots, x_n) = \prod_{i=1}^{n} f_i(x_i) c_f\left(F_1(x_1), \ldots, F_n(x_n)\right),$$

by

$$g(x_1, \ldots, x_n) = \prod_{i=1}^{n} f_i(x_i) c_g\left(F_1(x_1), \cdots, F_n(x_n)\right),$$

then we have

$$\begin{aligned} I(f|g) &= \int_{\mathbb{R}^n} f(x_1, \ldots, x_n) \log \\ &\quad \left(\frac{f(x_1, \ldots, x_n)}{g(x_1, \ldots, x_n)}\right) \mathrm{d}x_1 \cdots \mathrm{d}x_n \\ &= \int_{\mathbb{R}^n} c_f(F_1(x_1), \ldots, F_n(x_n)) \log \\ &\quad \left(\frac{c_f(F_1(x_1), \ldots, F_n(x_n))}{c_g(F_1(x_1), \ldots, F_n(x_n))}\right) \\ &\quad \mathrm{d}F_1(x_1) \cdots \mathrm{d}F_n(x_n). \end{aligned} \tag{2}$$

Therefore, if $f(x_1, \ldots, x_n)$ is the true law and $c_g(F_1(x_1), \ldots, F_n(x_n))$ has the minimum relative information w.r.t. $c_f(F_1(x_1), \ldots, F_n(x_n))$, then $g(x_1, \ldots, x_n)$ has the minimum relative information w.r.t. $f(x_1, \ldots, x_n)$. In what follows, we say an n-variate copula is minimally informative if the relative information of it from the independence copula is minimal. Therefore, a minimally informative copula is the most "independent" copula among all the qualified copulas. See (Jaynes 2003, Chapter 11) for an enlightening explanation on relative information (therein called entropy), which gives justification for the employment of minimally informative copulas for analyzing multivariate data. Given a multivariate data set, Eq. (2) reduces the problem of finding the minimally informative multivariate distribution to the problem of finding the minimally informative multivariate copula. Then, how to find the minimally informative multivariate copula? The second property of relative information claims that a vine-dependent distribution is minimally informative if and only if all its bivariate copulas are minimally informative (Bedford and Cooke 2002). Therefore, to guarantee that a multivariate copula be minimally informative, we only need to find the minimally informative bivariate copula for each edge in the regular vine.

We now frame our line of approach to modelling multivariate data as follows. *Given a multivariate data set and a regular vine on the involved random variables, we will formulate the optimal bivariate copula for every edge in the regular vine from the top level to the bottom level. A bivariate copula is optimal in the sense that it meets all the specified constraints and is minimally informative, making the corresponding multivariate copula minimally informative.* Clearly, there are two problems related to our approach, which will be attended to in the following section: (1) the type of constraints that a bivariate copula needs to meet and (2) how to analytically formulate the optimal copula.

Remark 1 Here, we presume that the structure of the regular vine is given. The determination of the structure of a regular vine is an open topic of great importance. A well-structured vine copula can capture the underlying multivariate law by copulas in the lower hierarchy of the vine and therefore can reduce computational load and mitigate the curse of dimensionality by simplifying copulas in the deeper hierarchy of the vine (Dißmann et al. 2013). This topic is beyond the scope of the current work and will be addressed in the future.

3 Minimally informative copula

In the following, for exposition convenience, we assume that the n random variables $X_1, \ldots, X_n$ are uniformly distributed. (We can readily transform an arbitrary multivariate data set into a data set with uniform marginal distributions by taking the probability integral transformation.) The constraints that a qualified copula for an edge needs to meet are called expectation constraints. We illustrate, via an example, the manner and rationale of specifying expectation constraints, while a more detailed explanation is given by Bedford et al. (2016).

Example 1 Let $\mathcal{V} = \{T_1, \ldots, T_{n-1}\}$ be a regular vine on the uniform random variables $\{X_1, \ldots, X_n\}$. For now, we concentrate on an edge $e \in \mathcal{E}_1$; that is, the conditioning set of e is empty. Let X_i and X_j $(1 \leq i < j \leq n)$ be the two uniform random variables joined by the edge e. A bivariate copula density $\ddot{c}_e(x_i, x_j)$ for the edge e is said to be qualified, if the following k (≥ 1) expectation constraints are satisfied:

$$\alpha_\ell = \int_0^1 \int_0^1 h_\ell(x_i, x_j)\ddot{c}_e(x_i, x_j)\mathrm{d}x_i\mathrm{d}x_j, \quad \ell = 1, \ldots, k. \tag{3}$$

Here, the real-valued functions $h_1(x_i, x_j), \ldots, h_k(x_i, x_j)$ are linearly independent, modulo the constants; $\{\alpha_1, \ldots, \alpha_k\}$ are known, whose values can be obtained from data or expert elicitation. Equation (3) says that a qualified copula should satisfy the constraints that the expected value of the random variable $h_\ell(X_i, X_j)$ is α_ℓ for $\ell = 1, \ldots, k$. For example, when $h_k(X_i, X_j) = X_i X_j$, then the rank correlation of a qualified copula should be α_k.

In practice, we know a priori the expected values of the random variables $\{h_1(X_i, X_j), \ldots, h_k(X_i, X_j)\}$; then, every qualified copula for edge e should satisfy the constraints given in Eq. (3), and we select from these qualified copulas the minimally informative one. Many constraints can be written in the form of expectation constraints. For example, constraints are commonly specified in the form of probabilities. Yet, a probability can be expressed as the expectation of an identity function. Another conventional way to specify constraints is in the form of various kinds of correlations, such as product-moment correlations. Yet, due to the one-to-one correspondence between the set of n-variate copulas and the set of n-variate distributions, any correlation can be expressed as an expectation w.r.t. an appropriate copula. The way of specifying expectation constraints also allows a wider range of constraints if desired.

Another major advantage of specifying expectation constraints is that the minimally informative bivariate copula for an edge, satisfying all the specified expectation constraints, can be readily determined.

Example 2 (continued) According to Nussbaum (1989) and Borwein et al. (1994) (see Bedford and Wilson (2014) for a summary), there exists uniquely a minimally informative bivariate copula satisfying all the expectation constraints in (3) with the copula density given by

$$\hat{c}_e(x_i, x_j) = d_1(x_i)d_2(x_j)\exp\left(\lambda_1 h_1(x_i, x_j) + \cdots + \lambda_k h_k(x_i, x_j)\right). \tag{4}$$

The Lagrange multipliers $\lambda_1, \ldots, \lambda_k$ are unknown and depend nonlinearly on $\alpha_1, \ldots, \alpha_k$. The functions $d_1(\cdot)$ and $d_2(\cdot)$ are two regularity functions, making $\hat{c}_e(x_i, x_j)$ a copula density. Let $A(x_i, x_j)$ denote the exponential part:

$$A(x_i, x_j) = \exp\left(\lambda_1 h_1(x_i, x_j) + \cdots + \lambda_k h_k(x_i, x_j)\right).$$

Though $A(x_i, x_j)$ has a closed-form expression, the two regularity functions don't. Hence, $\hat{c}_e(x_i, x_j)$ need to be determined numerically. (See Section 1 of the supplementary material for the determination of the two regularity functions and the Lagrange multipliers.)

Let $\boldsymbol{C}([0, 1]^2)$ denote the space of continuous functions defined on the unit square $[0, 1]^2$. Though $\hat{c}_e(x_i, x_j)$ is minimally informative, it may not well approximate the underlying true copula density $c_e(x_i, x_j)$. We want to approximate $c_e(x_i, x_j)$ by $\hat{c}_e(x_i, x_j)$ to any required degree, which is accomplished by letting $h_1(x_i, x_j), \ldots, h_k(x_i, x_j)$ be elements of a particular basis for the space $\boldsymbol{C}([0, 1]^2)$. Specifically, define $\mathcal{C}(f)$ by

$$\mathcal{C}(f) = \left\{c_{\dot{e}_1\dot{e}_2|D_e}(\cdot) : \forall e \in \mathcal{E}(\mathcal{V})\right\}.$$

Namely, $\mathcal{C}(f)$ is the set of all possible bivariate copula densities originated from the multivariate distribution $f(x_1, \ldots, x_n)$ and therefore is infinite. Furthermore, define $\mathcal{L}(f)$ by

$$\begin{aligned}\mathcal{L}(f) &= \{\log(c) : c \in \mathcal{C}(f)\} \\ &= \left\{\log(c_{\dot{e}_1\dot{e}_2|D_e}(\cdot)) : \forall e \in \mathcal{E}(\mathcal{V})\right\}.\end{aligned}$$

It has been proved by Bedford et al. (2016) that the set $\mathcal{C}(f)$ (and therefore $\mathcal{L}(f)$) is relatively compact in the space $\boldsymbol{C}([0, 1]^2)$. Therefore, by selecting sufficiently many basis functions, $\{h_1(x_i, x_j), \ldots, h_k(x_i, x_j)\}$, from a particular basis for the space $\boldsymbol{C}([0, 1]^2)$, we can approximate $\log(c_e(x_i, x_j))$ to any required degree $\epsilon(>0)$ by a linear combination of $h_1(x_i, x_j), \ldots, h_k(x_i, x_j)$:

$$\sup_{(x_i, x_j)\in[0,1]^2} || \log\left(c_e(x_i, x_j)\right) - \lambda_1 h_1(x_i, x_j) - \cdots - \lambda_k h_k(x_i, x_j)|| < \epsilon.$$

Then $\hat{c}_e(x_i, x_j)$ defined in Eq. (4) shall well approximate the true copula density $c_e(x_i, x_j)$. Here, the metric employed on the space $\boldsymbol{C}([0, 1]^2)$ is the sup norm, hence only requiring the continuity of $c_e(x_i, x_j)$.

For later reference, we call the set of basis functions $\{h_1(x_i, x_j), \ldots, h_k(x_i, x_j)\}$ as "information set." We further explain Example 2 from a backward point of view. We knew that the set $\mathcal{C}(f)$ is relatively compact in the

space $\boldsymbol{C}([0,1]^2)$. Let $\{h_1(\cdot),\ldots,h_k(\cdot),\ldots\}$ be a countable set of basis functions that span the space $\boldsymbol{C}([0,1]^2)$. Then for any copula density in $\mathcal{C}(f)$ and any required level ϵ, we can select from $\{h_1(\cdot),\ldots,h_k(\cdot),\ldots\}$ finite appropriate basis functions whose linear combination can approximate the copula density to the required level. However, the resulted linear combination is not minimally informative. Then we turn back to the logarithmic counterpart of $\mathcal{C}(f)$, i.e., $\mathcal{L}(f)$. Due to the one-to-one correspondence, $\mathcal{L}(f)$ is also a relatively compact set in the space $\boldsymbol{C}([0,1]^2)$. Therefore, for any element in $\mathcal{L}(f)$ and any required level $\delta(>0)$, we can select from $\{h_1(\cdot),\ldots,h_k(\cdot),\ldots\}$ finite appropriate basis functions whose linear combination can approximate the element to the required level δ. By the theoretical work given in Nussbaum (1989) and Borwein et al. (1994), we are able to derive the minimally informative copula density from the linear combination, i.e., Eq. (4). The derived minimally informative copula density well approximates the true underlying copula density.

There are many bases for the space $\boldsymbol{C}([0,1]^2)$, e.g., $\{x_i^p x_j^q : p,q \geq 0\}$. Note that we are not selecting basis functions from a whole basis, which is impossible and unnecessary. We are indeed selecting from a finite set, e.g., $\{x_i^p x_j^q : 0 \leq p,q \leq r\}$ with an appropriate power limit r. Theoretically, by letting k be sufficiently large, we can well approximate any copula density in $\mathcal{C}(f)$ by the same information set $\{h_1(x_i,x_j),\ldots,h_k(x_i,x_j)\}$. However, more basis functions will bring about more λ_ℓ's to be estimated and, consequently, more computational load. Hence, when approximating an individual copula density, we can determine the entry of a basis function into the information set according to its contribution to the approximation. Specifically, let $\{(x_i^{(v)},x_j^{(v)}) : 1 \leq v \leq m\}$ denote a sample of m data points from $c_e(x_i,x_j)$. Let B denote a finite set of candidate basis functions, e.g., $B = \{x_i^p x_j^q : 0 \leq p,q \leq r\}$. The procedure for selecting basis functions is outlined in Algorithm 1.

Algorithm 1 Information Set Determination

1: Set the information set to be empty.
2: Select from B the basis function that yields the largest value of the log-likelihood

$$\sum_{v=1}^{m} \log(\hat{c}_e(x_i^{(v)},x_j^{(v)})). \tag{5}$$

3: **while** the stopping criterion is not met **do**
4: Move the selected basis function from B into the information set.
5: Select from B the basis function which, together with the basis functions in the information set, yields the largest value of the log-likelihood (5).
6: **end while**

For ease of exposition, in what follows, we refer the log-likelihood from fitting a minimally informative copula to a data set as "estimated log-likelihood" and refer the log-likelihood from fitting the true underlying copula to a data set as "true log-likelihood." We should note the following points.

- The stopping criterion in Algorithm 1 could be the maximal number of basis functions or the minimal improvement in the estimated log-likelihood. By adding a new basis function to the information set, the estimated log-likelihood will always increase. Hence, there is no optimal number of basis functions. The choice of the number of basis functions involves the trade-off between approximation accuracy and computational load.
- Selecting basis functions according to the estimated log-likelihood is in consistent with information minimization. It is well known that the values of the Lagrange multipliers λ_ℓ $(\ell = 1,\ldots,k)$, satisfying the expectation constraints, are also maximum likelihood estimates (see, e.g., Barron and Sheu 1991).

A distinguishing feature of our approach is that we have made no assumption on the structure of the underlying multivariate distribution $f(x_1,\ldots,x_n)$, except that all the (conditional) bivariate copulas should be continuous. The expectation constraints are extracted from available data or expert judgement, and will be further studied in the following section.

4 Conditional copula approximation

For the practical implementation of our approach, one fundamental problem needs to be solved: how to evaluate $\{\alpha_1,\ldots,\alpha_k\}$ according to the data at hand. In Sect. 3, we took, for example, an edge in $\mathcal{E}_1$. Therein, α_ℓ $(\ell = 1,\ldots,k)$ can be readily evaluated by calculating the sample mean of the random variable $h_\ell(X_i,X_j)$. For example, if $h_k(X_i,X_j) = X_iX_j$, then α_k can be approximated by the sample mean of X_iX_j: $\alpha_k = \frac{1}{m}\sum_{v=1}^{m} x_i^{(v)}x_j^{(v)}$. It should be noted that for an edge e in tree T_i $(i = 2,\ldots,n-1)$, the conditioning set D_e is no longer empty. For notational simplicity, we define $U_{\mathbf{x}_e} = F_{\dot{e}_1|D_e}(X_{\dot{e}_1}|\,\boldsymbol{X}_e = \mathbf{x}_e)$ and $W_{\mathbf{x}_e} = F_{\dot{e}_2|D_e}(X_{\dot{e}_2}|\,\boldsymbol{X}_e = \mathbf{x}_e)$. Let $u_{\mathbf{x}_e}$ (resp. $w_{\mathbf{x}_e}$) denote the generic value of $U_{\mathbf{x}_e}$ (resp. $W_{\mathbf{x}_e}$). The expectation constraints now should take into account the value of $\boldsymbol{X}_e$:

$$\alpha_\ell(\mathbf{x}_e) = \int_0^1\int_0^1 c_{\dot{e}_1\dot{e}_2|D_e}\left(u_{\mathbf{x}_e},w_{\mathbf{x}_e}|\,\boldsymbol{X}_e = \mathbf{x}_e\right) \times h_\ell(u_{\mathbf{x}_e},w_{\mathbf{x}_e})\mathrm{d}u_{\mathbf{x}_e}\mathrm{d}w_{\mathbf{x}_e}. \tag{6}$$

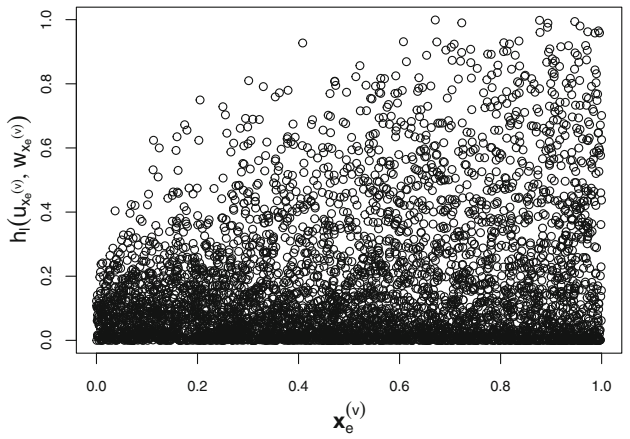

Fig. 2 An illustrative scatter plot of $\{(\mathbf{x}_e^{(v)},\ h_\ell(u_{\mathbf{x}_e^{(v)}},\ w_{\mathbf{x}_e^{(v)}})) : 1 \le v \le 5000\}$

Clearly, $\alpha_\ell(\cdot)$ is now a function of $\boldsymbol{X}_e$—the conditioning random vector related to the edge e. If, at a point $\boldsymbol{X}_e = \mathbf{x}_e$, we have a sizable sample of $(U_{\mathbf{x}_e}, W_{\mathbf{x}_e})$, we intuitively estimate $\alpha_\ell(\mathbf{x}_e)$ by the sample mean of the random variable $h_\ell(U_{\mathbf{x}_e}, W_{\mathbf{x}_e})$. Furthermore, if there are sufficiently many realizations of the conditioning vector—conditioned on each of which we have a sizable sample of $(U_{\mathbf{x}_e}, W_{\mathbf{x}_e})$—we will be able to approximate the functional form of $\alpha_\ell(\boldsymbol{X}_e)$, which is a classical regression problem. Apparently, collected real-life data will never be what we are fancying here. Because $\{X_1, \ldots, X_n\}$ are continuous random variables, for any point $\boldsymbol{X}_e = \mathbf{x}_e$, there will be only one realization of the random variable $h_\ell(U_{\mathbf{x}_e}, W_{\mathbf{x}_e})$. We certainly cannot replace "estimating $\alpha_\ell(\mathbf{x}_e)$ by the sample mean of $h_\ell(U_{\mathbf{x}_e}, W_{\mathbf{x}_e})$" with "estimating $\alpha_\ell(\mathbf{x}_e)$ by one realization of $h_\ell(U_{\mathbf{x}_e}, W_{\mathbf{x}_e})$." One simple explanation is that we conventionally treat a realization of a random variable as the mode of the distribution of that random variable (e.g., when conducting maximum likelihood estimation). The mean and the mode of a distribution usually take different values, and the relationship between them changes from distribution to distribution. As stated in "Appendix," under some mild assumptions, $\alpha_\ell(\mathbf{x}_e)$ (and, therefore, the Lagrange multipliers $\{\lambda_1, \ldots, \lambda_k\}$) is a continuous function of $\mathbf{x}_e$. One may suggest to approximate such function by fitting a regression model to the data $\{(\mathbf{x}_e^{(v)},\ h_\ell(u_{\mathbf{x}_e^{(v)}},\ w_{\mathbf{x}_e^{(v)}})) : 1 \le v \le m\}$ (with $\{\mathbf{x}_e^{(v)} : 1 \le v \le m\}$ being the predictors). However, the fact is that there is no deterministic relationship between $h_\ell(u_{\mathbf{x}_e^{(v)}},\ w_{\mathbf{x}_e^{(v)}})$ and $\mathbf{x}_e^{(v)}$; the scatter plot of the data $\{(\mathbf{x}_e^{(v)},\ h_\ell(u_{\mathbf{x}_e^{(v)}},\ w_{\mathbf{x}_e^{(v)}})) : 1 \le v \le m\}$ is rather erratic (see Fig. 2).

We have to come up with an efficient surrogate for the sample mean of the random variable $h_\ell(U_{\mathbf{x}_e}, W_{\mathbf{x}_e})$.

Bedford et al. (2016) approached the above problem by dividing the domain of $\boldsymbol{X}_e$ into equal-volume subregions and assuming that the copula density $c_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$ dose not change when $\boldsymbol{X}_e$ varies within an individual subregion. Evidently, their approach suffers from certain inherent drawbacks. For example, the partition of the domain of $\boldsymbol{X}_e$ is rather subjective. Even if they divide the domain by using, say, the CART (classification and regression tree), the fitted copula density is still not appealing: It is bumpy. In the following, we propose to relax the conditional expectation (6) and compute an average over a neighborhood of $\boldsymbol{X}_e = \mathbf{x}_e$, which is achieved by invoking the kernel-regression technique; see (Hastie et al. 2009, Chapter 6) for a brief introduction on kernel regression. Compared with parametric methods, kernel smoothing methods have the advantage that they make relatively milder structural assumptions. By employing kernel smoothing methods, two approximations are happening here:

- expectation is approximated by averaging over sample data;
- conditioning at a point is relaxed to conditioning on a region encircling that point.

Remark 2 Note that Eq. (6) provides a method to test if the simplifying assumption can be employed for a particular edge. Specifically, if the simplifying assumption holds, then the conditional copula $c_{\dot{e}_1\dot{e}_2|D_e}\left(u_{\mathbf{x}_e}, w_{\mathbf{x}_e} | \boldsymbol{X}_e = \mathbf{x}_e\right)$ does not depend on the value of $\boldsymbol{X}_e$. Therefore, $\alpha_\ell(\mathbf{x}_e)$ should be a constant:

$$\alpha_\ell = \int_0^1 \int_0^1 c_{\dot{e}_1\dot{e}_2|D_e}\left(u_{\mathbf{x}_e}, w_{\mathbf{x}_e}\right) h_\ell(u_{\mathbf{x}_e}, w_{\mathbf{x}_e}) \mathrm{d}u_{\mathbf{x}_e} \mathrm{d}w_{\mathbf{x}_e}.$$

Hence, for each value of $\boldsymbol{X}_e = \mathbf{x}_e$, we calculate $\alpha_\ell(\mathbf{x}_e)$. If $\alpha_\ell(\mathbf{x}_e)$ is constant (or varies within a small range), then we might employ the simplifying assumption. Otherwise, if $\alpha_\ell(\mathbf{x}_e)$ varies within a wide range or demonstrates an obvious relationship with $\mathbf{x}_e$, then we cannot employ the simplifying assumption.

4.1 Weighted average and weighted linear regression

For an edge $e \in \mathcal{E}_i$ $(2 \le i \le n-1)$, $X_{\dot{e}_1}$ and $X_{\dot{e}_2}$ are the two conditioned random variables, and $\boldsymbol{X}_e$ is the conditioning random vector (having $(i-1)$ elements). Let $(x_{\dot{e}_1}^{(v)},\ x_{\dot{e}_2}^{(v)},\ \mathbf{x}_e^{(v)})$ denote the vth realization of $(X_{\dot{e}_1},\ X_{\dot{e}_2},\ \boldsymbol{X}_e)$ for $v = 1, \ldots, m$. We now approximate the conditional expectation of the random variable $h_\ell(U_{\mathbf{x}_e},\ W_{\mathbf{x}_e})$, for $1 \le \ell \le k$ and an arbitrary point $\boldsymbol{X}_e = \mathbf{x}_e$.

Let $K_\mu(\mathbf{x}_e,\ \mathbf{x})$ be a kernel function, allocating an appropriate weight to $\mathbf{x}$ $(\in [0,1]^{i-1})$ according to its distance from $\mathbf{x}_e$. For example, the radial Epanechnikov kernel is defined by

$$K_\mu(\mathbf{x}_e,\ \mathbf{x}) = D\left(\frac{||\mathbf{x}_e - \mathbf{x}||}{\mu}\right),$$

in which $||\cdot||$ is the Euclidean norm, and

$$D(z) = \begin{cases} 3(1-z^2)/4, & \text{if } |z| < 1; \\ 0, & \text{otherwise.} \end{cases}$$

Here, the parameter $\mu(>0)$, controlling the range of the local neighborhood, is usually called the bandwidth or window width. (Section 2.2 of the supplementary material discusses the high-dimensional problem for local learning methods.) The normal kernel with $D(z) = \phi(z)$ is another popular kernel, where $\phi(z)$ is the standard normal density function. The radial Epanechnikov kernel is optimal in terms of mean squared error (Epanechnikov 1969), while the normal kernel is more mathematically tractable. Intuitively, we can approximate $\alpha_\ell(\mathbf{x}_e)$ by the Nadaraya–Watson kernel-weighted average $\hat{\alpha}_\ell(\mathbf{x}_e, \mu)$:

$$\hat{\alpha}_\ell(\mathbf{x}_e, \mu) = \frac{\sum_{v=1}^m K_\mu(\mathbf{x}_e,\ \mathbf{x}_e^{(v)}) h_\ell(u_{\mathbf{x}_e^{(v)}},\ w_{\mathbf{x}_e^{(v)}})}{\sum_{v=1}^m K_\mu(\mathbf{x}_e,\ \mathbf{x}_e^{(v)})}, \tag{7}$$

in which $u_{\mathbf{x}_e^{(v)}} = F_{\dot{e}_1|D_e}(x_{\dot{e}_1}^{(v)} |\ \boldsymbol{X}_e = \mathbf{x}_e^{(v)})$ and $w_{\mathbf{x}_e^{(v)}} = F_{\dot{e}_2|D_e}(x_{\dot{e}_2}^{(v)} |\ \boldsymbol{X}_e = \mathbf{x}_e^{(v)})$. The weighted average $\hat{\alpha}_\ell(\mathbf{x}_e, \mu)$ puts more weight on the data points that fall within distance μ from $\mathbf{x}_e$.

One drawback of locally weighted average is that it can be biased approaching the boundary of the domain of $\boldsymbol{X}_e$, because of the asymmetry of the data near the boundary. Locally weighted linear regression can help reduce bias dramatically at a modest cost in variance (Fan 1992). It exploits the fact that, over a small enough subset of the domain, any sufficiently nice function can be well approximated by an affine function. (See "Appendix" for the continuity and differentiability of the function $\alpha_\ell(\cdot)$.)

However, locally weighted linear regression cannot be directly applied here, because it may return an impractical estimate of $\alpha_\ell(\mathbf{x}_e)$. We take the polynomial basis $\{x_i^p x_j^q\colon\ p, q\ \geq\ 0\}$, for example. For any basis function from the polynomial basis, the value of the conditional expectation $\alpha_\ell(\mathbf{x}_e)$ should falls within the interval $(0, 1)$. However, locally weighted linear regression cannot guarantee that its estimate falls into the interval $(0, 1)$. To avoid impractical estimates, we put an inequality restriction on regression coefficients. The inequality-constrained weighted linear regression approach to approximating $\alpha_\ell(\mathbf{x}_e)$ proceeds as follows. Let $\boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_{i-1})$ be a vector of coefficients. At any point $\boldsymbol{X}_e = \mathbf{x}_e$, solve the following inequality-constrained least-square problem:

$$\hat{\boldsymbol{\theta}}(\mathbf{x}_e, \mu) = \arg\min_{\boldsymbol{\theta}} \sum_{v=1}^m K_\mu(\mathbf{x}_e,\ \mathbf{x}_e^{(v)}) \times \left[h_\ell(u_{\mathbf{x}_e^{(v)}},\ w_{\mathbf{x}_e^{(v)}}) - (1,\ \mathbf{x}_e^{(v)})\boldsymbol{\theta}^{\mathrm{tr}}\right]^2,$$

subject to $0 < (1,\ \mathbf{x}_e)\boldsymbol{\theta}^{\mathrm{tr}} < 1$. Here, "tr" is the transpose operator. Then the optimal estimate of $\alpha_\ell(\mathbf{x}_e)$ is

$$\hat{\alpha}_\ell(\mathbf{x}_e, \mu) = (1,\ \mathbf{x}_e)\hat{\boldsymbol{\theta}}(\mathbf{x}_e, \mu)^{\mathrm{tr}}. \tag{8}$$

Guaranteed by the additional constraint, the estimate $\hat{\alpha}_\ell(\mathbf{x}_e, \mu)$ will always be practical. The estimate $\hat{\boldsymbol{\theta}}(\mathbf{x}_e, \mu)$ can be calculated by Dantzig–Cottle algorithms (Cottle and Dantzig 1968). According to Theorem 1 of Liew (1976), it is easy to prove that, when m is sufficiently large, the inequality-constrained least-square problem will reduce to a non-constrained least-square problem. Note that though we fit a linear model to the data within the neighborhood of $\boldsymbol{X}_e = \mathbf{x}_e$, we only utilize the fitted linear model to evaluate $\hat{\alpha}_\ell(\mathbf{x}_e, \mu)$ at the single point $\boldsymbol{X}_e = \mathbf{x}_e$. Apparently, compared with locally weighted average, inequality-constrained locally weighted linear regression is more computationally demanding.

Like any local learning problem, one needs to determine the optimal range of the neighborhood, i.e., the value of μ. A variety of automatic, data-based methods have been developed for optimizing the bandwidth, with the general consensus that the plug-in technique (Sheather and Jones 1991) and cross-validation technique (Rudemo 1982) are the most powerful; see Köhler et al. (2014) for a recent review. Plug-in methods require subjective estimators of certain unknown functions and perform badly in multivariate regression. Hence, we here illustrative the leave-one-out cross-validation approach to determining the optimal bandwidth μ. For five- or tenfold cross-validation, the appropriate translations are obvious.

For an edge $e \in \mathcal{E}_i$ $(2 \leq i \leq n-1)$, given the data $\{(x_{\dot{e}_1}^{(v)}, x_{\dot{e}_2}^{(v)}, \mathbf{x}_e^{(v)})\colon 1 \leq v \leq m\}$, we in turn approximate $\alpha_\ell(\mathbf{x}_e^{(v)})$ for $1 \leq v \leq m$. For a particular v and given the information set $\{h_1(u, w), \ldots, h_k(u, w)\}$, utilize the remaining data $\{(x_{\dot{e}_1}^{(j)},\ x_{\dot{e}_2}^{(j)},\ \mathbf{x}_e^{(j)})\colon 1 \leq j \neq v \leq m\}$ and Eqs. (7) or (8) to calculate the estimate of $\alpha_\ell(\mathbf{x}_e^{(v)})$, denoted by $\hat{\alpha}_\ell(\mathbf{x}_e^{(v)}, \mu)$ for $\ell = 1, \ldots, k$. According to $\{\hat{\alpha}_1(\mathbf{x}_e^{(v)}, \mu), \cdots, \hat{\alpha}_k(\mathbf{x}_e^{(v)}, \mu)\}$, we can readily approximate the true copula density $c_{\dot{e}_1\dot{e}_2|D_e}(u,\ w|\ \boldsymbol{X}_e = \mathbf{x}_e^{(v)})$ by $\hat{c}_{\dot{e}_1\dot{e}_2|D_e}\left(u, w|\ \boldsymbol{X}_e = \mathbf{x}_e^{(v)};\ \mu\right)$:

$$\hat{c}_{\dot{e}_1\dot{e}_2|D_e}\left(u, w|\boldsymbol{X}_e = \mathbf{x}_e^{(v)};\ \mu\right) = d_1^{(v)}(u; \mu) d_2^{(v)}(w; \mu) \times \exp\left(\lambda_1^{(v)}(\mu) h_1(u, w) + \cdots + \lambda_k^{(v)}(\mu) h_k(u, w)\right),$$

where the two regularity functions, $d_1^{(v)}(u;\mu)$ and $d_2^{(v)}(w;\mu)$, and the Lagrange multipliers $\{\lambda_1^{(v)}(\mu), \ldots, \lambda_k^{(v)}(\mu)\}$ are all calculated in the manner of Section 1 in the supplementary material.

The optimal bandwidth for edge e, denoted by $\mu^*_{\dot{e}_1\dot{e}_2|D_e}$, is

$$\mu^*_{\dot{e}_1\dot{e}_2|D_e} = \arg\max_{\mu} \sum_{v=1}^{m} \log \left(\hat{c}_{\dot{e}_1\dot{e}_2|D_e}\left(u_{\mathbf{x}_e^{(v)}}, w_{\mathbf{x}_e^{(v)}} | \boldsymbol{X}_e = \mathbf{x}_e^{(v)};\ \mu\right)\right).$$

Note that the optimal bandwidths for different edges are different.

4.2 Basis function selection for conditional copulas

Recall that for an unconditional copula, we select from B the basis function that most improves the log-likelihood (5); if we have m data points, then the log-likelihood is a summation of m elements. However, for each conditional copula, the corresponding log-likelihood contains only one element. For example, if we have data $\{(x_{\dot{e}_1}^{(v)}, x_{\dot{e}_2}^{(v)}, \mathbf{x}_e^{(v)}) : 1 \le v \le m\}$ for edge e, then we will have m different conditional copulas: $c_{\dot{e}_1\dot{e}_2|D_e}\left(u, w | \boldsymbol{X}_e = \mathbf{x}_e^{(v)}\right)$, for $1 \le v \le m$. Of all the m data points $\{(u_{\mathbf{x}_e^{(v)}}, w_{\mathbf{x}_e^{(v)}}) : 1 \le v \le m\}$, only the datum $(u_{\mathbf{x}_e^{(v)}}, w_{\mathbf{x}_e^{(v)}})$ comes from the conditional copula $c_{\dot{e}_1\dot{e}_2|D_e}(u, w | \boldsymbol{X}_e = \mathbf{x}_e^{(v)})$. Therefore, the estimated log-likelihood contains only one element: $\log(\hat{c}_{\dot{e}_1\dot{e}_2|D_e}(u_{\mathbf{x}_e^{(v)}}, w_{\mathbf{x}_e^{(v)}} | \boldsymbol{X}_e = \mathbf{x}_e^{(v)}; \mu))$. If we select basis functions for $c_{\dot{e}_1\dot{e}_2|D_e}\left(u, w | \boldsymbol{X}_e = \mathbf{x}_e^{(v)}\right)$ according to the estimated log-likelihood, then the selected basis functions are optimal only in terms of the single datum $(x_{\dot{e}_1}^{(v)}, x_{\dot{e}_2}^{(v)}, \mathbf{x}_e^{(v)})$. Consequently, the approximating copula $\hat{c}_{\dot{e}_1\dot{e}_2|D_e}(u, w | \boldsymbol{X}_e = \mathbf{x}_e^{(v)}; \mu)$ will be overfitting. Another problem related to selecting basis functions for conditional copulas is the huge computational load. If we are dealing with an n-variate vine copula, then we will have to approximate $m \times \frac{(n-1)(n-2)}{2}$ conditional bivariate copulas. Even worse, taking account of cross-validation, if we determine the optimal value of μ among, say, a set of 100 values, then the computational load is $100 \times m \times \frac{(n-1)(n-2)}{2}$.

To alleviate the above-mentioned two problems, we here develop a two-stage procedure for approximating all the conditional copulas related to an edge $e \in \mathcal{E}_i$ $(2 \le i \le n-1)$; see Algorithm 2.

Algorithm 2 Two-Stage Procedure

procedure STAGE ONE
  Divide the domain of $\boldsymbol{X}_e$ into z regions: $[0,1]^{i-1} = R_1 \cup R_2 \cup \cdots \cup R_z$.
  Divide the data $\{(u_{\mathbf{x}_e^{(v)}}, w_{\mathbf{x}_e^{(v)}}) : 1 \le v \le m\}$ into z subsets: $\{(u_{\mathbf{x}_e^{(v)}}, w_{\mathbf{x}_e^{(v)}}) : \mathbf{x}_e^{(v)} \in R_j, 1 \le v \le m\}$, $j = 1, \ldots, z$.
  for $j = 1, \ldots, z$ **do**
    Treat the data $\{(u_{\mathbf{x}_e^{(v)}}, w_{\mathbf{x}_e^{(v)}}) : \mathbf{x}_e^{(v)} \in R_j, 1 \le v \le m\}$ as coming from an unconditional copula and determine the information set, denoted by B_j, for them by using Algorithm 1.
  end for
end procedure
procedure STAGE TWO
  Let μ denote the value of the bandwidth.
  for $1 \le v \le m$ **do**
    If $\mathbf{x}_e^{(v)} \in R_j$, then the information set used for approximating the conditional copula $c_{\dot{e}_1\dot{e}_2|D_e}(u, w | \boldsymbol{X}_e = \mathbf{x}_e^{(v)})$ is B_j.
    Use a local learning method to calculate $\{\hat{\alpha}_1(\mathbf{x}_e^{(v)}, \mu), \ldots, \hat{\alpha}_k(\mathbf{x}_e^{(v)}, \mu)\}$, where $k = \#B_j$.
    Use the method developed in Section 1 of the supplementary material to calculate the Lagrange multipliers $\{\lambda_1^{(v)}(\mu), \cdots, \lambda_k^{(v)}(\mu)\}$ and numerically determine the two regularity functions $d_1^{(v)}(u;\mu)$ and $d_2^{(v)}(w;\mu)$.
    The minimally informative copula for $c_{\dot{e}_1\dot{e}_2|D_e}\left(u, w | \boldsymbol{X}_e = \mathbf{x}_e^{(v)}\right)$ is

$$\hat{c}_{\dot{e}_1\dot{e}_2|D_e}\left(u, w | \boldsymbol{X}_e = \mathbf{x}_e^{(v)}; \mu\right) = d_1^{(v)}(u;\mu) d_2^{(v)}(w;\mu) \exp\left(\lambda_1^{(v)}(\mu) h_1(u,w) + \cdots + \lambda_k^{(v)}(\mu) h_k(u,w)\right),$$

    where $\{h_1(u,w), \ldots, h_k(u,w)\}$ are the basis functions in B_j.
  end for
  Calculate

$$\sum_{v=1}^{m} \log\left(\hat{c}_{\dot{e}_1\dot{e}_2|D_e}\left(u_{\mathbf{x}_e^{(v)}}, w_{\mathbf{x}_e^{(v)}} | \boldsymbol{X}_e = \mathbf{x}_e^{(v)};\ \mu\right)\right). \quad (9)$$

  Repeat steps 9–16 for different values of μ and select the optimal one that maximizes the estimated log-likelihood (9).
end procedure

We should note the following points.

– The two-stage procedure utilizes the continuity of $c_{\dot{e}_1\dot{e}_2|D_e}(u, w | \boldsymbol{X}_e = \mathbf{x}_e)$ on $\mathbf{x}_e$. According to Appendix A, the dependence structure $c_{\dot{e}_1\dot{e}_2|D_e}(u, w | \boldsymbol{X}_e = \mathbf{x}_e)$ changes continuously with $\mathbf{x}_e$. Therefore, for a small region, we can assign the same information set to all the conditional copulas whose conditioning variables are in that region. Though the basis functions are the same, the Lagrange multipliers will be different for different conditional copulas.

– For a conditional copula in tree T_2, it has only one conditioning variable. Since every individual variable is uniformly distributed, we can divide the domain (i.e., the [0, 1] interval) into a few equal-length subintervals. For a conditional copula in tree T_i ($i \geq 3$), the elements of the conditioning random vector are mutually dependent. Instead of subjectively dividing the domain of $\boldsymbol{X}_e$ into z regions, we can employ *k-means clustering* to partition the observations $\{\mathbf{x}_e^{(v)} : 1 \leq v \leq m\}$ into z clusters, such that the within-cluster distance is minimized and the between-cluster distance is maximized; see, e.g., Hartigan and Wong (1979). The "kmeans" function in R software serves this purpose.
– If parallel computing is possible, Stage Two can be calculated parallel on each tree level.

5 Numerical study

In this section, we examine the performance of the two-stage procedure via simulated data. Due to lack of space, a real data set is analyzed in the supplementary material.

For illustrative purpose, we focus on the D-vine structure of 6 random variables, with the nodes in tree T_1 from left to right being labeled by $\{X_1, X_2, X_3, X_4, X_5, X_6\}$. All the bivariate copulas originated from $f(x_1, \ldots, x_6)$ are from the same family. For example, all the bivariate copulas originated from a 6-variate t-copula are t-copulas and have the same degrees of freedom.

Three types of bivariate copulas are examined: the Gaussian copula, t-copula and Gumbel copula. The Gaussian copula and t-copula are able to model moderate and/or heavy tails; the Gumbel copula is able to capture asymmetric dependence. In terms of tail dependence, the Gaussian copula is neither lower nor upper tail dependent; the t-copula is both lower and upper tail dependent; the Gumbel copula is upper tail dependent. Due to lack of space, we here only present simulation results of the t-copula. Simulation results of the other copulas are given in the supplementary material. The bivariate t-copula with υ (> 2) degrees of freedom is given by $C_{\rho,\upsilon}(u, w) = t_{\rho,\upsilon}(t_\upsilon^{-1}(u), t_\upsilon^{-1}(w))$, where $t_\upsilon(\cdot)$ is the cdf of the one-dimensional t-distribution with υ degrees of freedom, and $t_{\rho,\upsilon}(\cdot)$ is the cdf of the bivariate t-distribution with υ degrees of freedom and correlation $\rho \in (-1, 1)$. The superscript "-1" denotes the inverse of a function.

The parameter setting for simulating data is described as follows. For the five bivariate t-copulas in tree T_1, the correlation parameter ρ takes in turn (from left to right) the following five values: $\{0.3, 0.4, 0.5, 0.6, 0.7\}$, which indicates that the dependence structure evolves from weak dependence to strong dependence. All the bivariate t-copulas have 3 degrees of freedom. For a conditional bivariate t-copula related to an edge e, the correlation parameter and the conditioning variables have the following relation:

$$\rho(\mathbf{x}_e) = 1.4 \times (\bar{\mathbf{x}}_e - 0.5), \tag{10}$$

in which $\bar{\mathbf{x}}_e$ is the average of the values of the conditioning variables. Consequently, $\alpha_\ell(\mathbf{x}_e)$ is differentiable. Under the above parameter setting, we randomly simulate a sample of 1000 data points from the 6-variate t-copula. The scatter plots for copulas $c_{12}(u, w)$ and $c_{56}(u, w)$ are given in Fig. 3, in which scatter plots for two Gaussian copulas and two Gum-

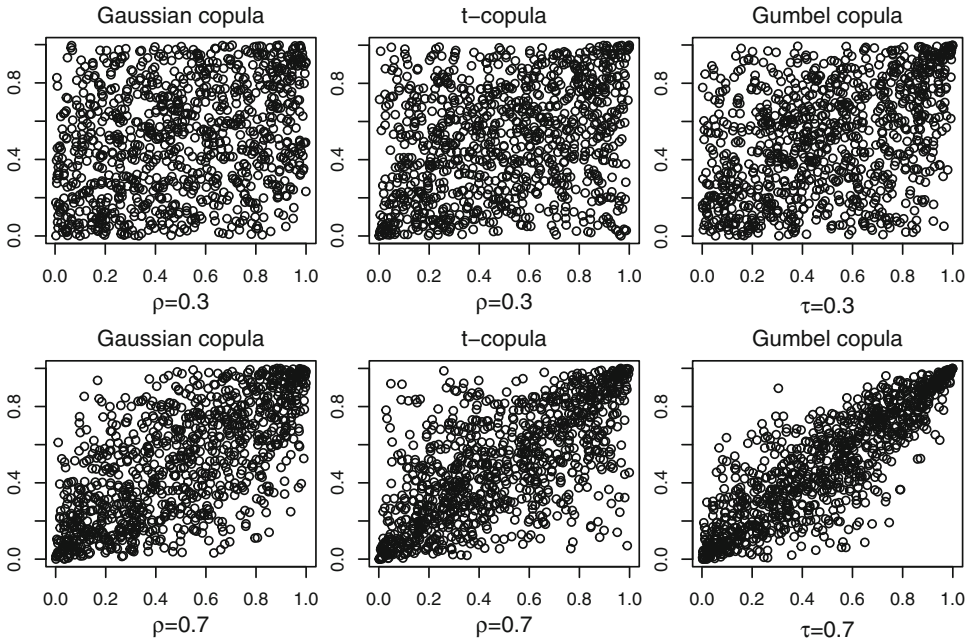

Fig. 3 Scatter plots of samples with size 1000 from different bivariate copulas

Table 1 The evolution of the estimated log-likelihood by gradually adding basis functions: t-copula

	Copula	ρ	Estimated log-likelihood	True log-likelihood
B	$c_{12}(u, w)$	0.3	{46.7498, 72.9410, 78.0555, 80.8781, 81.9875}	95.7630
	$c_{23}(u, w)$	0.4	{56.2788, 100.0158, 108.9762, 112.5108, 115.3198}	135.3031
	$c_{34}(u, w)$	0.5	{78.4825, 124.6716, 161.8266, 164.9900, 166.4577}	172.6276
	$c_{45}(u, w)$	0.6	{114.0621, 204.3366, 238.3059, 241.0224, 245.2159}	246.2912
	$c_{56}(u, w)$	0.7	{168.3862, 308.7592, 350.1087, 353.5052, 355.7738}	366.3090
P	$c_{12}(u, w)$	0.3	{49.1366, 53.8103, 61.8424, 64.0455, 72.5107}	95.7630
	$c_{23}(u, w)$	0.4	{76.2565, 86.2504, 92.4941, 98.2177, 99.0780}	135.3031
	$c_{34}(u, w)$	0.5	{123.2001, 132.2268, 139.3346, 144.1076, 147.4368}	172.6276
	$c_{45}(u, w)$	0.6	{173.8291, 187.4981, 201.6163, 209.5390, 216.7686}	246.2912
	$c_{56}(u, w)$	0.7	{271.1958, 292.0212, 309.8739, 317.4228, 327.6366}	366.3090

bel copulas are also included. τ represents the Kendall rank correlation coefficient. Figure 3 shows that, when $\rho = 0.3$ or $\tau = 0.3$, the simulated data do not have an evident pattern; when $\rho = 0.7$ or $\tau = 0.7$, the relation between the involved two random variables is noticeable from the data.

Two different bases are used to construct two different families of minimally informative copulas: the Bernstein basis functions, $\left\{\binom{6}{p}u^p(1-u)^{6-p}\binom{6}{q}w^q(1-w)^{6-q} : 0 \le p, q \le 6\right\}$, and the polynomial basis functions $\{u^p w^q : 0 \le p, q \le 6\}$. Here the polynomial degree is 6. (Via intensive simulation study, we found that increasing the power to larger than 6 will improve a little the approximation, but will impose a lot of additional computational load.) Although we can approximate a bivariate copula to any required degree, given only a finite set of candidate basis functions, different bases will have different efficiency. Hence, we want to compare Bernstein basis with the polynomial basis.

We now approximate the five bivariate copulas in tree T_1 by minimally informative copulas. The estimated and true log-likelihoods are summarized in Table 1.

In Table 1, rows 2–6 stand for approximating the five copulas by Bernstein basis functions, and rows 7–11 stand for approximating the same five copulas by polynomial basis functions. For example, when approximating $c_{12}(u, w)$, the estimated log-likelihood after selecting the first optimal Bernstein basis function (resp. polynomial basis function) is 46.7498 (resp. 49.1366). After adding the second optimal Bernstein basis function (resp. polynomial basis function), the estimated log-likelihood increases to 72.9410 (resp. 53.8103). After selecting five Bernstein basis functions (resp. polynomial basis functions), the final estimated log-likelihood is 81.9875 (resp. 72.5107), while the true log-likelihood is 95.7630. It is observed from Table 1 that, for Bernstein basis, the joining of the fifth basis function does not contribute much to the estimated log-likelihood. In Table 1, for each edge, the estimated log-likelihood with five Bernstein basis functions is larger than that with five polynomial basis functions. Indeed, for each edge, the estimated log-likelihood with three Bernstein basis functions is already larger than that with four polynomial basis functions, showing the competence of Bernstein basis.

Surface plots and contour plots for copulas $c_{12}(u, w)$ and $c_{56}(u, w)$ are drawn in Figs. 4 and 5.

In Fig. 4 (resp., Fig. 5), the left two panels are the surface plot and contour plot for the true bivariate t-copula $c_{12}(u, w)$ (resp., $c_{56}(u, w)$); the middle two panels are the surface plot and contour plot for the minimally informative copula having five Bernstein basis functions; the right two panels are the surface plot and contour plot for the minimally informative copula having five polynomial basis functions. Figures 4 and 5 show that, compared with the minimally informative copula having polynomial basis functions, the minimally informative copula having Bernstein basis functions bears a stronger resemblance with the true copula. Simulation results in the supplementary material also verified that Bernstein basis is more efficient than polynomial basis. Moreover, it is found that a combination of four Bernstein basis functions is capable of producing a good approximation. Hence, in the following, only Bernstein basis is used, and the cardinality of the information set for a conditional copula is set to be four.

We simulate another sample of 1000 data points from the 6-variate D-vine t-copula and approximate the five bivariate t-copulas in tree T_1 by minimally informative copulas having five Bernstein basis functions selected from $\left\{\binom{6}{p}u^p(1-u)^{6-p}\binom{6}{q}w^q(1-w)^{6-q} : 0 \le p, q \le 6\right\}$. For each t-copula in tree T_1, we calculate the correlation coefficient from the 1000 data points, which is a sample value of ρ for that copula. We denote such a value by $\hat{\rho}$. The correlation between the two random variables of a minimally informative copula can be evaluated via numerical

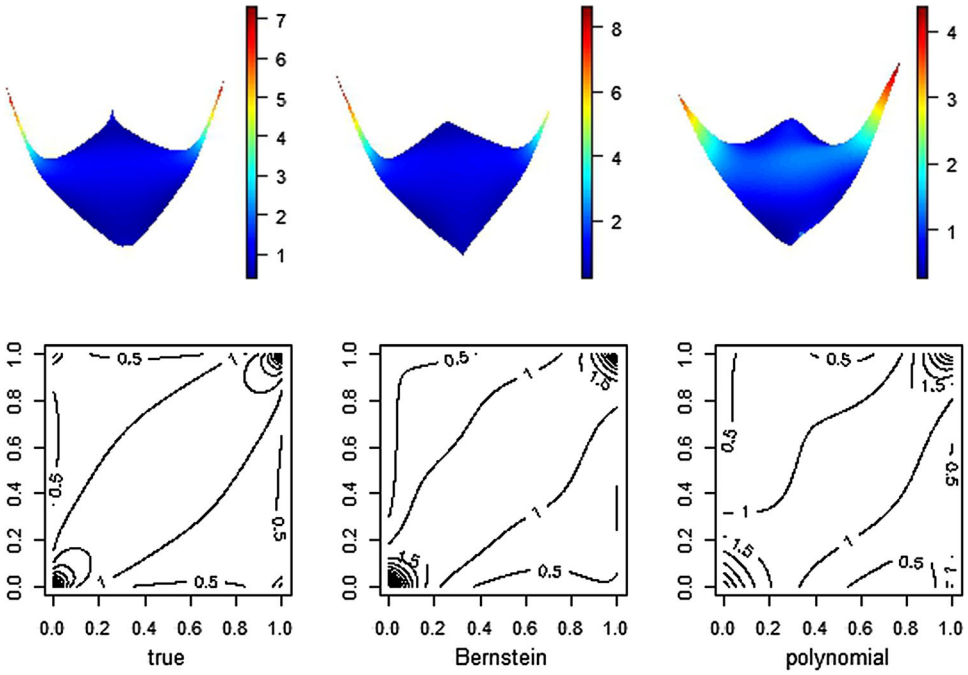

Fig. 4 Surface plots and contour plots of the true bivariate t-copula with $\rho = 0.3$ (*left*), of the minimally informative copula having five Bernstein basis functions (*middle*), and of the minimally informative copula having five polynomial basis functions (*right*)

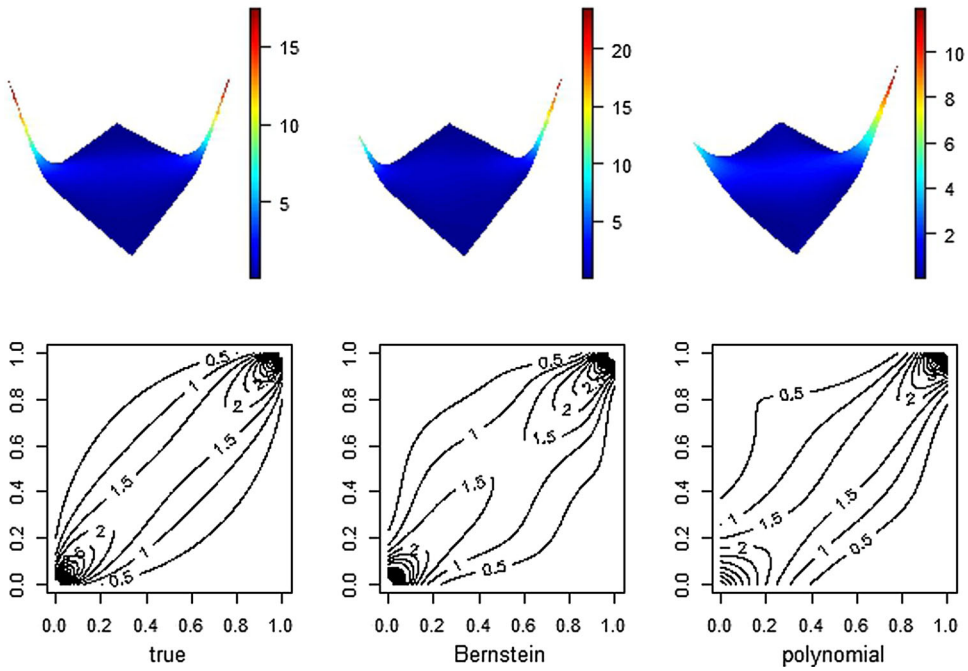

Fig. 5 Surface plots and contour plots of the true bivariate t-copula with $\rho = 0.7$ (*left*), of the minimally informative copula having five Bernstein basis functions (*middle*), and of the minimally informative copula having five polynomial basis functions (*right*)

integration. We denote such a value by $\bar{\rho}$. Apparently, if the minimally informative copula well approximates the true copula, the two sample values, $\hat{\rho}$ and $\bar{\rho}$, should be close. We repeat the above procedure for 100 times and obtain, for each t-copula in tree T_1, two sequences: $\{\hat{\rho}_i : i = 1, \cdots, 100\}$ and $\{\bar{\rho}_i : i = 1, \ldots, 100\}$. We plot them in Fig. 6, in which the five colors {"black," "red," "blue," "green," "purple"}, respectively, correspond to the five t-copulas having correlation coefficients $\{0.3, 0.4, 0.5, 0.6, 0.7\}$. Solid lines represent the sequence $\{\bar{\rho}_i : i = 1, \ldots, 100\}$, and dotted lines represent the sequence $\{\hat{\rho}_i : i = 1, \ldots, 100\}$. It is observed from Fig. 6 that, for each t-copula, the two sample values $\hat{\rho}_i$ and $\bar{\rho}_i$ are close to each other, showing the competence of the minimally informative copula.

We now approximate the conditional copulas in tree T_2 by minimally informative copulas. Note that, since there

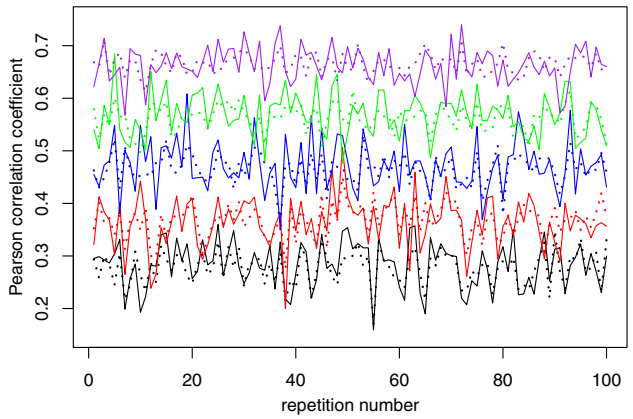

Fig. 6 Evolving paths of the five sequences $\{\hat{\rho}_i : i = 1, \ldots, 100\}$ (*dotted lines*) and of the five sequences $\{\bar{\rho}_i : i = 1, \ldots, 100\}$ (*solid lines*). (Color figure online)

are 1000 data points, there will be 1000×4 different conditional copulas. Separately determining the information set for each of them is time consuming and may result in overfitting. Hence, we divide the $[0, 1]$ interval into four subintervals: $[0, 1/4)$, $[1/4, 2/4)$, $[2/4, 3/4)$ and $[3/4, 1]$. For an edge, e.g., $\{1, 3|2\}$, we group the data $\{(F_{1|2}(x_1^{(v)}|x_2^{(v)}), F_{3|2}(x_3^{(v)}|x_2^{(v)})) : 1 \le v \le 1000\}$ into four subsets, respectively, corresponding to $x_2^{(v)}$ falling into subintervals $[0, 1/4)$, $[1/4, 2/4)$, $[2/4, 3/4)$ and $[3/4, 1]$. Then we employ Algorithm 2 to approximate each of the 1000×4 conditional copulas. To determine the optimal value of μ, we select it from a set of 10 candidates: $\{0.1, 0.2, \ldots, 0.9, 1\}$. The candidate bandwidths are determined to balance the computational load and the approximation accuracy. We approximate $\{\alpha_1(\mathbf{x}_e), \ldots, \alpha_k(\mathbf{x}_e)\}$ using locally weighted average.

The total true log-likelihoods for edges $\{\{1, 3|2\}, \{2, 4|3\}, \{3, 5|4\}, \{4, 6|5\}\}$ are $\{97.9351, 138.6133, 125.5207, 158.8430\}$; the corresponding total estimated log-likelihoods are $\{93.6227, 115.3315, 91.7621, 121.9359\}$. The four optimal bandwidths $\{u^*_{13|2}, u^*_{24|3}, u^*_{35|4}, u^*_{46|5}\}$ are $\{0.3, 0.3, 0.2, 0.3\}$. The total estimated log-likelihoods are close to the total true log-likelihoods, showing the feasibility of the two-stage procedure. To check whether every conditional copula is well approximated, for each edge in T_2 and for each of the 1000 conditional copulas, we calculate the log-likelihood deviation: the true log-likelihood subtracting the estimated log-likelihood. The four sequences of log-likelihood deviations are plotted in Fig. 7.

For example, the top-left panel shows the evolution of the log-likelihood deviation for edge $\{1, 3|2\}$, when the value of X_2 increases from 0 to 1. Figure 7 shows that the true log-likelihood is generally larger than the estimated log-likelihood, and the log-likelihood deviation fluctuates within a small range around zero.

From Remark 2, we know that Eq. (6) can be used to test the simplifying assumption. As here we know the underlying true law, to examine the performance of the two-stage procedure, we can also compare the true expected value $\alpha_\ell(\mathbf{x}_e)$ with its estimate $\hat{\alpha}_\ell(\mathbf{x}_e, \mu^*)$:

$$\hat{\alpha}_\ell(\mathbf{x}_e, \mu^*) = \int_0^1 \int_0^1 \hat{c}_{\dot{e}_1 \dot{e}_2 | D_e}(u, w | \boldsymbol{X}_e = \mathbf{x}_e;\ \mu^*) \, h_\ell(u, w) \mathrm{d}u \mathrm{d}w,$$

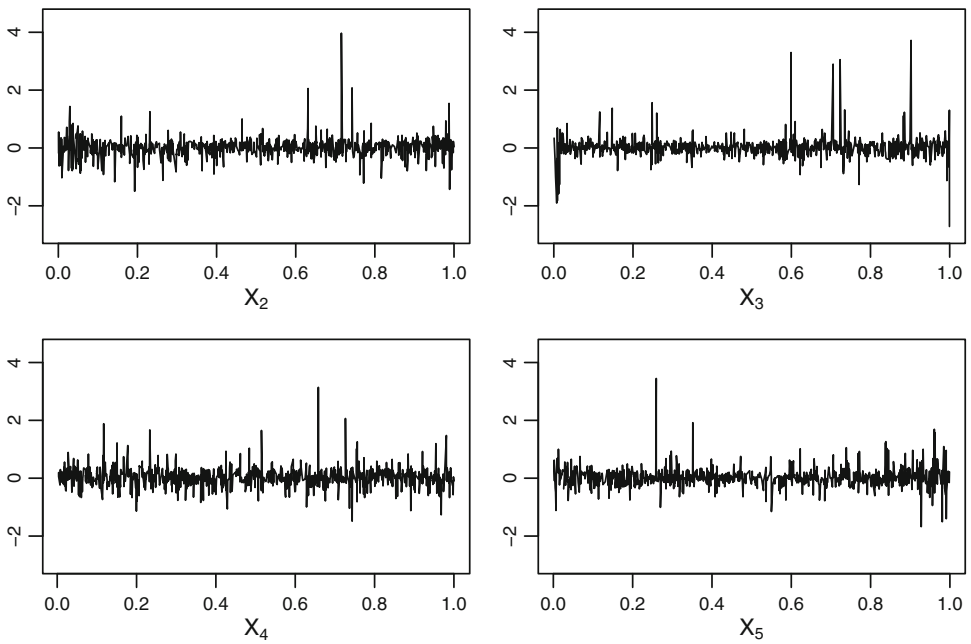

Fig. 7 Evolving paths of the difference between the true log-likelihood and the estimated log-likelihood (tree T_2)

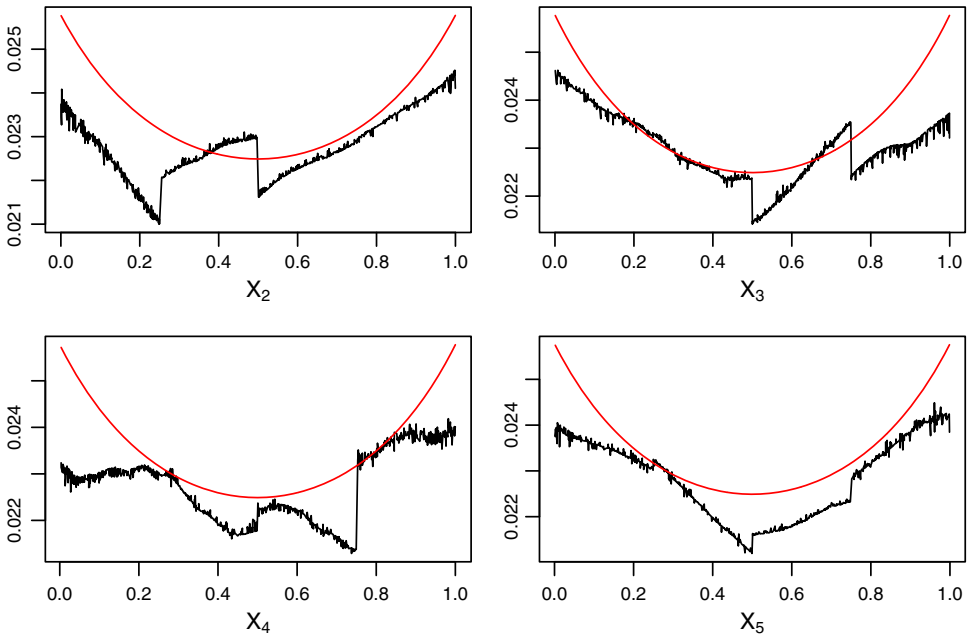

Fig. 8 Evolving paths of the true expected values $\{\alpha_\ell(\mathbf{x}_e)\}$ (*red*) and of the estimated expected values $\{\hat{\alpha}_\ell(\mathbf{x}_e, \mu^*)\}$ (*black*). (Color figure online)

where $\hat{c}_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$ is the approximating minimally informative copula. Note that, for testing purpose, $h_\ell(u, w)$ need not belong to the information set of $\hat{c}_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$. Here we might set $h_\ell(u, w)$ to be $\binom{6}{3}u^3(1-u)^3\binom{6}{3}w^3(1-w)^3$. Figure 8 plots the evolving paths (black) of the four sequences $\{\hat{\alpha}_\ell(x_2^{(v)}, \mu^*_{13|2})\colon 1 \le v \le 1000\}$, $\{\hat{\alpha}_\ell(x_3^{(v)}, \mu^*_{24|3})\colon 1 \le v \le 1000\}$, $\{\hat{\alpha}_\ell(x_4^{(v)}, \mu^*_{35|4})\colon 1 \le v \le 1000\}$ and $\{\hat{\alpha}_\ell(x_5^{(v)}, \mu^*_{46|5})\colon 1 \le v \le 1000\}$. In each panel, the red smooth curve is the evolving path of the true expected values $\{\alpha_\ell(\mathbf{x}_e)\}$. For each of the top two panels, there are two jumps. The bottom left panel has one jump, and the bottom right panel has two small jumps. The jumps in Fig. 8 are due to the fact that we assign different information sets to different subintervals. As we divide the [0, 1] interval into four subintervals, there are at most three jumps. Yet, the four panels in Fig. 8 all have jumps fewer than three, implying that the minimally informative copula evolves slowly even when the value of the conditioning variable crosses a splitting point. Figure 8 shows that the estimated expected values are close to the true expected values; the relative errors are all within the interval $(-0.02, 0.08)$. Figures 7 and 8 (and figures in the supplementary material) show that the two-stage procedure is capable of approximating conditional copulas.

We reuse the previously simulated 100 sets of data points (each with size 1000). For each data set, we have approximated the five bivariate copulas in tree T_1; we now approximate the conditional copulas in tree T_2. Note that, for each edge in tree T_2, the conditioning variable always distributes uniformly in the [0, 1] interval. Hence, it suffices to study only one edge, say edge $\{1, 3|2\}$. For each value of X_2, we first calculate the correlation coefficient of the true conditional copula and then numerically calculate the correlation

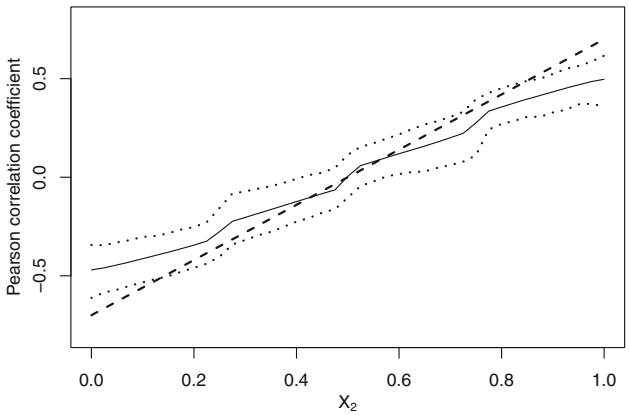

Fig. 9 The true and estimated correlation coefficients, varying with the value of the conditioning variable X_2. The true correlation coefficient is shown as a *dashed curve*, the average of the estimates taken over 100 Monte Carlo samples is displayed by the solid curve and the 90% Monte Carlo confidence intervals are given by *dotted curves*

coefficient of the approximating minimally informative copula that is obtained from one data set. Since we have 100 data sets, we will have 100 such minimally informative copulas. In other words, for each value of X_2, we will have one true correlation coefficient and 100 estimating correlation coefficients. We draw the 90% point-wise confidence intervals at 41 equally spaced grid points from 0 to 1; see Fig. 9.

In Fig. 9, it is observed that when X_2 varies in, say, interval $(0.2, 0.8)$, the estimated correlation coefficient is close to the true correlation coefficient. However, when X_2 is too large or too small, the estimated correlation coefficient is biased. This is because the locally weighted average becomes biased approaching the boundary of the domain of X_2, due to the asymmetry of the data near the boundary. One solution is to

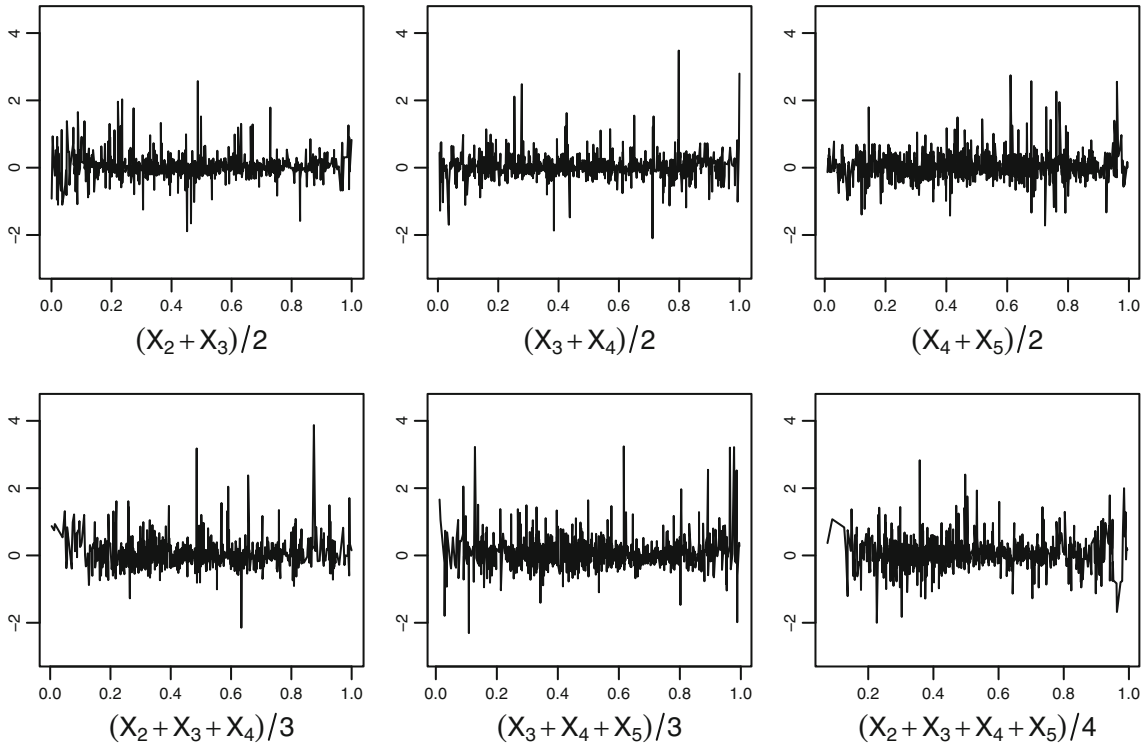

Fig. 10 Evolving paths of the difference between the true log-likelihood and the estimated log-likelihood (trees T_3, T_4 and T_5)

use the locally weighted linear regression. Figures 6 and 9 suggest that minimally informative copulas are competent and the two-stage procedure is robust.

We now approximate the conditional copulas in trees T_3, T_4 and T_5, in the same manner as approximating the conditional copulas in tree T_2, except that we do not divide the domain of the conditioning random vector into equal-volume subregions. We employ k-means clustering to partition the 1000 observations of the conditioning random vector into 4 clusters. For T_3, the longest distance between two points in the domain of $\mathbf{X}_e$ is $\sqrt{2}$; hence, the set of candidate values of μ is $\{0.1, 0.2, \ldots, 1.4\}$. Similarly, for T_4, the set of candidate values of μ is $\{0.1, 0.2, \ldots, 1.7\}$; for T_5, the set of candidate values of μ is $\{0.1, 0.2, \ldots, 2\}$.

The total true log-likelihoods for edges $\{\{1, 4|2, 3\}, \{2, 5|3, 4\}, \{3, 6|4, 5\}\}$ are $\{110.8164, 121.1668, 123.3986\}$; the corresponding total estimated log-likelihoods are $\{71.5413, 93.2329, 86.5560\}$. The three optimal bandwidths $\{u^*_{14|23}, u^*_{25|34}, u^*_{36|45}\}$ are $\{0.6, 0.5, 0.5\}$. The total true log-likelihoods for edges $\{1, 5|2, 3, 4\}$ and $\{2, 6|3, 4, 5\}$ are 108.7813 and 101.3734, respectively; the corresponding total estimated log-likelihoods are 59.4642 and 55.2778. The two optimal bandwidths $u^*_{15|234}$ and $u^*_{26|345}$ are 0.8 and 0.7, respectively. The total true log-likelihood for edge $\{1, 6|2, 3, 4, 5\}$ is 62.5579; the corresponding total estimated log-likelihood is 21.6081. The optimal bandwidth $u^*_{16|2345}$ is 0.9.

The 1000 log-likelihood deviations for each edge in trees T_3, T_4 and T_5 are plotted in Fig. 10.

The upper three panels correspond to tree T_3, the bottom left two panels correspond to tree T_4, and the bottom rightmost panel corresponds to tree T_5. Figures 7 and 10 show that the two-stage procedure performs well; the estimated log-likelihoods are close to the true log-likelihoods. Compared with Fig. 7, the log-likelihood deviation in Fig. 10 fluctuates more violently, implying that the performance of the two-stage procedure deteriorates with the tree level increasing. This is because the estimation error accumulates over tree level.

To show that our final 6-variate minimally informative vine copula well fits the given data, we randomly simulate 1000 data points from it. We first calculate the upper tail-dependence coefficient (Frahm et al. 2005). The upper tail-dependence coefficients of the five bivariate copulas in T_1, calculated from the original data, are $\{0.2837, 0.3546, 0.4145, 0.4848, 0.5776\}$. The upper tail-dependence coefficients of the corresponding copulas, calculated from the simulated data, are $\{0.2648, 0.3442, 0.4324, 0.4892, 0.5334\}$. The upper tail-dependence coefficients from the simulated data are very close to those from the original data. Given a bivariate data set $\{(u_i, w_i) : 1 \leq i \leq m\}$, the bivariate empirical cumulative distribution function (BECDF) is defined as

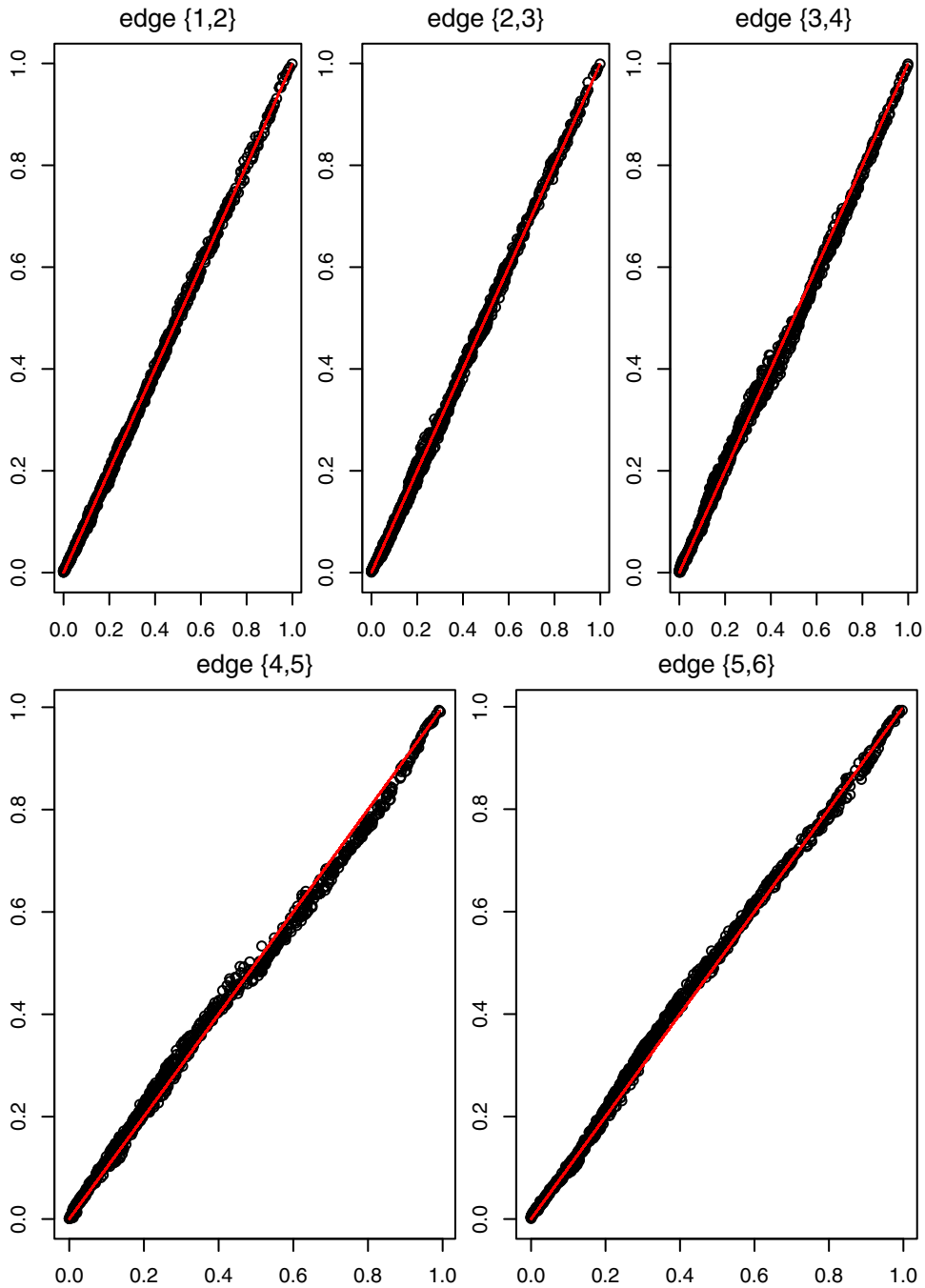

Fig. 11 Q–Q plots for the five copulas in tree T_1. The x-axis represents the quantiles of the simulated data calculated using $\hat{H}_0(u, w)$, and the y-axis represents the quantiles of the simulated data calculated using $\hat{H}_1(u, w)$

$$\hat{H}(u, w) = \frac{\#\{i : u_i \leq u, w_i \leq w\}}{m + 1}.$$

For a bivariate copula in T_1, we let $\hat{H}_0(u, w)$ denote the BECDF obtained from the original data; let $\hat{H}_1(u, w)$ denote the BECDF obtained from the simulated data. We then calculate the quantiles of the simulated data. To draw the Q–Q plot, we calculate two sets of quantiles: one set of quantiles are calculated using $\hat{H}_0(u, w)$ and the other set of quantiles are calculated using $\hat{H}_1(u, w)$. Then the Q–Q plots for the five bivariate copulas in T_1 are given in Fig. 11.

It is clear from Fig. 11 that, for each bivariate copula, the two BECFDs are very close to each other. Consequently, we can conclude that our final 6-variate minimally informative vine copula well fits the given data.

6 Conclusions

In this paper, we addressed the problem of approximating a conditional copula, the parameters of which change with

its conditioning variables. To avoid overfitting and to reduce computational load, we developed a two-stage procedure. Numerical study showed that the two-stage procedure is both feasible and competent. We use (reuse) all the data points that are local; hence, the two-stage procedure can be applied to relatively higher-dimensional vine copulas than the method developed by Bedford et al. (2016). From the illustrative examples, it is clear that modelling data by a vine hierarchy of minimally informative copulas will demand a lot of computing effort. Unfortunately, increased computational load is what we have to pay if we stand by the viewpoint that a conditional copula should change with its conditioning variables—we have to approximate the conditional copula for every configuration of its conditioning variables.

In our approach, there are a number of parameters whose values need to be subjectively determined. After intensive numerical study, we list below some rules of thumb for reference.

- To numerically determine the two regularity functions, $d_1(\cdot)$ and $d_2(\cdot)$, it is acceptable to break up the [0, 1] interval into 200 equal-length subintervals. (We tried four values, {100, 200, 300, 400}, and found that the estimated log-likelihoods are almost the same for 200, 300 and 400.)
- For an unconditional copula, three to five basis functions from $\{\binom{6}{p}u^p(1-u)^{6-p}\binom{6}{q}w^q(1-w)^{6-q} : 0 \le p, q \le 6\}$ yield an acceptable compromise between approximation accuracy and overfitting.
- For a conditional copula, three to four basis functions from $\{\binom{6}{p}u^p(1-u)^{6-p}\binom{6}{q}w^q(1-w)^{6-q} : 0 \le p, q \le 6\}$ yield an acceptable compromise between approximation accuracy and computational load.
- In stage one of the two-stage procedure, the number of regions $\{R_1, \ldots, R_z\}$ is usually determined with the purpose of maintaining a sizable sample in each region.

Note that the optimal number of basis functions for a particular bivariate data set may depend on multiple factors: the data size, the correlation coefficient, the tail-dependence coefficient, etc. For example, if the correlation coefficient is small and we use five basis functions, we may have the overfitting problem. On the other hand, if the tail-dependence coefficient is large and we use three basis functions, the approximation may be of low accuracy. One approach is to split the data into a training set and a testing set. For this issue, more intensive numerical study is needed. To select the optimal (number of) basis functions, an alternative approach is the regularization method: imposing an appropriate penalty on the log-likelihood [see, e.g., Kauermann and Schellhase (2014)]. However, in terms of the set $\{\binom{6}{p}u^p(1-u)^{6-p}\binom{6}{q}w^q(1-w)^{6-q} : 0 \le p, q \le 6\}$, the penalized log-likelihood function will have no less than 49 unknown parameters. Consequently, one has to develop a robust optimization method. Moreover, it is likely that the penalty parameter has to be numerically determined by using cross-validation technique.

This work can be enriched in several ways. One promising direction is to study when we can employ the simplifying assumption. With the number of conditioning variables increasing, their effect will be complicated and may cancel out. Hence, in high-dimensional problems, it may be better to employ the simplifying assumption on copulas in the deeper hierarchy of a vine.

Appendix: Properties of the expected value function

We rewrite Eq. (6) here (dropping the subscript "ℓ"):

$$\alpha(\mathbf{x}_e) = \int_0^1 \int_0^1 c_{\dot{e}_1\dot{e}_2|D_e}(u, w|\, \mathbf{X}_e = \mathbf{x}_e)\, h(u, w)\mathrm{d}u\mathrm{d}w.$$

In Sect. 4, we approximated $\alpha(\mathbf{x}_e)$ by utilizing two local learning methods with the assumption that $\alpha(\mathbf{x}_e)$ is a nice function of $\mathbf{x}_e$. Hence, herein, we study some properties of $\alpha(\mathbf{x}_e)$.

The dependence of a conditional copula on its conditioning variables is commonly expressed through the parameters of the conditional copula. Specifically, if θ is the parameter of $c_{\dot{e}_1\dot{e}_2|D_e}(\cdot)$, then $c_{\dot{e}_1\dot{e}_2|D_e}(u, w|\, \mathbf{X}_e = \mathbf{x}_e)$ is usually formulated as $c_{\dot{e}_1\dot{e}_2|D_e}(u, w; \theta(\mathbf{x}_e))$, indicating that the value of the parameter depends on the conditioning variables. Throughout, we might assume that $\theta(\mathbf{x}_e)$ is differentiable w.r.t. $\mathbf{x}_e$.

To examine the continuity of $\alpha(\mathbf{x}_e)$, we only need to prove that, for all $\epsilon > 0$, there exists a $\delta > 0$, such that for all $\ddot{\mathbf{x}}_e \in [0, 1]^{i-1}$ with $||\ddot{\mathbf{x}}_e - \mathbf{x}_e|| < \delta$, we have that $|\alpha(\ddot{\mathbf{x}}_e) - \alpha(\mathbf{x}_e)| < \epsilon$. A sufficient condition for $\alpha(\mathbf{x}_e)$ to be continuous is that $c_{\dot{e}_1\dot{e}_2|D_e}(u, w; \theta(\mathbf{x}_e))$ is continuous in $\theta(\mathbf{x}_e)$. To see this, we note that because $\theta(\mathbf{x}_e)$ is a continuous function and $c_{\dot{e}_1\dot{e}_2|D_e}(u, w; \theta(\mathbf{x}_e))$ is continuous in $\theta(\mathbf{x}_e)$, then $c_{\dot{e}_1\dot{e}_2|D_e}(u, w; \theta(\mathbf{x}_e))$ is a continuous function of $\mathbf{x}_e$. Hence, for all $\epsilon > 0$ there exists a $\delta > 0$ such that for all $\ddot{\mathbf{x}}_e \in [0, 1]^{i-1}$ with $||\ddot{\mathbf{x}}_e - \mathbf{x}_e|| < \delta$, we have

$$|c_{\dot{e}_1\dot{e}_2|D_e}(u, w;\ \theta(\ddot{\mathbf{x}}_e)) - c_{\dot{e}_1\dot{e}_2|D_e}(u, w;\ \theta(\mathbf{x}_e))| < \frac{\epsilon}{\int_0^1 \int_0^1 h(u, w)dudw},$$

assuming $\int_0^1 \int_0^1 h(u,w)\mathrm{d}u\mathrm{d}w$ is finite. Consequently we have

$$|\alpha(\ddot{\mathbf{x}}_e) - \alpha(\mathbf{x}_e)| < \int_0^1 \int_0^1 |c_{\dot{e}_1\dot{e}_2|D_e}(u,w;\ \theta(\ddot{\mathbf{x}}_e)) - c_{\dot{e}_1\dot{e}_2|D_e}(u,w;\ \theta(\mathbf{x}_e))|h(u,v)\mathrm{d}u\mathrm{d}w < \epsilon.$$

To examine the differentiability of $\alpha(\mathbf{x}_e)$, we take partial derivative w.r.t. x_e^1, the first variable of $\mathbf{x}_e$:

$$\frac{\partial \alpha(\mathbf{x}_e)}{\partial x_e^1} = \int_0^1 \int_0^1 \frac{\partial c_{\dot{e}_1\dot{e}_2|D_e}(u,w;\ \theta(\mathbf{x}_e))}{\partial x_e^1} h(u,w)\mathrm{d}u\mathrm{d}w.$$

If $c_{\dot{e}_1\dot{e}_2|D_e}(u,w;\ \theta(\mathbf{x}_e))$ is differentiable w.r.t. $\theta(\mathbf{x}_e)$, then we have

$$\frac{\partial c_{\dot{e}_1\dot{e}_2|D_e}(u,w;\ \theta(\mathbf{x}_e))}{\partial x_e^1} = \frac{\partial c_{\dot{e}_1\dot{e}_2|D_e}(u,w;\ \theta(\mathbf{x}_e))}{\partial \theta(\mathbf{x}_e)} \frac{\partial \theta(\mathbf{x}_e)}{\partial x_e^1}.$$

Hence, a sufficient condition for $\alpha(\mathbf{x}_e)$ to be differentiable is that $c_{\dot{e}_1\dot{e}_2|D_e}(u,w;\theta(\mathbf{x}_e))$ is differentiable w.r.t. $\theta(\mathbf{x}_e)$. Commonly used copulas, e.g., Gaussian copula, are all continuous and differentiable w.r.t. the involved parameters.

Remark 3 With $\mathbf{x}_e$ varying within a small region, we can closely approximate $c_{\dot{e}_1\dot{e}_2|D_e}(u,w;\theta(\mathbf{x}_e))$ using a single information set (with the information set containing sufficient basis functions). Since $c_{\dot{e}_1\dot{e}_2|D_e}(u,w;\theta(\mathbf{x}_e))$ is a continuous function of $\mathbf{x}_e$, it is easy to prove that each of the Lagrange multipliers in the minimally informative copula is a continuous function of $\mathbf{x}_e$.

References

Aas, K., Czado, C., Frigessi, A., Bakken, H.: Pair-copula constructions of multiple dependence. Insur. Math. Econ. **44**(2), 182–198 (2009)

Acar, E.F., Craiu, R.V., Yao, F.: Dependence calibration in conditional copulas: a nonparametric approach. Biometrics **67**(2), 445–453 (2011)

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

Barron, A.R., Sheu, C.H.: Approximation of density functions by sequences of exponential families. Ann. Stat. **19**(3), 1347–1369 (1991)

Bedford, T., Cooke, R.: Probability density decomposition for conditionally dependent random variables modeled by vines. Ann. Math. Artif. Intell. **32**(1–4), 245–268 (2001)

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

Bedford, T., Wilson, K.: On the construction of minimum information bivariate copula families. Ann. Inst. Stat. Math. **66**(4), 703–723 (2014)

Bedford, T., Daneshkhah, A., Wilson, K.J.: Approximate uncertainty modeling in risk analysis with vine copulas. Risk Anal. **36**(4), 792–815 (2016)

Borwein, J., Lewis, A., Nussbaum, R.: Entropy minimization, dad problems, and doubly stochastic kernels. J. Funct. Anal. **123**(2), 264–307 (1994)

Cottle, R.W., Dantzig, G.B.: Complementary pivot theory of mathematical programming. Linear Algebra Appl. **1**(1), 103–125 (1968)

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

Epanechnikov, V.A.: Non-parametric estimation of a multivariate probability density. Theory Probab. Appl. **14**(1), 153–158 (1969)

Fan, J.: Design-adaptive nonparametric regression. J. Am. Stat. Assoc. **87**(420), 998–1004 (1992)

Fan, Y., Patton, A.J.: Copulas in econometrics. Ann. Rev. Econ. **6**(1), 179–200 (2014)

Frahm, G., Junker, M., Schmidt, R.: Estimating the tail-dependence coefficient: properties and pitfalls. Insur. Math. Econ. **37**(1), 80–100 (2005)

Gijbels, I., Veraverbeke, N., Omelka, M.: Conditional copulas, association measures and their applications. Comput. Stat. Data Anal. **55**(5), 1919–1932 (2011)

Haff, I.H.: Parameter estimation for pair-copula constructions. Bernoulli **19**(2), 462–491 (2013)

Haff, I.H., Segers, J.: Nonparametric estimation of pair-copula constructions with the empirical pair-copula. Comput. Stat. Data Anal. **84**, 1–13 (2015)

Hao, Z., Singh, V.P.: Integrating entropy and copula theories for hydrologic modeling and analysis. Entropy **17**(4), 22–53 (2015)

Hartigan, J.A., Wong, M.A.: Algorithm as 136: a k-means clustering algorithm. J. R. Stat. Soc. Ser. C (Appl. Stat.) **28**(1), 100–108 (1979)

Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, vol. 2. Springer, New York (2009)

Jaynes, E.T.: Probability Theory: The Logic of Science. Cambridge University Press, Cambridge (2003)

Joe, H.: Families of m-variate Distributions with Given Margins and $m(m-1)/2$ Bivariate Dependence Parameters. In: Rüschendorf, L., Schweizer, B., Taylor, M.D. (eds.) Distributions with fixed marginals and related topics, vol. 28, pp. 120–141 (1996)

Joe, H.: Dependence Modeling with Copulas. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, London (2014)

Kauermann, G., Schellhase, C.: Flexible pair-copula estimation in d-vines using bivariate penalized splines. Stat. Comput. **24**(6), 1081–1100 (2014)

Köhler, M., Schindler, A., Sperlich, S.: A review and comparison of bandwidth selection methods for kernel regression. Int. Stat. Rev. **82**(2), 243–274 (2014)

Kurowicka, D.: Dependence Modeling: Vine Copula Handbook. World Scientific, Singapore (2011)

Liew, C.K.: Inequality constrained least-squares estimation. J. Am. Stat. Assoc. **71**(355), 746–751 (1976)

Lopez-Paz, D., Hernandez-Lobato, J.M., Ghahramani, Z.: Gaussian process vine copulas for multivariate dependence. In: Proceedings of the 30th International Conference on Machine Learning, vol. 28, pp. 10–18 (2013)

Nussbaum, R.: Iterated Nonlinear Maps and Hilbert's Projective Metric, II. American Mathematical Society: Memoirs of the American Mathematical Society, American Mathematical Society, Providence (1989)

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

Pircalabelu, E., Claeskens, G., Gijbels, I.: Copula directed acyclic graphs. Stat. Comput. (2015). doi:10.1007/s11222-015-9599-9

Rudemo, M.: Empirical choice of histograms and kernel density estimators. Scand. J. Stat. **9**(2), 65–78 (1982)

Sheather, S., Jones, C.: A reliable data-based bandwidth selection method for kernel density estimation. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **53**(3), 683–690 (1991)

Sklar, A.: Fonctions de répartition à n dimensions et leurs marges. Publications de l'Institut de Statistique de Université de Paris **8**, 229–231 (1959)

So, M.K., Yeung, C.Y.: Vine-copula garch model with dynamic conditional dependence. Comput. Stat. Data Anal. **76**, 655–671 (2014)

Soto, M., Ochoa, A., González-Fernández, Y., Milanés, Y., Álvarez, A., Carrera, D., Moreno, E.: Vine Estimation of Distribution Algorithms with Application to Molecular Docking. Springer, Berlin, Heidelberg (2012)

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

Valizadeh Haghi, H., Lotfifard, S.: Spatiotemporal modeling of wind generation for optimal energy storage sizing. IEEE Trans. Sustain. Energy **6**(1), 113–121 (2015)

Veraverbeke, N., Omelka, M., Gijbels, I.: Estimation of a conditional copula and association measures. Scand. J. Stat. **38**(4), 766–780 (2011)