International Statistical Review

International Statistical Review (2018), 86, 3, 488–511 doi:10.1111/insr.12263

Vine Copulas for Imputation of Monotone Non-response

Caren Hasler[1], Radu V. Craiu[2] and Louis-Paul Rivest[3]

[1]*Department of Computer and Mathematical Sciences, University of Toronto Scarborough, Toronto, ON, Canada*
[2]*Department of Statistical Sciences, University of Toronto, Toronto, ON, Canada*

[3]*Département de mathématiques et de statistique, Université Laval, Quebec City, QC, Canada*
E-mail: caren.hasler@utoronto.ca

Summary

Monotone patterns of non-response may occur in longitudinal studies. When the measured variables are dependent, it is beneficial to use their joint statistical model to impute the missing values. We propose to use vine copulas to factorise the density of the observed variables into a cascade of bivariate copulas that yield a flexible model of their joint distribution. The structure of the vine depends on the non-response pattern. We propose a method to select the model, to estimate the parameters of the bivariate copulas of the selected model and to impute using the constructed model. The imputed values are drawn from the conditional distribution of the missing values, given the observed data. We discuss the generalisation of our results to more global non-response patterns.

Key words: copula; D-vine; imputation; non-response.

1 Introduction

In multivariate data, a monotone non-response pattern means that it is possible to rearrange the variables so that, for each sample unit, the first ℓ variables are observed and the remaining are missing, where ℓ is unit-specific. In a longitudinal study, units that drop out of the study and never return lead to a typical example of monotone non-response. It is well documented that non-responses increase the variance of estimates and may induce estimation bias. Imputation is a commonly used technique to handle non-response, which consists of filling in the gaps with hypothetic values called imputed values.

Two main approaches are applied when imputing multivariate data: joint modelling (JM) and full conditional specification (FCS). A review and comparison of these two approaches is presented in van Buuren (2007). When using JM, a model for the joint distribution of the data is postulated and used to generate imputed values. In Schafer (1997), JM is based on a multivariate normal distribution, and the imputed values are simulated from their conditional distribution via a Markov chain Monte Carlo algorithm. Honaker *et al.* (2011) also assume a multivariate normal distribution and draw imputations with an expectation–maximisation with bootstrapping (EMB) algorithm, which is implemented in the R package Amelia II. The well-known families of joint distributions usually require that the marginal distributions belong to

the same family. To overcome this lack of flexibility, Käärik & Käärik (2009) and Di Lascio *et al.* (2015) use multivariate copulas selected among a limited set of parametric families to model the joint distribution. They draw imputed values from the conditional distribution of the missing values, conditioned by the observed values. R package CoImp (Di Lascio & Giannerini, 2014) implements the copula-based imputation of Di Lascio *et al.* (2015). Other authors (Ding & Song, 2016) propose an EM algorithm in Gaussian copula with missing data.

Joint modelling is sometimes criticised for its lack of flexibility. The available models for the joint distribution may fail to capture some features of the data and to impute different types of variables (continuous and categorical). FCS is a more flexible option because the multivariate distribution is modelled by a sequence of conditional models. A model for the conditional density of each variable, conditioned by the other available variables, is postulated. Imputed values are drawn by iterating over the conditional densities. With FCS, the joint model is specified via conditional models only. This allows us to postulate models for which no known joint distribution exits. FCS approach is known under different names such as the multivariate sequential regression approach of Raghunathan *et al.* (2001) implemented in the Imputation and Variance Estimation software (IVEware), the Multivariate Imputation by Chained Equations (MICE) of van Buuren & Groothuis-Oudshoorn (2011) implemented in R package mice, and the chained equation models of Harrell *et al.* (2016) implemented in R package Hmisc and that of Gelman & Hill (2011) in R package mi. Even though they are flexible, FCS approaches usually restrict the choice of families for the marginals, such as normality for continuous variables.

The paper enlarges the copula families considered by Käärik & Käärik (2009) and Di Lascio *et al.* (2015) by using vine copulas. The proposed JM strategy is inspired by the work of Aas *et al.* (2009) and can be applied to impute continuous multivariate data that are missing completely at random (MCAR). It flexibly builds a joint model by a factorisation of the joint density into a cascade of bivariate copulas. Bedford & Cooke (2002) introduced graphical models for the dependency between variables called vines. Throughout the paper, we work with D-vines, a particular class of vines that facilitates pair-copula factorisations of the joint density. The method consists of four steps: (i) specification of the structure of the vine using the non-response pattern and the observed dependencies; (ii) identification of the pair-copula families of the vine through a sequential procedure; (iii) pseudo maximum likelihood estimation of the pair-copulas parameters; and (iv) imputation of missing values using their conditional distributions, given the observed data. The copula component of the method adds a great deal of flexibility, as it does not constrain the marginals to belong to preset families, and captures some complex features of the data, for example, tail dependence, that current JM and FCS proposals fail to account for. Similarly to FCS, our method allows to consider models for which no known joint distribution exists. The simulation studies show that our proposed method performs significantly better than existing methods in different situations.

In survey sampling, non-response is dealt with either by modelling the response process or by imputing the missing values. Imputation can be non-parametric, such as NN imputation, or relies on a model as does fractional imputation. When the aim is to estimate bivariate parameters of interest, such as a correlation coefficient, imputing the variables separately may yield severely biased estimators. An imputation method that preserves the relationship between variables is a more appropriate solution. See Chauvet & Haziza (2012) for a recent discussion. This work proposes a semi-parametric imputation method that preserves the relationship between variables and adopts the second aforementioned approach; it builds an imputation model using D-vines.

The paper is organised as follows: Section 2 introduces the framework and notations; Section 3 gives an overview of D-vines; Section 4 presents our estimation method; Sections 5 and 6 introduce our procedure of model selection; Section 7 compares the performance of our

method with that of some of the existing methods via simulations on real and hypothetical data; Section 8 discusses the generalisation of our method to other non-response mechanisms and patterns; and Section 9 closes the paper with concluding remarks. The Appendices are included in the Supporting Information.

2 Framework and Notations

We consider a finite sample $s = \{1, 2, \ldots, k, \ldots, n\}$ of size n. Let $X_1, X_2, \ldots, X_d$ be d variables of interest that may be measured for each unit, and let x_{ki} represent the value of the i-th variable taken by unit k. The purpose of the paper is to address the situation in which some of the units have missing variable values. We assume a monotone non-response pattern, which is a specific case of item non-response implying that it is possible to label the variables such that, for each unit k, if an observation x_{ki} is missing, then all the observations $\{x_{kj} : j > i\}$ are also missing. In what follows, we assume such a labelling of the variables and that any unit will have at least one variable observed. The sample is partitioned into d subsamples $s_1, s_2, \ldots, s_d$ where s_ℓ is the subsample of those units k with x_{ki} observed for $i \leq \ell$ and missing for $i > \ell$, $\ell = 1, 2, \ldots, d$. We denote by n_ℓ the size of s_ℓ. We consider without loss of generality that the units are rearranged in increasing order of the number of observed variables (Figure 1).

We adopt a superpopulation (or model-assisted) approach in which we assume that the vectors $(x_{k1}, x_{k2}, \ldots, x_{kd})$ are independent and identically distributed (i.i.d.) outcomes of a vector of random variables $(X_1, X_2, \ldots, X_d)$ with joint density function f. In what follows, f and F are used to denote joint, conditional and marginal distribution functions. The arguments of these functions indicate the variables we refer to, for example, $F(x_2|x_1)$ is used for $F_{X_2|X_1}(x_2|x_1)$, $F(x_1, x_2)$ for $F_{X_1,X_2}(x_1, x_2)$ and so on. We use an index to indicate the variable we refer to whenever it is not obvious, for example, we may use F_2 for the distribution function of X_2. Finally, we assume that the missing data are MCAR and that the missing data process is unrelated to the joint distribution of the variables (see Rubin, 1976 for detailed definitions).

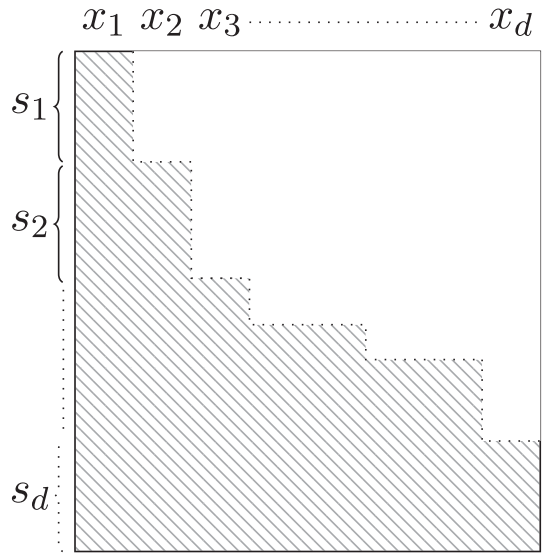

Figure 1. *Monotone non-response pattern. The hatched area represents observed values and the blank area missing values.*

3 D-vine

Following Bedford & Cooke (2002), a *d-dimensional D-vine* is a sequence of $d-1$ linked trees T_1, T_2, T_{d-1} such that

1 Tree T_1 is a path-like tree with nodes $1, 2, \ldots, d$ and $d-1$ edges: each node is connected by at least 1 but no more than two edges,
2 Each edge in tree T_j is a node in tree T_{j+1},
3 Two nodes in tree T_{j+1} are connected by an edge if and only if the corresponding edges in tree T_j share a node in tree T_j.

3.1 Model Construction

Figure 2 shows two four-dimensional D-vines. The sequence of trees of a D-vine is fully determined by the order of the first tree. We suppose for now that the variables are ordered by decreasing number of observed values in the first tree (Figure 2, left). We will relax this assumption in Section 6. The paper assumes that the joint density $f(x_1, x_2, \ldots, x_d)$ may be written in terms of a d-dimensional D-vine (see Aas et al., 2009, equation (8)) as

$$f(x_1, x_2, \ldots, x_d) = \prod_{k=1}^{d} f(x_k) \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} c_{i,i+j|i+1,\ldots,i+j-1} \left\{ F(x_i|x_{i+1}, \ldots, x_{i+j-1}), F(x_{i+j}|x_{i+1}, \ldots, x_{i+j-1}) \right\}, \tag{1}$$

where $c_{i,i+j|i+1,\ldots,i+j-1}$ is a bivariate copula density belonging to some parametric family for the dependency between the transformed variables $F(x_i|x_{i+1}, \ldots, x_{i+j-1})$ and $F(x_{i+j}|x_{i+1}, \ldots, x_{i+j-1})$. To each edge of the D-vine is attached a bivariate copula density; such decomposition of the joint distribution in terms of pair copula is called a pair-copula construction. Equation (1) makes the so called simplifying assumption as $c_{i,i+j|i+1,\ldots,i+j-1}$ represents a single bivariate copula density that does not depend on the conditioning variables $(x_{i+1}, \ldots, x_{i+j-1})$. More general models could be constructed by letting $c_{i,i+j|i+1,\ldots,i+j-1}$ vary with the conditioning variables. See Joe (2014, p.118) for a detailed discussion on the simplifying assumption. Note that a joint density can always be written in terms of a D-vine; the only assumption in Eqn (1) is the simplifying assumption.

In (1), the conditional distribution $F(x_{i+j}|x_{i+1}, \ldots, x_{i+j-1})$ can be deduced from the bivariate copulas of the D-vine. Indeed, $F(x_{i+j}|x_{i+1}, \ldots, x_{i+j-1})$ can be constructed using the D-vine for the joint density of the j variables, $(X_{i+1}, \ldots, X_{i+j})$. This is a sub-vine of the general D-vine for the d variables involving $j(j-1)/2$ nodes. We illustrate the construction of $F(x_{i+j}|x_{i+1}, \ldots, x_{i+j-1})$ when $j = 4$. The conditional density of X_{i+4} given $(X_{i+1}, X_{i+2}, X_{i+3})$ is the ratio of two D-vine densities, given by (1). It is equal to

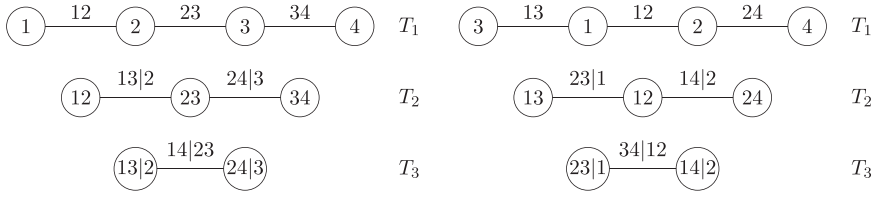

Figure 2. *Four-dimensional D-vines.*

$$
\begin{aligned}
&f(x_{i+4}|x_{i+1}, x_{i+2}, x_{i+3}) \\
&\quad = \frac{f(x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4})}{f(x_{i+1}, x_{i+2}, x_{i+3})} \\
&\quad = f(x_{i+4})c_{i+3,i+4}\{F(x_{i+3}), F(x_{i+4})\}c_{i+2,i+4|i+3}\{F(x_{i+2}|x_{i+3}), F(x_{i+4}|x_{i+3})\} \\
&\qquad c_{i+1,i+4|i+2,i+3}\{F(x_{i+1}|x_{i+2}, x_{i+3}), F(x_{i+4}|x_{i+2}, x_{i+3})\} \\
&\quad = \frac{f(x_{i+2}, x_{i+3}, x_{i+4})}{f(x_{i+2}, x_{i+3})} c_{i+1,i+4|i+2,i+3}\{F(x_{i+1}|x_{i+2}, x_{i+3}), F(x_{i+4}|x_{i+2}, x_{i+3})\} \\
&\quad = f(x_{i+4}|x_{i+2}, x_{i+3})c_{i+1,i+4|i+2,i+3}\{F(x_{i+1}|x_{i+2}, x_{i+3}), F(x_{i+4}|x_{i+2}, x_{i+3})\}.
\end{aligned} \tag{2}
$$

Integrating the two sides of (2) allows to express the conditional distribution of X_{i+4} in terms of $C_{i+1,i+4|i+2,i+3}$, the copula distribution function, as

$$
F(x_{i+4}|x_{i+1}, x_{i+2}, x_{i+3}) = \frac{\partial C_{i+1,i+4|i+2,i+3}\{F(x_{i+1}|x_{i+2}, x_{i+3}), F(x_{i+4}|x_{i+2}, x_{i+3})\}}{\partial F(x_{i+1}|x_{i+2}, x_{i+3})}.
$$

This holds for arbitrary values of j giving

$$
\begin{aligned}
&F(x_{i+j}|x_{i+1}, \ldots, x_{i+j-1}) \\
&\quad = \frac{\partial C_{i+1,i+j|i+2,\ldots,i+j-1}\left\{F(x_{i+1}|x_{i+2}, \ldots, x_{i+j-1}), F(x_{i+j}|x_{i+2}, \ldots, x_{i+j-1})\right\}}{\partial F(x_{i+1}|x_{i+2}, \ldots, x_{i+j-1})}.
\end{aligned} \tag{3}
$$

A similar construction for the conditional distribution $F(x_i|x_{i+1}, \ldots, x_{i+j-1})$ gives

$$
\begin{aligned}
&F(x_i|x_{i+1}, \ldots, x_{i+j-1}) \\
&\quad = \frac{\partial C_{i,i+j-1|i+1,\ldots,i+j-2}\left\{F(x_i|x_{i+1}, \ldots, x_{i+j-2}), F(x_{i+j-1}|x_{i+1}, \ldots, x_{i+j-2})\right\}}{\partial F(x_{i+j-1}|x_{i+1}, \ldots, x_{i+j-2})}.
\end{aligned} \tag{4}
$$

These are special cases of a result first noticed by Joe (1996). Equations (3) and (4) allow the sequential evaluation of the conditional distributions appearing in (1).

3.2 Simulation from a D-vine

We assume that only the first ℓ variables $(x_1, \ldots, x_\ell)$ are observed for a unit. To impute the missing values, we would like to simulate from the conditional distribution of $(X_{\ell+1}, \ldots, X_d)$ given that $X_1 = x_1, \ldots, X_\ell = x_\ell$. We first consider the problem of simulating $X_{\ell+1}$.

This is performed using (3), with $i = 0$ and $j = \ell + 1$. In that equation, $F(x_1|x_2, \ldots, x_\ell)$ is known as it depends on variables that have been observed. It can can be evaluated using (4), with $i = 1$ and $j = \ell$. A random variable $v_{\ell+1}$ distributed as $F(X_{\ell+1}|x_2, \ldots, x_\ell)$ is first obtained. If W is uniformly distributed on (0, 1), then $v_{\ell+1}$ is obtained by solving

$$
W = \frac{\partial C_{1,\ell+1|2,\ldots,\ell}\{F(x_1|x_2, \ldots, x_\ell), v_{\ell+1}\}}{\partial F(x_1|x_2, \ldots, x_\ell)}.
$$

If $\ell = 1$, one simply takes $x_2^* = F_2^{-1}(v_{\ell+1})$, and the algorithm stops. If $\ell > 1$, one obtains a random variable $v_{\ell+1,1}$ distributed as $F(X_{\ell+1}|x_3, \ldots, x_\ell)$ by solving

$$v_{\ell+1} = \frac{\partial C_{2,\ell+1|2,\ldots,\ell}\left\{F(x_2|x_3,\ldots,x_\ell), v_{\ell+1,1}\right\}}{\partial F(x_2|x_3,\ldots,x_\ell)}.$$

If $\ell = 2$, one simply takes $x_3^* = F_3^{-1}(v_{\ell+1,1})$, and the algorithm stops. Note that this step requires the evaluation of the numerical value of $F(x_2|x_3,\ldots,x_\ell)$. It can can be evaluated using (4), with $i = 2$ and $j = \ell - 1$. For an arbitrary value of ℓ, this algorithm has to be iterated $\ell - 1$ times to obtain $x_{\ell+1}^*$, a simulated value for variable $X_{\ell+1}$. To simulate $X_{\ell+2}$, one repeats the algorithm starting at the values $X_1 = x_1, \ldots, X_\ell = x_\ell, X_{\ell+1} = x_{\ell+1}^*$. These calculations are relatively technical, and they can be organised efficiently; see algorithm 2 of Aas *et al.* (2009). We used their algorithm to simulate from our vine model.

3.3 Example: Four-dimensional Case

We first show how (3) and (4) allow the sequential evaluation of the conditional distributions appearing in (1) when $d = 4$. In this case, the density becomes

$$\begin{aligned} f(x_1,x_2,x_3,x_4) = {} & f(x_1)f(x_2)f(x_3)f(x_4) \\ & c_{12}\{F(x_1),F(x_2)\}\, c_{23}\{F(x_2),F(x_3)\}\, c_{34}\{F(x_3),F(x_4)\} \\ & c_{13|2}\{F(x_1|x_2),F(x_3|x_2)\}\, c_{24|3}\{F(x_2|x_3),F(x_4|x_3)\} \\ & c_{14|23}\{F(x_1|x_2,x_3),F(x_4|x_2,x_3)\}. \end{aligned}$$

The first two conditional distributions $F(x_1|x_2)$ and $F(x_3|x_2)$ are obtained using (4) and (3), with $i = 1$ and $j = 2$. We obtain

$$\begin{aligned} F(x_1|x_2) &= \frac{\partial C_{12}\{F(x_1),F(x_2)\}}{\partial F(x_2)}, \\ F(x_3|x_2) &= \frac{\partial C_{23}\{F(x_2),F(x_3)\}}{\partial F(x_2)}. \end{aligned}$$

The next two conditional distributions $F(x_2|x_3)$ and $F(x_4|x_3)$ are obtained using (4) and (3), with $i = 2$ and $j = 2$. We obtain

$$\begin{aligned} F(x_2|x_3) &= \frac{\partial C_{23}\{F(x_2),F(x_3)\}}{\partial F(x_3)}, \\ F(x_4|x_3) &= \frac{\partial C_{34}\{F(x_3),F(x_4)\}}{\partial F(x_3)}. \end{aligned}$$

The last two conditional distributions $F(x_1|x_2,x_3)$ and $F(x_4|x_2,x_3)$ are obtained using (4) and (3), with $i = 1$ and $j = 3$, and the numerical values of the conditional distributions obtained previously. We obtain

$$\begin{aligned} F(x_1|x_2,x_3) &= \frac{\partial C_{13|2}\{F(x_1|x_2),F(x_3|x_2)\}}{\partial F(x_3|x_2)}, \\ F(x_4|x_2,x_3) &= \frac{\partial C_{24|3}\{F(x_2|x_3),F(x_4|x_3)\}}{\partial F(x_2|x_3)}. \end{aligned}$$

Now, we show how to simulate from the conditional distribution of the missing values given the observed values. With monotone non-response, there are three conditional distributions that we want to simulate from: (X_2, X_3, X_4) given $X_1 = x_1$, (X_3, X_4) given $X_1 = x_1$ and $X_2 = x_2$

and X_4 given $X_1 = x_1, X_2 = x_2$ and $X_3 = x_3$. We first consider the problem of simulating X_2 given $X_1 = x_1$, X_3 given $X_2 = x_2$ and $X_1 = x_1$ and X_4 given $X_3 = x_3$, $X_2 = x_2$ and $X_1 = x_1$.

To simulate X_2 given $X_1 = x_1$, we consider (3) with $i = 0$ and $j = 2$:

$$F(x_2|x_1) = \frac{\partial C_{12}\{F(x_1), F(x_2)\}}{\partial F(x_1)}.$$

In that equation, $F(x_1)$ is known as it depends on a variable that has been observed. A random variable v_2 distributed as $F(X_2|x_1)$ is first obtained. If W is uniformly distributed on $(0, 1)$, then v_2 is obtained by solving

$$W = \frac{\partial C_{12}\{F(x_1), v_2\}}{\partial F(x_1)}.$$

The simulated value of X_2 given $X_1 = x_1$ is $x_2^* = F_2^{-1}(v_2)$. To simulate X_3 given $X_2 = x_2$ and $X_1 = x_1$, we consider (3), with $i = 0$ and $j = 3$. We obtain

$$F(x_3|x_1, x_2) = \frac{\partial C_{13|2}\{F(x_1|x_2), F(x_3|x_2)\}}{\partial F(x_1|x_2)}.$$

In that equation, $F(x_1|x_2)$ is known, as it depends on variables that have been observed. It can be evaluated as shown previously. A random variable v_3 distributed as $F(X_3|x_2)$ is first obtained. If W is uniformly distributed on $(0, 1)$, then v_3 is obtained by solving

$$W = \frac{\partial C_{13|2}\{F(x_1|x_2), v_3\}}{\partial F(x_1|x_2)}.$$

Then, one obtains a random variable $v_{3,1}$ distributed as $F(X_3)$ by solving

$$v_3 = \frac{\partial C_{23}\{F(x_2), v_{3,1}\}}{\partial F(x_2)}.$$

The simulated value of X_3 given $X_2 = x_2$ and $X_1 = x_1$ is $x_3^* = F_3^{-1}(v_{3,1})$. To simulate from X_4 given $X_3 = x_3$, $X_2 = x_2$ and $X_1 = x_1$, one applies a similar construction. Three steps will be required.

We show now how we can simulate from (X_2, X_3, X_4) given $X_1 = x_1$ in three steps: first, one simulates x_2^* from X_2 given $X_1 = x_1$; second, one simulates x_3^* from X_3 given $X_1 = x_1$ and $X_2 = x_2^*$; and lastly, one simulates x_4^* from X_4 given $X_1 = x_1, X_2 = x_2^*$ and $X_3 = x_3^*$. Applying the same construction, we simulate from (X_3, X_4) given $X_1 = x_1$ and $X_2 = x_2$ in two steps and from X_4 given $X_3 = x_3$, $X_2 = x_2$ and $X_1 = x_1$ in one step.

4 Estimation

This section addresses the D-vine parameters estimation for a given D-vine structure and given pair-copula families. We apply a margin-free method and maximise an observed-data pseudo log-likelihood function. We explain how this function can be maximised using a method to evaluate the pseudo log-likelihood function in the complete-data case. Section 4.1 presents first our proposed estimation method for four variables, and in Section 4.2, we extend the ideas to the general case of d variables.

4.1 Observed-data Pseudo-likelihood for Four Variables

With complete response, the contribution of a unit to the likelihood function is $f(x_1, x_2, x_3, x_4)$. As assumed earlier, the missing data are MCAR. In this case, ignoring the process that causes missing data yields proper inference (see Rubin, 1976). When we ignore the process that causes missing data, we use the observed-data likelihood where the contribution to the likelihood of a unit is

$$f(x_{\text{obs}}) = \int f(x_{\text{obs}}, x_{\text{mis}}) dx_{\text{mis}}, \tag{5}$$

where x_{obs} and x_{mis} are the observed and missing variables of this unit, respectively. In the case of monotone non-response, the contribution to the likelihood of a unit in s_3 is

$$f(x_1, x_2, x_3) = \int f(x_1, x_2, x_3, x_4) dx_4.$$

By integrating the four-dimensional D-vine density in (1), we obtain

$$f(x_1, x_2, x_3) = f(x_1) f(x_2) f(x_3) c_{12}\{F(x_1), F(x_2)\} c_{23}\{F(x_2), F(x_3)\} c_{13|2}\{F(x_1|x_2), F(x_3|x_2)\}.$$

That is, the contribution to the likelihood of a unit in s_3 is $f(x_1, x_2, x_3)$, which is a three-dimensional D-vine density. This three-dimensional D-vine is a sub-vine of the initial four-dimensional D-vine considered. We apply recursively the same construction and obtain the contribution to the likelihood of a unit in s_2:

$$f(x_1, x_2) = \int f(x_1, x_2, x_3) dx_3 = f(x_1) f(x_2) c_{12}\{F(x_1), F(x_2)\},$$

which is the density of a two-dimensional sub-D-vine of the initial four-dimensional D-vine considered. Hence, we can associate a sub-D-vine to each subsample s_ℓ. Figure 3 shows the three sub-D-vines and the associated subsamples.

We use the margin-free semi-parametric estimation method of Genest *et al.* (1995) to estimate the copula parameters of the D-vine. The idea is to estimate the margins using empirical distribution functions and the copula parameters via a parametric family. We consider the contributions of the units in the different subsamples developed previously, and we select the copula parameters that maximise the following observed-data pseudo log-likelihood function:

$$\begin{aligned}\log \tilde{L}(\boldsymbol{\theta}|x_1, x_2, x_3, x_4) = &\sum_{k \in s_2} \log c_{12}^k + \sum_{k \in s_3} \left(\log c_{12}^k + \log c_{23}^k + \log c_{13|2}^k\right) \\ &+ \sum_{k \in s_4} \left(\log c_{12}^k + \log c_{23}^k + \log c_{34}^k + \log c_{13|2}^k + \log c_{24|3}^k + \log c_{14|23}^k\right),\end{aligned} \tag{6}$$

where $c^k_{i,i+j|i+1,\ldots,i+j-1}$ is a shortcut for

$$c_{i,i+j|i+1,\ldots,i+j-1}\left[F(y_{k,i}|y_{k,i+1}, \ldots, y_{k,i+j-1}), F(y_{k,i+j}|y_{k,i+1}, \ldots, y_{k,i+j-1})\right].$$

Here, $y_{k,i} = \widehat{F}_i(x_{k,i})$ and $\widehat{F}_i$ is $\frac{n_i}{n_i+1}$ times the empirical marginal distribution of the i-th variable, and n_i is the number of observed values of this variable. Note that when the missing data are MCAR, $\widehat{F}_i$ is a consistent estimator of the marginal distribution of the

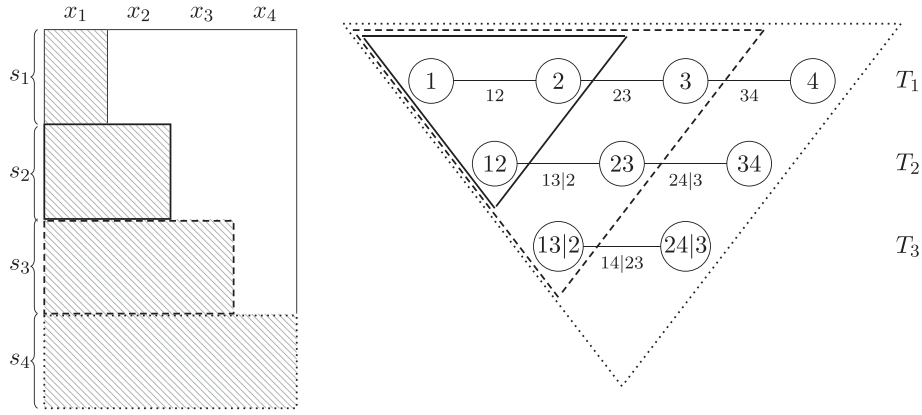

Figure 3. *Monotone non-response pattern for four variables and four-dimensional D-vine with sub-D-vine associated with s_2 (solid), s_3 (dashed) and s_4 (dotted).*

i-th variable. In addition $F(y_{k,i}|y_{k,i+1},\ldots,y_{k,i+j-1})$ and $F(y_{k,i+j}|y_{k,i+1},\ldots,y_{k,i+j-1})$ are evaluated recursively, using (3) and (4), as functions of $y_{k,i}$ and of the bivariate copulas for the sub-D-vine for variables $(i,\ldots,i+j)$. The observed-data pseudo log-likelihood function in (6) can be rewritten:

$$\begin{aligned}&\log\tilde{L}(\boldsymbol{\theta}|x_1,x_2,x_3,x_4)\\&\quad=\log L_2(\boldsymbol{\theta}_{11}|x_1,x_2)+\log L_3(\boldsymbol{\theta}_{11},\boldsymbol{\theta}_{12},\boldsymbol{\theta}_{21}|x_1,x_2,x_3)+\log L_4(\boldsymbol{\theta}|x_1,x_2,x_3,x_4),\end{aligned}\tag{7}$$

where

$$\begin{aligned}\log L_2(\boldsymbol{\theta}_{11}|x_1,x_2)&=\sum_{k\in s_2}\log c^k_{12},\\\log L_3(\boldsymbol{\theta}_{11},\boldsymbol{\theta}_{12},\boldsymbol{\theta}_{21}|x_1,x_2,x_3)&=\sum_{k\in s_3}\left(\log c^k_{12}+\log c^k_{23}+\log c^k_{13|2}\right),\\\log L_4(\boldsymbol{\theta}|x_1,x_2,x_3,x_4)&=\sum_{k\in s_4}\Big(\log c^k_{12}+\log c^k_{23}+\log c^k_{34}+\log c^k_{13|2}+\log c^k_{24|3}\\&\qquad+\log c^k_{14|23}\Big),\end{aligned}$$

where $\boldsymbol{\theta}_{ji}$ is the parameter of the copula density $c_{i,i+j|i+1,\ldots,i+j-1}$ and $\boldsymbol{\theta}=(\boldsymbol{\theta}_{11},\boldsymbol{\theta}_{12},\boldsymbol{\theta}_{13},\boldsymbol{\theta}_{21},\boldsymbol{\theta}_{22},\boldsymbol{\theta}_{31})$ is the parameter vector of the four-dimensional D-vine. The function $\log L_2(\boldsymbol{\theta}_{11}|x_1,x_2)$ is the complete-data pseudo log-likelihood function of the two-dimensional sub-D-vine on x_1 and x_2 given the observations in s_2. Similarly, $\log L_3(\boldsymbol{\theta}_{11},\boldsymbol{\theta}_{12},\boldsymbol{\theta}_{21}|x_1,x_2,x_3)$ is the complete-data pseudo log-likelihood function of the three-dimensional sub-D-vine on x_1, x_2 and x_3 given the observations in s_3, and $\log L_4(\boldsymbol{\theta}|x_1,x_2,x_3,x_4)$ is the complete-data pseudo log-likelihood function of the four-dimensional D-vine on x_1, x_2, x_3 and x_4 given the observations in s_4. These three functions are complete-data pseudo log-likelihood functions restricted to subsamples of the initial sample.

Aas *et al.* (2009) propose an algorithm (algorithm 4, p. 188) to evaluate the complete-data pseudo log-likelihood function of a D-vine when there are no missing data. We apply their algorithm to evaluate L_2, L_3 and L_4 in (7). We obtain an estimate of the vector parameter $\boldsymbol{\theta}$ by numerical maximisation of the observed-data pseudo log-likelihood function $\log\tilde{L}$. A

procedure to set the starting value of the parameters for the numerical maximisation is highlighted in Section 5.

We should emphasise that the observed-data pseudo log-likelihood function underlying the approach described previously is the full observed data one, and therefore, we do not expect any loss in information or statistical efficiency. The vine-induced factorisation of the pseudo log-likelihood makes the EM algorithm, commonly used in missing data problem, irrelevant for pair-copula models.

4.2 Observed-data Pseudo-likelihood for d Variables

In the d-dimensional case, the sample is partitioned into d subsamples s_ℓ, $\ell = 1, \ldots, d$, where s_ℓ contains those units having the first ℓ variables observed and the last $d - \ell$ variables missing. Repeating the same construction as for four variables, we obtain the following observed-data pseudo log-likelihood function:

$$\begin{aligned} \log \tilde{L}(\boldsymbol{\theta}|x_1, x_2, \ldots, x_d) &= \sum_{\ell=2}^{d} \log L_\ell(\boldsymbol{\theta}_{11}, \ldots, \boldsymbol{\theta}_{\ell-1,1}|x_1, x_2, \ldots, x_\ell) \\ &= \sum_{\ell=2}^{d} \sum_{k \in s_\ell} \sum_{j=1}^{\ell-1} \sum_{i=1}^{\ell-j} \log c^k_{i,i+j|i+1,\ldots,i+j-1}, \end{aligned} \tag{8}$$

where

$$\log L_\ell(\boldsymbol{\theta}_{11}, \ldots, \boldsymbol{\theta}_{\ell-1,1}|x_1, x_2, \ldots, x_\ell) = \sum_{k \in s_\ell} \sum_{j=1}^{\ell-1} \sum_{i=1}^{\ell-j} \log c^k_{i,i+j|i+1,\ldots,i+j-1}.$$

The contribution of subsample s_ℓ to the likelihood is $\log L_\ell$, which is the complete-data pseudo log-likelihood function of a ℓ-dimensional D-vine for $x_1, x_2, \ldots, x_\ell$ given the observations in s_ℓ. We obtain an estimate of the vector parameter $\boldsymbol{\theta}$ by numerical maximisation of the pseudo log-likelihood function adapted for monotone non-response $\log \tilde{L}$ in (8). The algorithm of Aas *et al.* (2009) is applied to evaluate $\log L_\ell$, $\ell = 2, \ldots, d$.

5 Selection of the Bivariate Copula Families

This section addresses the selection of the bivariate copula families given a D-vine tree structure. We apply a sequential procedure that fully uses the observed data. To select the bivariate copula families of a given D-vine, we propose a simple modification for monotone non-response of the sequential procedure introduced in Section 6 of Aas *et al.* (2009). In Section 4, we have considered that the inputs of a pair-copula density $c_{i,i+j|i+1,\ldots,i+j-1}$ are $F(y_{k,i}|y_{k,i+1}, \ldots, y_{k,i+j-1})$ and $F(y_{k,i+j}|y_{k,i+1}, \ldots, y_{k,i+j-1})$. In what follows, we will refer to these inputs to as pseudo-observations. The general idea of our method is the following:

1B. Use a measure of quality such as Akaike Information Criterion to separately select a copula family for each pair copula $c_{i,i+1}$ of the tree 1; use the largest possible set of pseudo-observations, that is, use $y_{k,i}$, $y_{k,i+1}$ for $k \in s_{i+1} \cup \ldots \cup s_d$.

1C. Estimate the copula parameter of each pair copula $c_{i,i+1}$ separately by maximisation of the pseudo log-likelihood associated $\sum \log c^k_{i,i+1}$; use the largest possible set of pseudo-observations as in the previous step.

2A. Construct the pseudo-observations $F(y_{k,i}|y_{k,i+1})$, $F(y_{k,i+2}|y_{k,i+1})$ associated with the tree 2 using (3) and (4), the pseudo-observations of the previous tree and the pair-copula families and parameters selected for the the previous tree.

2B. Use a measure of quality such as Akaike Information Criterion to separately select a copula family for each pair copula $c_{i,i+2|i+1}$ of tree 2; use the largest possible set of pseudo-observations, that is use $F(y_{k,i}|y_{k,i+1})$, $F(y_{k,i+2}|y_{k,i+1})$ for $k \in s_{i+2} \cup \ldots \cup s_d$.

2C. Estimate the copula parameter of each pair copula $c_{i,i+2|i+1}$ separately by maximisation of the pseudo log-likelihood associated $\sum \log c^k_{i,i+2|i+1}$; use the largest possible set of pseudo-observations as in the previous step.

3. Iterate for trees 3 to $\ell - 1$.

The purpose of this sequential procedure is to select the bivariate copula families, but its outputs also include estimated values of the parameters. These values may be used as starting values of the parameters in the numerical maximisation of the observed-data pseudo log-likelihood of Section 4.

6 Tree Structure Selection

This section addresses the selection of a D-vine tree structure. Until now, we have supposed that the variables were ordered by increasing number of observed values in the first tree (see Figure 2, left, for the four-dimensional case). We show in this section that our estimation method can be applied for other D-vine tree structures and present a procedure to select such a structure for four and d variables in Sections 6.1 and 6.2, respectively.

6.1 Selection of a Tree Structure for Four Variables

An important condition associated with the estimation method presented in Section 4 is that the subsamples can be associated with sub-D-vines of the initial D-vine considered. Figure 4 shows two examples of D-vine tree structures having this particularity.

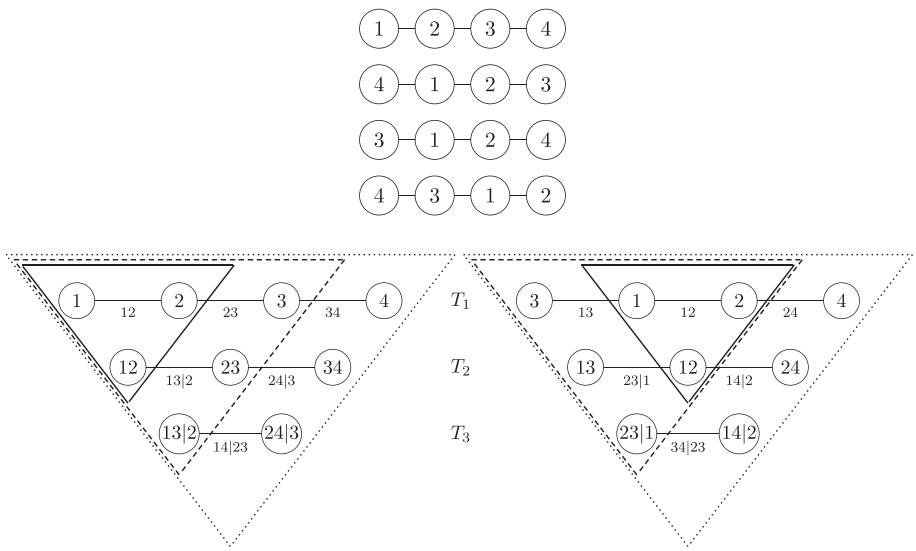

Figure 4. *Two four-dimensional D-vine tree structures for which our estimation method can be applied. Sub-D-vines associated with s_2 (solid), s_3 (dashed) and s_4 (dotted) are shown.*

In the case of four variables, there are four such D-vine tree structures determined by the following orders of the variables in the first tree.

Note that two symmetric trees define the same decomposition, which is the reason why we listed here four rather than eight tree structures. The idea is to select the tree structure that maximises the dependency accounted for in the first tree. The dependency is quantified via the empirical Kendall's tau. Let τ_{ij} be the empirical Kendall's τ of x_i and x_j, $i < j$. When estimating the Kendall's τ of data with non-response, we may want to use as many observations as possible for efficiency purposes. In this case, we use those units that allow to compute the pairwise Kendall's tau. Because of the non-response pattern, the different Kendall's τs may be estimated using samples of different sizes. For instance, with monotone non-response, τ_{12} is estimated via a sample of size $n_2 + n_3 + n_4$ and τ_{14} via a sample of size n_4. As a result, τ_{12} has less variability than τ_{14}, and we are more confident about a strong dependency between x_1 and x_2 when τ_{12} is high than we are about a strong dependency between x_1 and x_4 when τ_{14} is high. Therefore, extra care has to be taken when comparing the different Kendall's τ's. We describe below the proposed procedure to select a D-vine tree structure that maximises the dependency accounted for in the first tree. Our proposed procedure bypasses the problem of different variability in the τs by comparing pairs of τs that are estimated based on samples of same size.

1 Start with variable x_1 and x_2 adjacent, that is, $T_1 = (1, 2)$.
2 Compute τ_{13} and τ_{23} using $s_3 \cup s_4$. If $\tau_{13} > \tau_{23}$, set x_3 to the left of x_1, that is, $T_1 = (3, 1, 2)$. Otherwise, set x_3 to the right of x_2, that is, $T_1 = (1, 2, 3)$.
3 If $T_1 = (3, 1, 2)$, compute τ_{34} and τ_{24} using s_4.

   If $\tau_{34} > \tau_{24}$, set x_4 to the left of x_3, that is, $T_1 = (4, 3, 1, 2)$.
   Otherwise, set x_4 to the right of x_2, that is, $T_1 = (3, 1, 2, 4)$.

   Else if $T_1 = (1, 2, 3)$, compute τ_{14} and τ_{34} using s_4.

   If $\tau_{14} > \tau_{34}$, set x_4 to the left of x_1, that is, $T_1 = (4, 1, 2, 3)$.
   Otherwise, set x_4 to the right of x_3, that is, $T_1 = (1, 2, 3, 4)$.

Algorithm 1 Selection of a D-vine tree structure. Selects the order of the variables in the first tree T_1 of the D-vine decomposition. The Algorithm returns a vector $v(d, j)$, $j = 1, \ldots, d$ which gives the order of the variables in the first tree.

```
Set v(2,1) = 1, v(2,2) = 2
for i ← 3,...,d do
    Compute τ_{v(i−1,1),i} and τ_{v(i−1,i−1),i} using ⋃_{k=i}^{d} s_k;
    if τ_{v(i−1,1),i} > τ_{v(i−1,i−1),i} then
        v(i,1) = i;
        for j ← 2,...,i do
            v(i,j) = v(i−1,j−1);
        end for
    else
        v(i,i) = i;
        for j ← 1,...,i−1 do
            v(i,j) = v(i−1,j);
        end for
    end if
end for
```

6.2 Selection of a D-vine Tree Structure for d Variables

We generalise the procedure applied for four variables. The estimation method of Section 4 exploits the maximal amount of data possible if variables x_1 to x_ℓ define a ℓ-dimensional sub-D-vine, for $\ell = 2, \ldots, d$. There are 2^{d-2} such d-dimensional D-vine tree structures (remember that two symmetric tree structures determine the same decomposition). Algorithm 1 presents the proposed procedure to select one of those.

The algorithm compares pairs of τs estimated via the same sample size to select a D-vine tree structure that maximises the dependency accounted for in the first tree. This algorithm constructs the first tree of the D-vine sequentially. It does not require to list all possible D-vine tree structures. This may be computationally interesting in high dimensions because the number of possible D-vine tree structures grows exponentially with the dimension.

7 Simulation Studies

7.1 Real Data

We consider the Labour Force Survey Five-Quarter Longitudinal Dataset January 2014–March 2015 (Office for National Statistics. Social Survey Division and Northern Ireland Statistics and Research Agency. Central Survey Unit, 2015). The data are distributed by the UK Data Archive at the University of Essex. We consider the total actual hours in main and second job (hereafter total actual hours) for five consecutive quarters as variables of interest. We denote these variables by x_ℓ, $\ell = 1, \ldots, 5$, where the index refers to the quarter. Only surveyed individuals with available and non-null total actual hours for the five consecutive quarters are considered. This results in a sample of $n = 1\,738$ individuals aged between 16 and 69. In this sample, the total actual hours ranges from 1 to 97. The value 97 indicates a total actual hours greater than or equal to 97. Figure 5 shows the resulting data. This plot suggests a slight asymmetry and tail dependence.

Tree structure and pair-copulas families When applied to the complete data (1,738 observations), Algorithm 1 selects the D-vine tree structure that agrees with time order. That is, the variables are ordered by time in the first tree of the D-vine. We also conduct 1,000 simulations to study the impact of non-response on the selected tree structure. For each simulation run, we randomly generate a monotone non-response pattern as follows: we randomly partition the units into five subsamples $s_1, s_2, \ldots, s_5$ of approximately equal size; for a unit $k \in s_\ell$, $\ell = 1, 2, \ldots, 5$, we discard the values x_{ki} for $i > \ell$. Note that the missing data are MCAR. For each incomplete data generated, we apply Algorithm 1 and select a D-vine tree structure. Table 1 shows the selected tree structures with the frequency of occurrence across 1,000 simulations.

Only 5 out of the 16 possible tree structures are observed. The tree with the variables ordered by time is selected for almost 75% of the simulation runs. The other four selected tree structures are very close to the first one. First, the orders are very similar. For instance, the order of the first two tree structures is the same except for the fifth variable. Second, the strength of dependency accounted for in the first tree is very similar across the five selected tree structures. Indeed, the average Kendall's τ of the pairs in the first tree for these five tree structures computed on the complete data ranges from 0.578 (fifth tree structure) to 0.593 (first tree structure). As a result, there would be almost no difference in terms of accuracy and stability of the parameters estimators between these five tree structures. We consider the D-vine with the variables ordered by time in the first tree in what follows.

We apply the procedure of Section 5 to the complete data to select the pair-copula families of the D-vine. We obtain survival Gumbel family for pairs 1, 2, 4 and 10, Gumbel family for

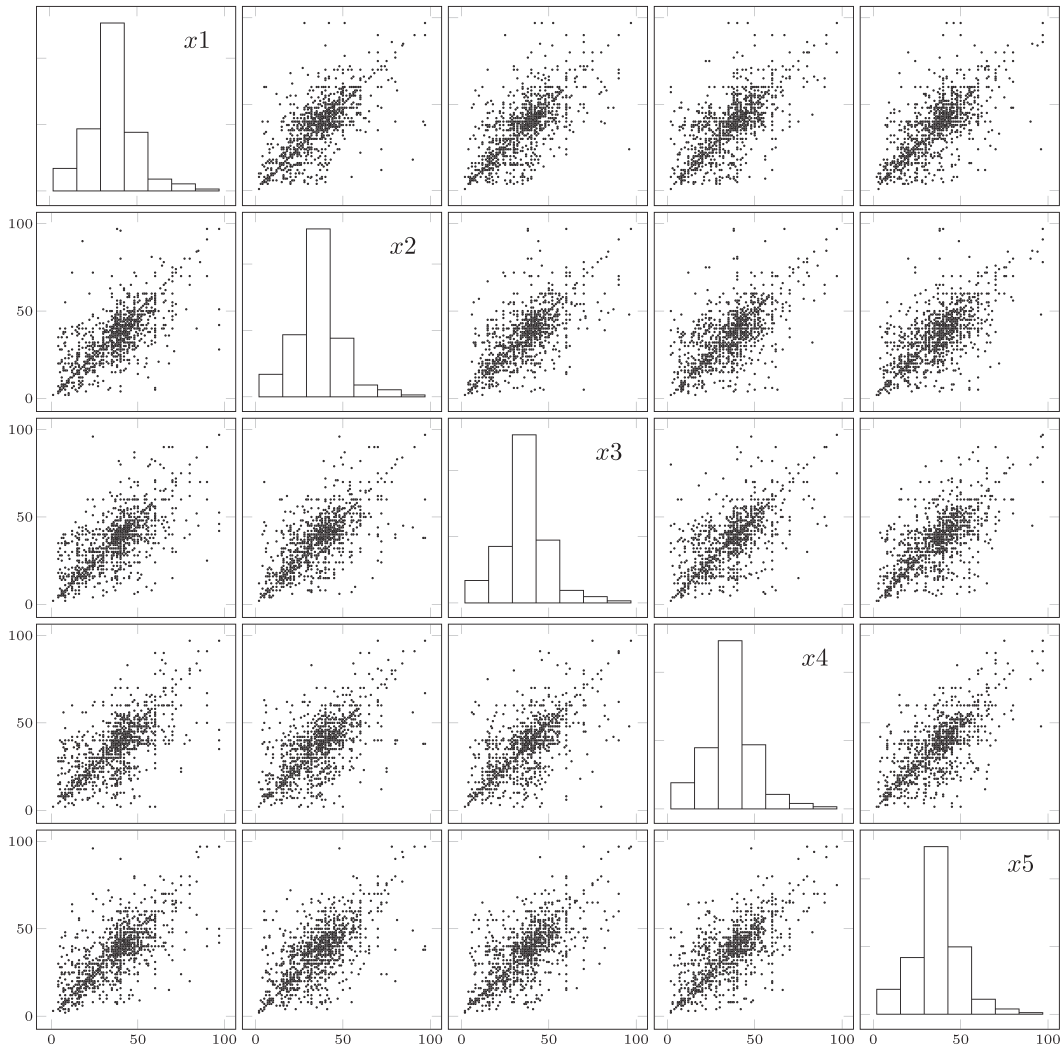

Figure 5. *Total actual hours of 1 738 individuals for five consecutive quarters.*

Table 1. *Frequency of the orders selected with Algorithm 1 for the total actual hours of 1 738 individuals for five consecutive quarters over 1 000 simulations.*

Order	Frequency
1 - 2 - 3 - 4 - 5	0.73
5 - 1 - 2 - 3 - 4	0.20
5 - 4 - 1 - 2 - 3	0.03
5 - 4 - 3 - 1 - 2	0.03
4 - 1 - 2 - 3 - 5	<0.01

pairs 3, 5, 6, 7 and 9 and Frank family for pair 8. This confirms the presence of asymmetry and tail dependence that we have graphically observed.

Imputation We conduct 200 simulations to compare the performance of our method to other methods. For each simulation run, we randomly generate a monotone non-response pattern as described previously. For each simulation run, we impute the missing data using six imputation methods:

1. **MEan IMPutation (MEIMP)** The missing values of each variable are replaced with the mean of this variable.
2. **AMElia (AME)**(Honaker *et al.*, 2011) The algorithm runs an EM algorithm on each of multiple bootstrapped samples selected from the incomplete data. Then, it draws a set of imputed values from the parameters estimated from each bootstrap sample (multiple imputation). We select five bootstrap samples. Amelia assumes a multivariate normal distribution. We apply function boxcox of R package Mass (Venables & Ripley, 2002) to check whether a Box–Cox power transformation should be used to achieve normality. The selected parameters of the Box–Cox transformation being close to 1, we use the original untransformed data.
3. **Multivariate Imputation by Chained Equations (MICE)**(van Buuren & Groothuis-Oudshoorn, 2011) MICE imputes the data by chained equations. It assumes an imputation model separately for each column in the data. For continuous variable, it applies predictive mean matching cyclicly. The idea of predictive matching is the following: (i) it fits a linear model with the variable being imputed as dependent variable and some fully observed covariates; (ii) it predicts the missing values of the variable being imputed using the fitted model; and (iii) it imputes a missing value with the observed value of the variable being imputed that is the closest to the fitted value. MICE starts with an initial imputation of each variable. Then, it cyclicly imputes each variable with predictive mean matching with the other variables (observed and imputed values) as covariates until a convergence criterion is reached. Finally, the algorithm generates multiple imputations for incomplete multivariate data by Gibbs sampling. We consider five multiple imputations. With continuous variables, MICE assumes normality of the variables. We use here the original untransformed data (same reason as for AME).
4. **Copula IMPutation (COIMP)**(Di Lascio and Giannerini, 2014; Di Lascio *et al.*, 2015) COIMP is a copula-based method to impute multivariate missing data. Four steps of the method are as follows: (i) non-parametric estimation of the margins through local polynomial likelihood and parametric estimation of the copula model through maximum likelihood on the available data; (ii) derivation of the joint distribution; (iii) derivation of the conditional distribution of the missing values, conditioned on the observed values; and (iv) imputation by generating from the conditional distribution of the previous step with the Hit or Miss Monte Carlo method. The copula models allowed are normal, Frank, Clayton and Gumbel. Note that this method is the slowest among the five considered. For Labour Force Survey data, the normal copula model is selected (for both the complete data and data with non-response) and kept constant throughout the simulations. We carry out five repeated imputations (multiple imputation).
5. **Nearest Neighbor imputation (NN)** We use function impute.NN_HD of R package HotDeckImputation (Joenssen, 2015). Function impute.NN_HD finds the nearest neighbor in the complete cases for each case with missing values using the observed values of this case. The Manhattan distance is considered. The variables are scaled with respect to their range prior to computing the distance.
6. **Pair-copula Imputation (PCI)** We consider the D-vine with the variables ordered by time in the first tree and the pair-copulas families selected from the complete data (see previous

paragraph). We keep them constant throughout the simulations so that we can study the effect of imputing solely on the estimates. For each simulation run, we estimate the pair-copula parameters using the procedure described in Section 4, and we sample imputed values from the conditional distribution of the missing values, conditioned on the observed values using the procedure described in Section 3.2. We carry out five repeated imputations (multiple imputation). We transform the variables back to the original scale using the inverse of the functions $\widehat{F}_i$ described in Section 4.1.

For each imputation method and each simulation run, we estimate three vectors of parameters of interest: (i) the mean of each variable, (ii) the Pearson's correlation of each pair of variables and (iii) the 99th percentile of each variable. Appendix S1 explains how to compute point and variance estimates with multiple imputation. Consider $Q^{(i)}$ the point estimate obtained at the i-th simulation run for a given imputation method and a generic vector of parameters of interest Q. We assess the performance of the imputation methods via three criteria:

1. The Monte Carlo relative bias (in absolute value) defined as

$$\mathrm{RB} = \frac{|\mathrm{B}|}{\mathrm{Q}},$$

   where $\mathrm{B} = \overline{Q}^{(I)} - Q$ is the Monte Carlo bias and $\overline{Q}^{(I)} = \sum_{i=1}^{I} Q^{(i)}/I$ is the average point estimate over the $I = 200$ simulations.
2. The Monte Carlo relative root variance (or relative standard deviation) defined as

$$\mathrm{RRV} = \frac{\mathrm{V}^{1/2}}{Q},$$

   where

$$\mathrm{V} = \frac{1}{I-1}\sum_{i=1}^{I}\left(Q^{(i)} - \overline{Q}^{(I)}\right)^2.$$

3. The Monte Carlo relative root mean square error defined as

$$\mathrm{RRMSE} = \frac{\left(\mathrm{B}^2 + \mathrm{VAR}\right)^{1/2}}{Q}.$$

Figure 6 shows the results.

The top plots show three comparison criteria for the mean of the variables ordered by decreasing number of observed values, the middle plots three comparison criteria for the Pearson's correlation coefficient of the pairs of the variables ordered by decreasing number of pairwise observed values and the bottom plots three comparison criteria for the 99th percentile of the variables ordered by decreasing number of observed values. Appendix S2 contains three tables showing these results. We observe that PCI is associated with smallest variance in all considered cases considered except one. A possible explanation is that our estimation method (Section 4) uses all the information in the data. We discuss the bias associated with the six imputation methods independently for each parameter of interest. For the mean, all five methods are associated with an RB less than 3%. MEIMP, AMEL, MICE and NN yield the smallest RB. The reason is that the mean is a measure of central tendency, which is unrelated to the dependence and tails structure. Therefore, an imputation method that imputes each variable separately (MEIMP),

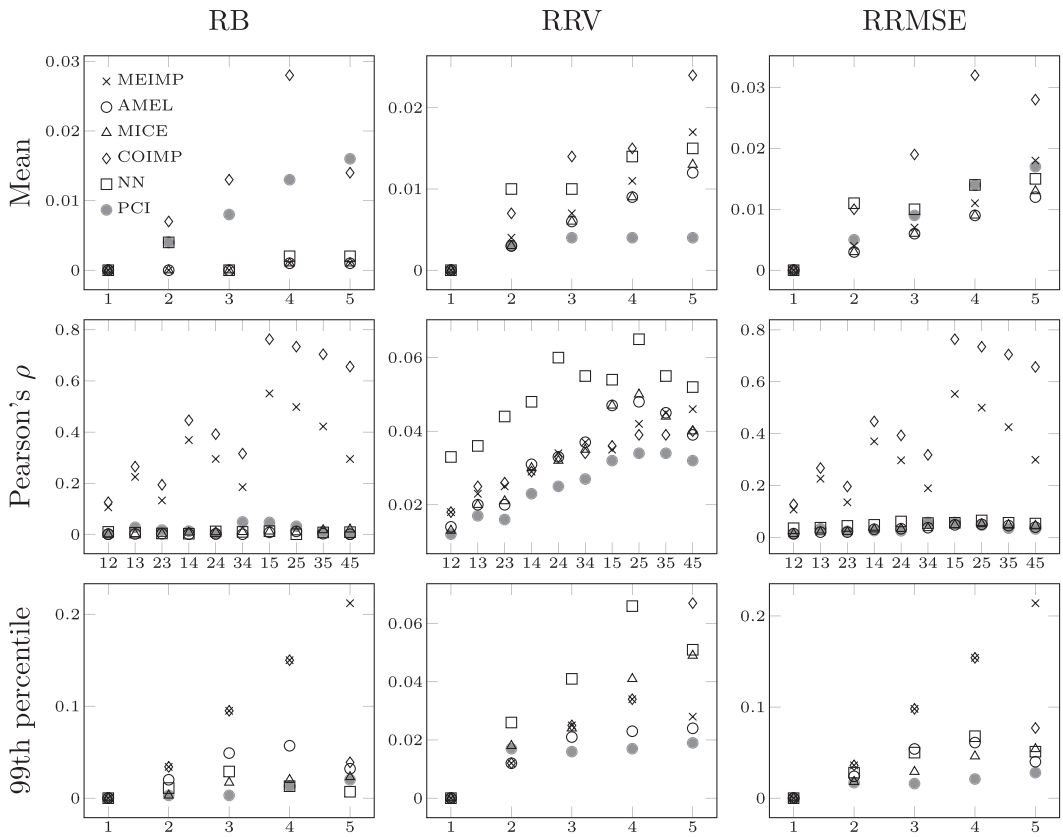

Figure 6. *Three comparison criteria (from left to right: RB, RRV and RRMSE) for three parameters of interest (from top to bottom: mean, Pearson's correlation coefficient and 99th percentile) with six imputation methods for Labour Force Survey data. The x-axis shows the index of the variables or pair of variables for Pearson's correlation coefficient. RB, relative bias; RRV, relative root variance; RRMSE, relative root mean square error.*

independently to the dependence structure (MEIMP) and the tails features (MEIMP, AMEL, MICE and NN) provides satisfactory results. COIMP shows the poorest performance; it is also the slowest. For the Pearson's correlation coefficient, AMEL, MICE, NN and PCI yield the smallest RB. This was expected because these four imputation methods account for the dependence structure of the data. For this parameter of interest, MEIMP and COIMP yield an RB that can be as high as nearly 60% and 80%, respectively. For MEIMP, the reason is that it does not account for the dependence structure as the variables are imputed separately. The bad performance of COIMP, however, is surprising because this method accounts for the dependence structure via a copula model. A possible explanation is that the family of copulas used by this method is not wide enough to capture the dependence structure of the data. For the 99th percentile, PCI globally provides the best results. The reason is that this method accounts for the tails features.

7.2 Simulated Data 1

We simulated a sample of size $n = 500$ from a four-dimensional D-vine with Gumbel pair copulas with parameter equal to 2. Figure 7 shows the simulated data.

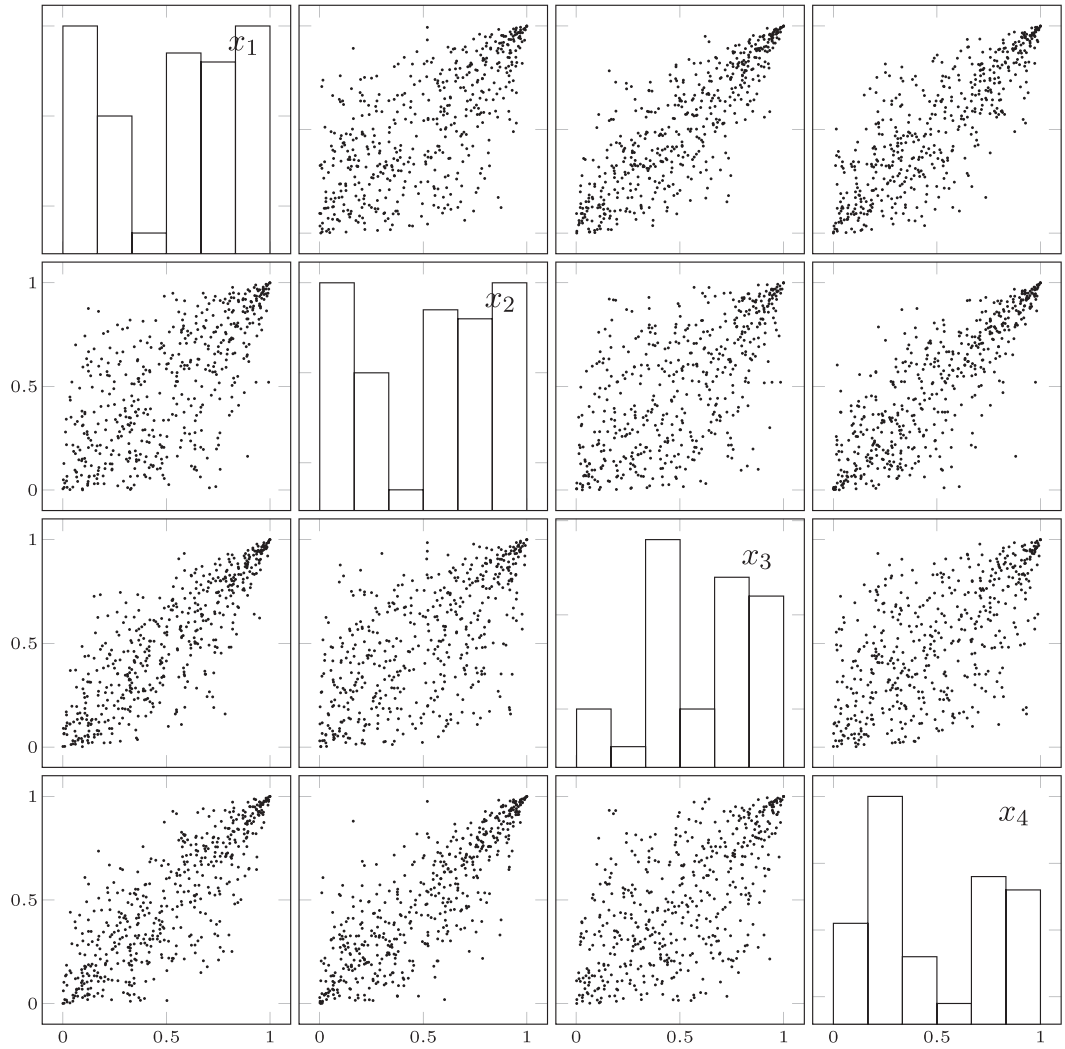

Figure 7. *Five hundred observations from a four-dimensional D-vine with Gumbel pair copulas with parameter equal to 2.*

The model for the joint distribution is correctly specified by PCI and incorrectly specified by AME. As a multivariate Gumbel copula cannot be reconstructed using bivariate Gumbel copulas, the model is also misspecified for COIMP. We conduct 200 simulations to compare the performance of our imputation method to other imputation methods as described in Section 7.1. We apply function boxcox of R package Mass (Venables & Ripley, 2002) to check whether a Box–Cox power transformation should be used to achieve normality. The selected parameters of the Box–Cox transformation being close to 1, we apply AME and MICE to the original untransformed data. We use Gumbel copula for the multivariate copula of COIMP and the pair copulas of PCI. For PCI, we select the D-vine tree structure where the variables are in decreasing order of observed values in the first tree. Figure 8 shows the results of the simulations. Appendix S2 also contains three tables showing these results.

These results are similar to those of Section 7.1. PCI overall performs the best with the smallest RRMSE. It provides the smallest RRV but is not necessarily the best in terms of RB.

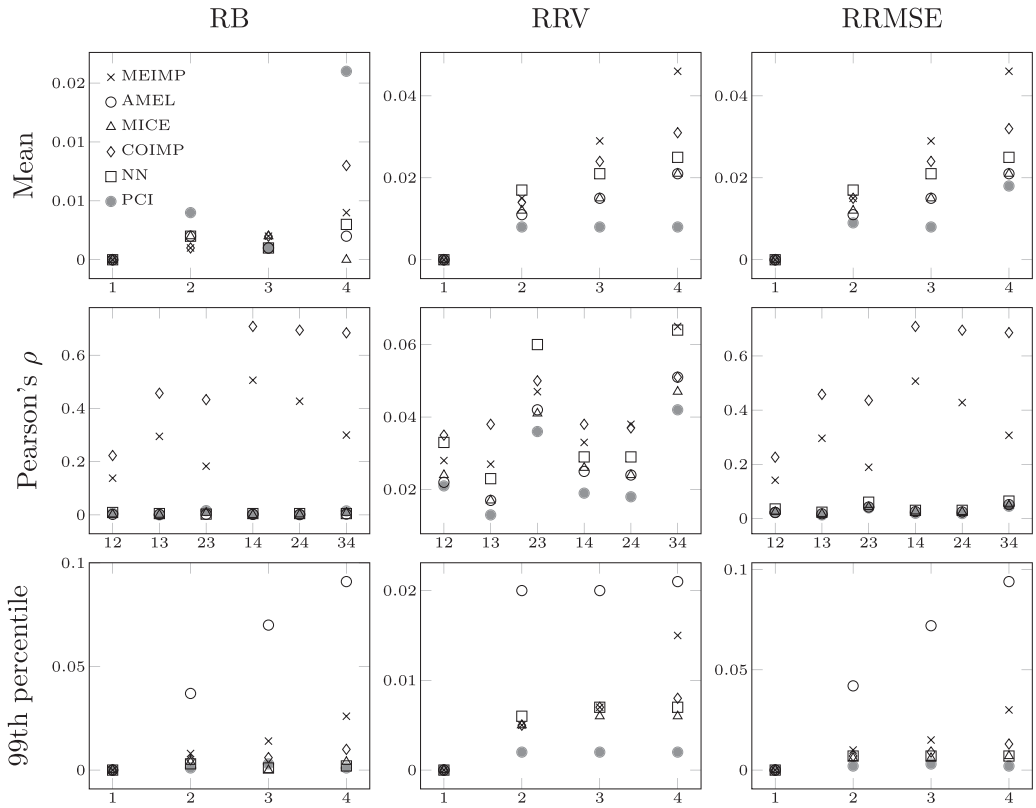

Figure 8. *Three comparison criteria (from left to right: RB, RRV and RRMSE) for three parameters of interest (from top to bottom: mean, Pearson's correlation coefficient and 99th percentile) with six imputation methods for simulated data 1. The x-axis shows the index of the variables or pair of variables for Pearson's correlation coefficient. RB, relative bias; RRV, relative root variance; RRMSE, relative root mean square error.*

7.3 Simulated Data 2 (misspecified models)

We generate $n = 1\,000$ independent and identically distributed observations $(x_{k1}, x_{k2}, x_{k3}, x_{k4}), k = 1, \ldots, 1\,000$ of a vector of four random variables as follows:

$$\begin{aligned} x_{k1} &= 3 + u_k + \varepsilon_{k1}, \\ x_{k2} &= u_k + \varepsilon_{k2}, \\ x_{k3} &= \log(u_k + 1) + \varepsilon_{k3}, \\ x_{k4} &= 1 + \exp^{1/2} u_k + \varepsilon_{k4}, \end{aligned}$$

where u_k is generated from a uniform distribution on interval $[1, 2]$ and ε_{k1}, ε_{k2}, ε_{k3} and ε_{k4} from normal distributions with mean 0 and standard deviation $u_k^{1/10}$, 1/10, 1/10 and $(u_k - 1)^{1/4}$, respectively. Variable u is a latent variable that creates dependency between the variables of interest. Figure 9 shows the simulated data.

The joint distribution is incorrectly specified by all three JM methods (AME, COIMP and PCI). We conduct 200 simulations to compare the performance of our imputation method to other imputation methods as described in Section 7.1. We apply function boxcox of R package mass (Venables & Ripley, 2002) to check whether a Box–Cox power transformation should be used to achieve normality. The selected parameters of the Box–Cox transformation being close

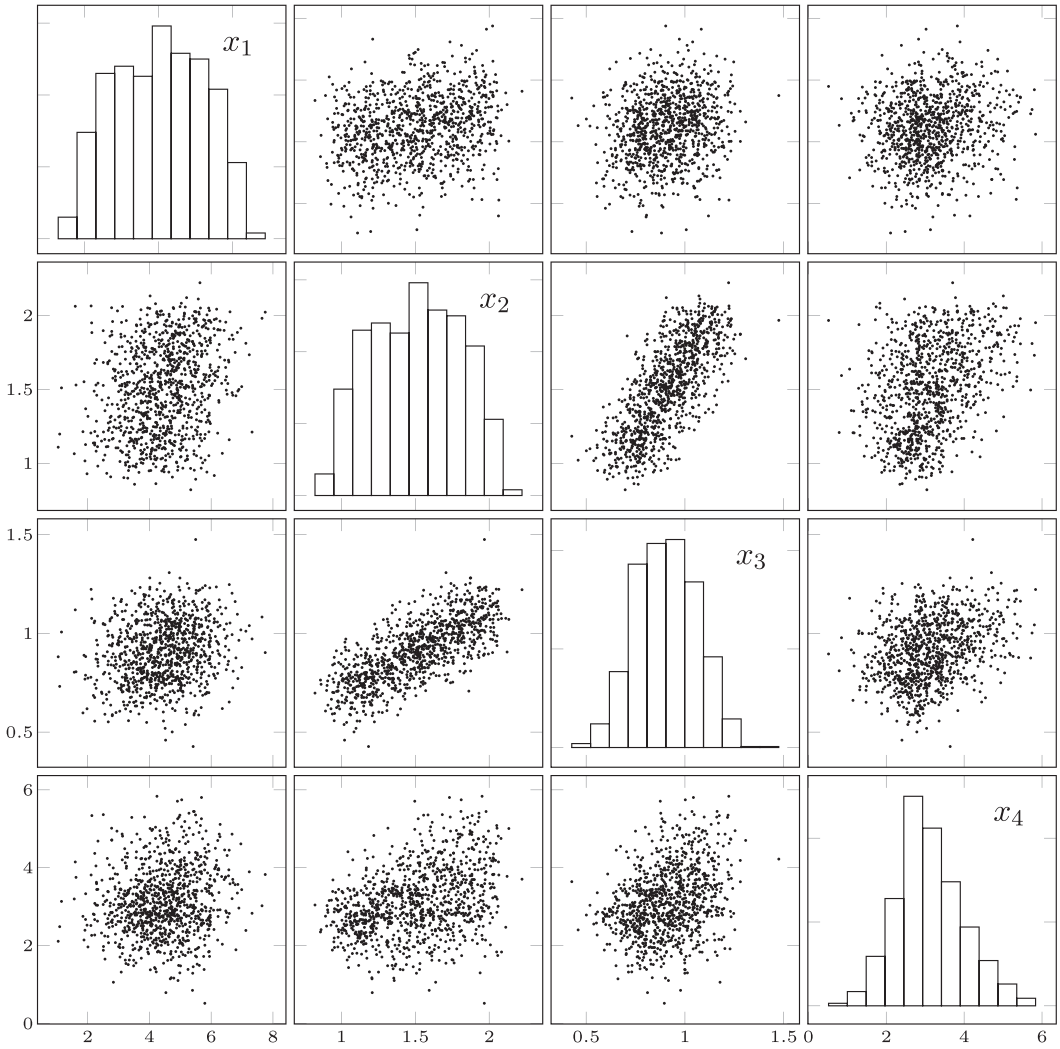

Figure 9. *One thousand independent and identically distributed observations.*

to 1, we apply AME and MICE to the original untransformed data. For COIMP, we use the complete data to select the copula families. We obtain the normal copula and consider this one across the 200 simulations. For PCI, we transform the variables as described in Section 7.1 to obtain uniform margins. We use the complete data to select the pair-copula families and the D-vine tree structure. We obtain the D-vine tree structure where the variables are in decreasing order of observed values in the first tree. We obtain Frank copula for pairs 1, 2, 3 and 5 and independence copula for pairs 4 and 6. We consider this D-vine tree structure and these pair-copula families across the 200 simulations. Figure 10 shows the results of the simulations. Appendix S2 also contains three tables showing these results.

Pair-copula imputation overall performs the best with the smallest RRMSE for all three parameters of interest. For two parameters of interest (mean and 99th percentile), PCI, NN and MICE perform equally and better than the other methods in terms of RB, and PCI performs the best in terms of RRV. For the Pearson's correlation coefficient, AMEL, MICE and NN yield the

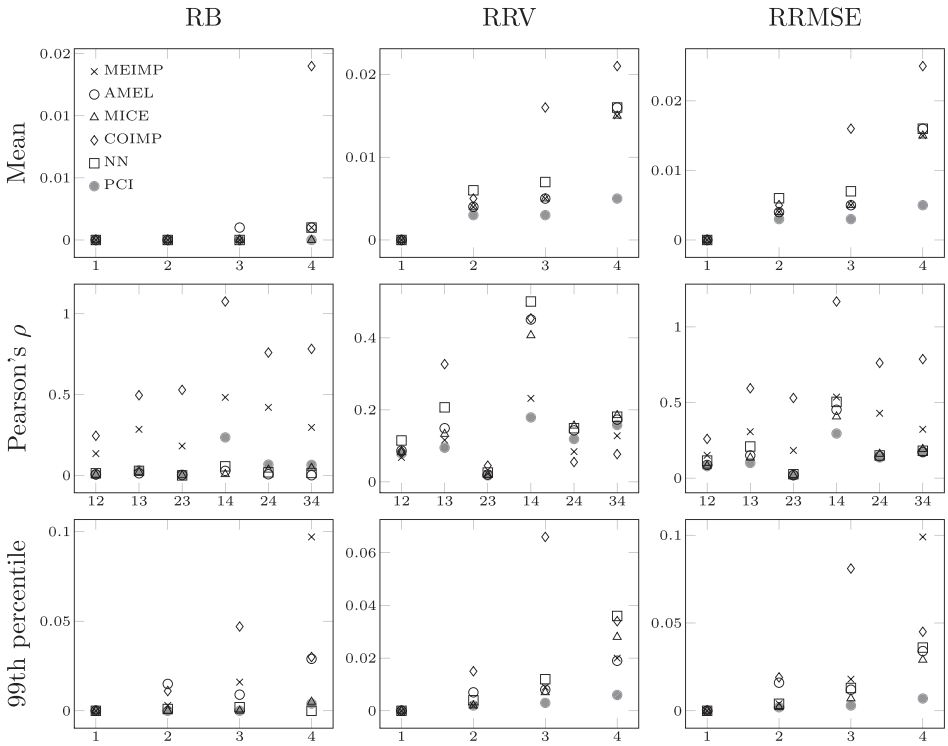

Figure 10. *Three comparison criteria (from left to right: RB, RRV and RRMSE) for three parameters of interest (from top to bottom: mean, Pearson's correlation coefficient and 99th percentile) with six imputation methods for simulated data 2. The x-axis shows the index of the variables or pair of variables for Pearson's correlation coefficient. RB, relative bias; RRV, relative root variance; RRMSE, relative root mean square error.*

smallest RB. JM methods are usually criticised for their lack of flexibility. The results of this setting show that PCI, as a JM method, is very flexible; it still performs very well under model misspecification.

8 Discussion

8.1 Generalisation to Data Missing at Random

We suppose in the paper that the missing data are MCAR. Unfortunately, the proposed D-vine imputation method does not work when the missing data are missing at random (MAR). In this case, (5) does not represent the likelihood contribution of a unit with non-response, and in (6), $\widehat{F}_i$ is not a consistent estimator of the marginal distribution function of X_i. Adapting the D-vine model to MAR data is not straightforward. The estimator for F_i could possibly be corrected using a model for the response mechanism, as considered in Cassel *et al.* (1983), and parameter estimation could be performed through a conditional likelihood. Additional research is needed to investigate the feasibility of these changes.

8.2 Generalisation to More Global Non-response Patterns

This section briefly discusses the generalisation of our method to more global non-response patterns. We have seen that the estimation method presented in Section 4 can be applied if

and only if any group of variables that are jointly observed in the sample defines a sub-D-vine of the original D-vine. More generally, our method can be applied if we observe only non discontinuous sequences of variables, that is, if it is possible to rearrange the variables such that, for each unit k, there exist two scalars ℓ^- and ℓ^+ such that $1 \leq \ell^- < \ell^+ \leq d$ and x_{ki} is observed for $\ell^- \leq i \leq \ell^+$ and missing for $i < \ell^-$ or $i > \ell^+$. In this case, the sample can be partitioned into $d(d-1)/2$ subsamples:

$$s_{\ell^-,\ell^+} = \left\{k | x_{ki} \text{ is observed if and only if } \ell^- \leq i \leq \ell^+\right\},$$

where $\ell^- = 1, \ldots, d-1$, $\ell^+ = \ell^- + 1, \ldots, d$. Figure 11 shows the non-response patterns for which our estimation method can be applied and the sub-D-vines associated in the four-dimensional case.

In dimension d, there are therefore $d(d-1)/2$ non-response patterns that our method can handle. Note that our estimation method presented in Section 4 and imputation methods in Section 3.2 might need to be adapted to suit the non-response patterns presented in this section.

So far, we have considered that all variables are observed for some units (subsample s_{14} in the four-dimensional case). Consider now that we jointly observe at most $p < d$ variables in the sample. A typical example is a longitudinal study with a cyclical selection: once a unit is sampled, it is retained for exactly p consecutive waves. In this case, the sample provides no

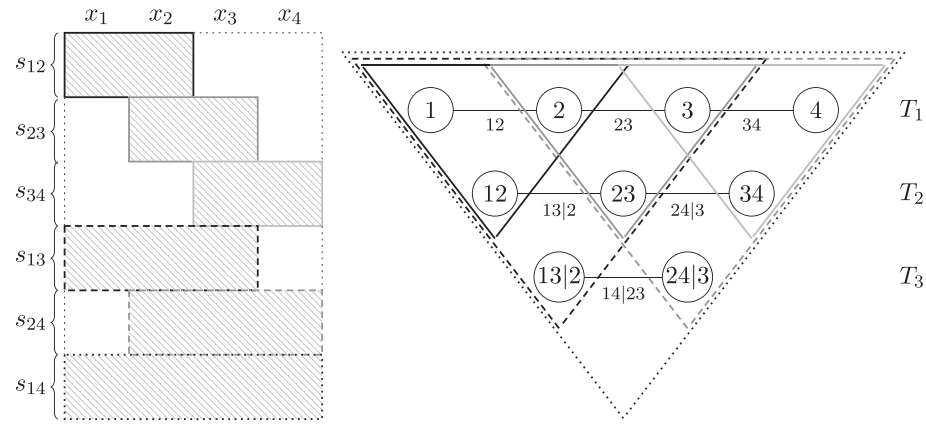

Figure 11. *Non-response patterns for which our estimation method can be applied (left) and sub-D-vines associated (right) in the four-dimensional case.*

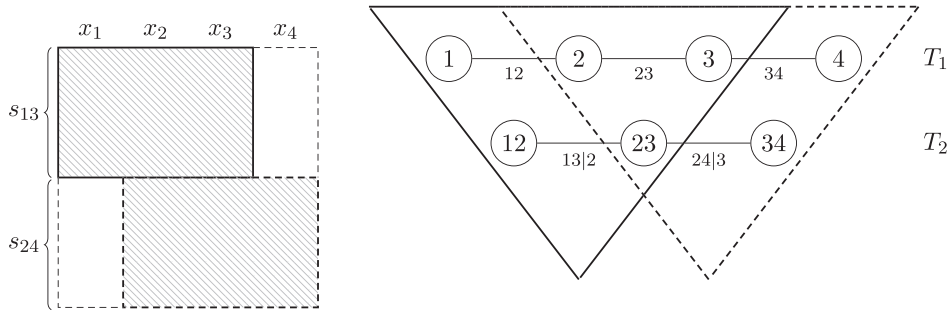

Figure 12. *Non-response pattern when units are retained for three consecutive waves in the four-dimensional case, truncated tree at level 2 and D-vines associated with the subsamples.*

information about the pair copulas in the trees p to $d - 1$. A solution to apply our method is to set all pair copulas in trees p and higher to independence copulas. This implies a truncated tree at level $p - 1$ (Brechmann *et al.*, 2012). Figure 12 shows an example where the units are retained for three consecutive waves ($p = 3$) in the four-dimensional case. In this case, the sample provides no information about the pair copula $C_{14|23}$ of Tree T_3 because we never observe jointly all four variables. This pair is set to independence copula, which results in a truncated tree at level 2. Each subsample is associated with a D-vine, and our method can be applied.

9 Conclusion

The paper proposes an imputation method for continuous variables based on vine copulas that yields a flexible joint model using a cascade of bivariate copulas. Our method can capture some complex features of the data such as tail dependence that other methods fail to capture, and it does not restrict the marginals to belong to preset families. We suppose MCAR data and a monotone non-response pattern, and, in this case, our proposed method outperforms all competing alternatives. It would be interesting to generalise our method to MAR data and non-monotone non-response patterns.

Acknowledgements

This research is supported by the Canadian Statistical Sciences Institute and by NSERC of Canada. The authors thank a reviewer, the Editor and Pr. Harry Joe for constructive comments.

Supporting Information

Additional supporting information may be found online in the supporting information tab for this article.

References

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

Bedford, T. & Cooke, R. M. (2002). A new graphical model for dependent random variables. *Ann. Statist.*, **30**(4), 1031–1068.

Brechmann, E. C., Czado, C. & Aas, K. (2012). Truncated regular vines in high dimensions with application to financial data. *Can. J. Stat.*, **40**(40), 68–85.

Cassel, C. M., Särndal, C.-E. & Wretman, J. H. (1983). Some uses of statistical models in connexion with the nonresponse problem. In *Incomplete Data in Sample Surveys*, Vol. 3, Eds. W. G. Madow & I. Olkin, pp. 143–160. New York: Academic Press.

Chauvet, G. & Haziza, D. (2012). Fully efficient estimation of coefficients of correlation in the presence of imputed survey data. *Can. J. Stat.*, **40**(1), 124–149.

Di Lascio, F. M. L. & Giannerini, S. (2014). *CoImp: Copula based imputation method*. R package version 0.2-3.

Di Lascio, F. M. L., Giannerini, S. & Reale, A. (2015). Exploring copulas for the imputation of complex dependent data. *Stat. Methods Appl.*, **24**(1), 159–175.

Ding, W. & Song, P. X. -K. (2016). EM algorithm in Gaussian copula with missing data. *Comput. Stat. Data Anal.*, **101**, 1–11.

Gelman, A. & Hill, J. (2011). Opening windows to the black box. *J. Stat. Softw.*, **40**(3), 1–25.

Genest, C., Ghoudi, K. & Rivest, L.-P. (1995). A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, **82**(3), 543–552.

Harrell, Jr, F. E. with contributions from Charles Dupont and many others. (2016). *Hmisc: Harrell Miscellaneous*. R package version 3.17-2.

Honaker, J., King, G. & Blackwell, M. (2011). Amelia II: A program for missing data. *J. Stat. Softw.*, **45**(7), 1–47.
Joe, H. (1996). Families of m-variate distributions with given margins and $m(m-1)/2$ bivariate dependence parameters. In *Distributions with Fixed Marginals and Related Topics*, Eds. L. Rueschendorf, B. Schweizer & M. Taylor, pp. 120–141. Hayward, CA, IMS Lecture Notes-Monograph Series.
Joe, H. (2014). *Dependence Modeling with Copulas.* Boca Raton, FL: Chapman and Hall/CRC.
Joenssen, D. W. (2015). *HotDeckImputation: Hot deck imputation methods for missing data*. R package version 1.1.0.
Käärik, E. & Käärik, M. (2009). Modeling dropouts by conditional distribution, a copula-based approach. *J. Stat. Plan. Infer.*, **139**, 3830–3835.
Office for National Statistics. Social Survey Division and Northern Ireland Statistics and Research Agency. Central Survey Unit. (2015). *Labour force survey five-quarter longitudinal dataset*. january 2014 - march 2015. 2nd edition. Colchester, Essex: UK Data Archive [distributor].
Raghunathan, T. E., Lepkowski, J. M., van Hoewyk, J. & Solenberger, P. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, **27**(1), 85–95.
Rubin, D. B. (1976). Inference and missing data. *Biometrika*, **63**, 581–592.
Rubin, D. B. (1987). *Multiple Imputation for Nonreponse in Surveys.* New York: Wiley.
Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data.* London: Chapman and Hall.
van Buuren, S. (2007). Multiple imputation of discrete and continuous data by fully conditional specification. *Stat. Methods Med. Res.*, **16**(3), 219–242.
van Buuren, S. & Groothuis-Oudshoorn, K. (2011). MICE: Multivariate imputation by chained equations in R. *J. Stat. Softw.*, **45**(3), 1–67.
Venables, W. N. & Ripley, B. D. (2002). *Modern Applied Statistics with S*, 4th ed. New York: Springer. ISBN 0-387-95457-0.

[*Received July 2016, accepted February 2018*]