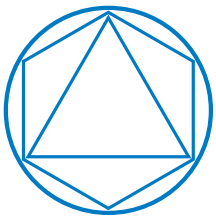

Technische Universität München

Department of Mathematics

D- and C-vine quantile regression for large data sets

Master's Thesis

by

Marija Tepegjozova

Supervisor:	Prof. Claudia Czado, Ph.D.
Advisor:	Prof. Claudia Czado, Ph.D.
Submission date:	November 20, 2019

I hereby declare that this thesis is my own work and that no other sources have been used except those clearly indicated and referenced.

Munich, November 20, 2019

Acknowledgments

First of all, I would like to express my sincere gratitude to Prof. Claudia Czado for the opportunity to work under her supervision on this project. I would like to thank her for all the guidance and support provided. Also, I am especially grateful for the encouragement to continue evolving as a researcher.

Second, I would like to thank Nicole Barthel, for all the help she provided, for the useful advices, the support and guidance. Further, a special thanks to Alexander Kreuzer for providing me technical support for the LRZ cluster.

Also, I would like to thank my colleagues and friends Ozge, Anes and many others, for all the support provided.

Finally, I want to express my deep gratitude to my family, for all the unconditional love and support they provided all these years.

Abstract

Quantile regression is a field with growing importance for statistical modeling. It has a broad range of applications and it emerged as a complementary method to linear regression in many fields. Ever since the formal definition for quantile regression has been formulated, by Koenker and Bassett Jr (1978), there have been many attempts to improve this methods shortfalls. The occurrence of quantile crossings and the linearity assumption are just a few disadvantages of linear quantile regression.

One of the ideas how to overcome such shortfalls, is to use vine copula based quantile regression. Vine-copulas allow highly flexible modeling of high-dimensional dependence structures. The first work in this field by Kraus and Czado (2017), introduced D-vine quantile regression for the subclass of D-vines. The idea how to build the D-vine copula is based on the maximal improvement of the conditional log likelihood in the next tree. Our first goal is to extend this method to the subclass of C-vines, so that more flexible dependence structures can be modeled. The next step is to introduce new algorithms for both D-vine and C-vine copulas, where we look on the next two trees for the maximal improvement in the conditional log likelihood. Furthermore, an additional goal is to be able to use these algorithms on big data sets, and thus, we introduce some modifications in our two step ahead algorithm in order to reduce the computational complexity.

At the end, we try to examine the performance of the algorithms through an extensive simulation study, where we compare the proposed algorithms based on several performance measures which include, among others, the out of sample mean square error, conditional log likelihood and computational time.

Contents

Part I

Theoretical background

Chapter 1

Concept of copulas

In order to define the copula approach to multivariate data we will start with univariate and multivariate distributions, and gradually define the concept of copulas, vine copulas and its properties. In this section, we are following Czado (2019).

1.1 Univariate and multivariate distributions

In general, we denote random variables by capital letters and observed values by small letters, i.e., we write $X = x$. We consider absolute continuous functions, so that corresponding density functions exist. We use F for the distribution function and f for the corresponding density.
Furthermore, the parameters of the distribution function of a random variable X are unknown and need to be estimated. That can be done by considering a sample $x_1, ..., x_n$ of independent identically distributed (i.i.d.) observations of X. One way to do such estimation, is to parametrically model X with a parameter vector $\boldsymbol{\theta} \in \Theta$, i.e., $X \sim f(\cdot; \boldsymbol{\theta})$, where Θ is the corresponding parameter space. The parameter vector is most often estimated with maximum likelihood method in which we have

$$\widehat{\boldsymbol{\theta}} := \arg\max_{\boldsymbol{\theta} \in \Theta} \prod_{i=1}^{n} f(x_i; \boldsymbol{\theta}).$$

So, the distribution function $F(\cdot; \boldsymbol{\theta})$ is estimated by $F(\cdot; \widehat{\boldsymbol{\theta}})$.
If one does not want to make any parametric assumptions, most often the empirical distribution is being used.

Definition 1.1. (Empirical distribution function)
Let $x_1, ..., x_n$ be an i.i.d. sample from a distribution function F, then the empirical distribution is defined as

$$\widehat{F}(x) := \frac{1}{n+1} \sum_{i=1}^{n} I(x_i \leq x),$$

for all x.

Remark 1.2.

- *$I(x_i \leq x)$ is an indicator function, being equal to 1, when the condition $x_i \leq x$ is satisfied, and equal to 0, otherwise.*
- *Division by $n+1$ instead of n is used in order to avoid boundary problems.*
- *So, the distribution function $F(x)$ is estimated by $\widehat{F}(x)$.*

Now, we will continue talking about modeling the behaviour of multiple random variables using multivariate distributions. In such a case, we distinguish between marginal, joint and conditional distributions which arise from the multivariate distribution. The joint distribution consists of marginal and conditional distributions, in a way that the marginals describe the behaviour of a single variable on its own, and the conditional distributions describe the effect of a conditioning set of variables over the conditioned variables. For each of them, we will use the following notation:

Definition 1.3. *For a random vector $\boldsymbol{X} = (X_1, ..., X_d)^T$ we define:*

- ***joint** distribution and density of $\boldsymbol{X}$: $F(\boldsymbol{x})$ and $f(\boldsymbol{x})$ for $\boldsymbol{x} = (x_1, ..., x_d)^T$,*
- ***marginal** distribution and density function of X_j: $F_j(x_j)$ and $f_j(x_j)$, for $j = 1, ..., d$,*
- ***conditional** distribution and density function of X_j given X_i: $F_{j|i}(x_j|x_i)$ and $f_{j|i}(x_j|x_i)$ for $j \neq i$.*

Next, in order to be able to characterize the dependence between the random variables, we need to standardize them. Therefore each of the random variables X_j for $j = 1, ...d$ is standardized using the probability integral transform, defined as:

Definition 1.4. (Probability Integral Transform)
Let $X \sim F$ be a continuous random variable and let x be an observed value of X. Then the transformation

$$u := F(x)$$

is called the probability integral transform (PIT) at x.

Remark 1.5. (Distribution of the probability integral transform)

- *If $X \sim F$, then $U := F(X)$ is uniformly distributed, because*

$$P(U \leq u) = P\left(F(X) \leq u\right) = P\left(X \leq F^{-1}(u)\right) = F\left(F^{-1}(u)\right) = u,$$

  holds for every $u \in [0, 1]$.
- *If F is estimated parametrically by $F(\cdot; \widehat{\boldsymbol{\theta}})$ or nonparametrically by the empirical distribution $\widehat{F}$, then this holds only approximately.*

Now, if the parametric assumptions are to be avoided, we use the multivariate empirical distribution function.

Definition 1.6. (Multivariate empirical distribution function)
Let $x_i = (x_{1i}, ..., x_{di})$ be an i.i.d. sample of size n from the multivariate $d-$dimensional distribution F, then the multivariate empirical distribution function is defined as

$$\widehat{F}(\boldsymbol{x}) := \frac{1}{n+1} \sum_{i=1}^{n} 1_{[x_{1i} \leq x_1, ..., x_{di} \leq x_d]},$$

for all $\boldsymbol{x} := (x_1, ..., x_d)^T$.

1.2 Copulas

The copula approach to multivariate data allows individual modelling of the marginal distributions. Also, it separates the dependence between the components from the marginal distributions. The dependence of a marginally standardised multivariate distribution is modeled by a corresponding joint distribution function called a copula.

Definition 1.7. (Copula)
A $d-$dimensional copula C is a multivariate distribution function on the $d-$dimensional hypercube $[0,1]^d$ with uniformly distributed marginals.

Remark 1.8. (Copula density)
The corresponding copula density for an absolutely continuous copula, denoted by c, can be computed by partial differentiation

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

for all $\boldsymbol{u} :=(u_1, ..., u_d)^T \in [0,1]^d$.

One of the fundamental results in the copula methods is the Sklar's Theorem. Sklar (1959) proved the representation theorem for multivariate distributions in terms of their marginal distributions and a corresponding copula. The proof can be found in Nelsen (2007).

Theorem 1.9. (Sklar's Theorem)
Let $\boldsymbol{X} := (X_1, ..., X_d)^T$ be a $d-$dimensional random vector with joint distribution function F and marginal distribution functions F_i, $i = 1, ..., d$, then the joint distribution function can be expressed as

$$F(x_1, ..., x_d) = C(F_1(x_1), ..., F_d(x_d)),$$

with associated density or probability mass function

$$f(x_1, ..., x_d) = c(F_1(x_1), ..., F_d(x_d)) f_1(x_1) ... f_d(x_d), \tag{1.1}$$

for some $d-$dimensional copula C with copula density c. For absolutely continuous distributions, the copula C is unique.
The inverse also holds: the copula corresponding to a multivariate distribution function F with marginal distribution functions F_i, $i = 1, ..., d$ can be expressed as

$$C(u_1, ..., u_d) = F\left(F_1^{-1}(u_1), ..., F_d^{-1}(u_d)\right),$$

and its copula density or probability mass function is determined by

$$c(u_1, ..., u_d) = \frac{f\left(F_1^{-1}(u_1), ..., F_d^{-1}(u_d)\right)}{f_1\left(F_1^{-1}(u_1)\right) ... f_d\left(F_d^{-1}(u_d)\right)}.$$

Lemma 1.10. (Conditional densities and distribution functions of bivariate distributions in therms of their copula)
The conditional density can be rewritten as

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

and the distribution function as

$$F_{1|2}(x_1|x_2) = \frac{\partial}{\partial u_2}\left(F_1(x_1), u_2\right)|_{u_2=F_2(x_2)},$$

which we abbreviate by

$$\frac{\partial}{\partial F_2(x_2)} C_{12}\left(F_1(x_1), F_2(x_2)\right).$$

Proof. Using the definition of a conditional density and the equation (1.1) we have

$$\begin{aligned}
f_{1|2}(x_1|x_2) =& \frac{f_{12}(x_1, x_2)}{f_2(x_2)} \\
&= \frac{c_{12}(F_1(x_1), F_2(x_2)) f_1(x_1) f_2(x_2)}{f_2(x_2)} \\
&= c_{12}(F_1(x_1), F_2(x_2)) f_1(x_1) \\
&= \frac{\partial^2 C_{12}(u_1, u_2)}{\partial u_1 \partial u_2}|_{u_1=F_1(x_1), u_2=F_2(x_2)} \frac{\partial u_1}{\partial x_1} \\
&= \frac{\partial}{\partial u_2}\left(\frac{\partial}{\partial x_1} C_{12}(F_1(x_1), u_2)\right)|_{u_2=F_2(x_2)}.
\end{aligned}$$

Thus, we have

$$\begin{aligned}
F_{1|2}(x_1|x_2) =& \int_{-\infty}^{x_1} \frac{\partial}{\partial u_2}\left(\frac{\partial}{\partial z_1} C_{12}(F_1(z_1), u_2)\right)|_{u_2=F_2(x_2)} dz_1 \\
=& \frac{\partial}{\partial u_2}\left(\int_{-\infty}^{x_1} \frac{\partial}{\partial z_1} C_{12}(F_1(z_1), u_2) dz_1\right)|_{u_2=F_2(x_2)} \\
=& \frac{\partial}{\partial u_2} C_{12}(F_1(x_1), u_2)|_{u_2=F_2(x_2)}.
\end{aligned}$$

□

Remark 1.11. (Bivariate conditional copula and conditional distribution)

- *Lemma 1.10 can be applied to the bivariate copula distribution C_{12}. We denote the conditional distribution and density as $C_{1|2}$ and $c_{1|2}$, respectively. We have that*

$$C_{1|2}(u_1|u_2) = \frac{\partial}{\partial u_2} C_{12}(u_1, u_2) \qquad \forall u_1 \in [0,1].$$

- *The relationship between $F_{1|2}$ and $C_{1|2}$ using Lemma 1.10 is given by*

$$F_{1|2}(x_1|x_2) = \frac{\partial}{\partial u_2} C_{12}(F_1(x_1), u_2)|_{u_2=F_2(x_2)} = C_{1|2}\left(F_1(x_1), F_2(x_2)\right).$$

- *Using all of the above, the inverse function of the conditional distribution function is given as*

$$F_{1|2}^{-1}(u_1|x_2) = F_1^{-1}\left(C_{1|2}^{-1}(F_1(x_1)|F_2(x_2))\right).$$

The conditional distribution function $C_{1|2}$ associated with a copula is also known as a h function, notation introduced by Aas et al. (2009).

Definition 1.12. (h-functions of bivariate copulas)
For all $(u_1, u_2) \in [0,1]^2$ the h-functions that correspond to a bivariate copula C_{12} are defined as

$$h_{1|2}(u_1|u_2) = \frac{\partial}{\partial u_2} C_{12}(u_1, u_2),$$

and

$$h_{2|1}(u_2|u_1) = \frac{\partial}{\partial u_1} C_{12}(u_1, u_2).$$

1.3 Dependence measures

In order to capture and quantify the dependence between random variables we introduce dependence measures, such as Kendall's tau and partial correlation, which we later use to introduce our algorithms.

Definition 1.13. (Kendall's tau)
The Kendall's tau between two continuous random variables X_1 and X_2 is defined as

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

where (X_{11}, X_{12}) and (X_{21}, X_{22}) are independent and identically distributed copies of the pair (X_1, X_2).

Lemma 1.14. (Kendall's tau expressed in terms of the copula)
Let X_1 and X_2 be two continuous random variables, then Kendall's tau can be expressed in terms of their bivariate copula $C(u_1, u_2)$ as

$$\tau = 4 \int_{[0,1]^2} C(u_1, u_2) \quad dC(u_1, u_2) - 1.$$

Proof. This result follows directly from Definition 1.13 and the proof can be seen in Czado (2019). □

Remark 1.15. *Kendall's tau, τ is a rank-based dependence measure and it is invariant with respect to monotone transformations of the margins. Also, it can be expressed in terms of the associated copula, as seen in the Definition 1.14, so its values does not depend on the marginal distributions. That is why it is so suitable dependence measure when working with copulas.*

In the case of d variables we look into the dependence of every pair of variables. A dependence measure between two variables after the effect of the remaining variables is removed is called partial correlation.

Definition 1.16. (Partial correlation)
Let $X_1, \ldots, X_d$ be random variables with zero mean and variance σ_i^2 for $i = 1, \ldots, d$. Let $I^d_{-(i,j)}$ be the set $\{1, \ldots, d\}$ with removed indices i and j, for $i \neq j$. Define partial regression coefficients $b_{i,j;I^d_{-(i,j)}}$ for $i < j$ as the quantities that minimize

$$E[(X_i - \sum_{j=2, j\neq i}^{d} a_{i,j;I^d_{-(i,j)}} X_j)^2].$$

Then, define the partial correlation $\rho_{i,j;I^d_{-(i,j)}}$ as

$$\rho_{i,j;I^d_{-(i,j)}} := sign(b_{i,j;I^d_{-(i,j)}}) \times \sqrt{b_{i,j;I^d_{-(i,j)}} \times b_{i,j;I^d_{-(i,j)}}}.$$

We use the following recursion formula for the computation of the partial correlations, proved by Yule and Kendall (1950).

Theorem 1.17. (Recursion for partial correlations)
The partial correlation satisfy the following

$$\rho_{i,j;I^d_{-(i,j)}} = \frac{\rho_{i,j;I^{d-1}_{-(i,j)}} - \rho_{i,d;I^{d-1}_{-(i,j)}} \rho_{j,d;I^{d-1}_{-(i,j)}}}{\sqrt{1 - \rho_{i,d;I^{d-1}_{-(i,j)}}} \times \sqrt{1 - \rho_{j,d;I^{d-1}_{-(i,j)}}}}.$$

In order to examine the probability of the joint occurrence of extremely small or big values we define the following coefficients

Definition 1.18. (Upper and lower tail dependence coefficient)
The upper tail dependence of a bivariate distribution with copula C is defined as

$$\begin{aligned}\lambda^{upper} &= \lim_{t\to 1^-} P\left(X_2 > F_2^{-1}(t) | X_1 > F_1^{-1}(t)\right) \\ &= \lim_{t\to 1^-} \frac{1-2t+C(t,t)}{1-t},\end{aligned}$$

and the lower dependence coefficient as

$$\begin{aligned}\lambda_{lower} &= \lim_{t\to 0^+} P\left(X_2 \leq F_2^{-1}(t) | X_1 \leq F_1^{-1}(t)\right) \\ &= \lim_{t\to 0^+} \frac{C(t,t)}{t}.\end{aligned}$$

1.4 Bivariate copulas

We start with the introduction of parametric copulas in 2 dimensions. They are characterized by the copula family and the corresponding parameter. We now introduce various examples of bivariate parametric copulas. Depending on their construction, we distinguish between elliptical copulas, based on elliptical distributions using the inverse statement of Sklar's Theorem 1.9, such as Gaussian and Student t copula, and Archemedian copulas, constructed using a generator function ϕ.

Example 1.19. (Elliptical copulas)

- *Bivariate Gaussian copula*
  Let $\Phi_1(\cdot)$ be the distribution function of a univariate standard normal distribution, i.e. $N(0,1)$ and let $\Phi_2(\cdot)$ be the distribution function of a bivariate standard normal distribution $N\left((0,0)^T, \Sigma\right)$, where Σ is a symmetric positive definite 2×2 correlation matrix with unit variance. Then, by applying the inverse Sklar's theorem 1.9 we obtain the bivariate Gaussian copula (abbreviated by Gauss. in later use)

$$C(u_1,u_2) = \Phi_2\left(\Phi_1^{-1}(u_1), \Phi_1^{-1}(u_2)\right).$$

- *Bivariate Student t copula*
  Let $T_{1,v}(\cdot)$ be the distribution function of a univariate standard Student t distribution with $v>0$ degrees of freedom, $t_1(v,0,1)$ and let $T_{2,v}(\cdot)$ be the distribution function of a bivariate standard Student t distribution, $t_2(v,(0,0)^T,\Sigma)$, where Σ is a scale parameter matrix , $\Sigma \in [-1,1]^{2\times 2}$. By applying the inverse Sklar's theorem 1.9 we obtain the bivariate Student t copula

$$C(u_1,u_2) = T_2\left(T_1^{-1}(u_1), T_1^{-1}(u_2)\right).$$

Example 1.20. (Archimedean bivariate copulas with a single parameter)

- *Clayton copula*

$$C(u_1, u_2) = (u_1^{-\delta} + u_2^{-\delta} - 1)^{-\frac{1}{\delta}},$$

where $0 < \delta < \infty$ is the parameter controlling the degree of dependence. Independence correspond to $\delta \to 0$, while full dependence when $\delta \to \infty$.

- *Gumbel copula*

$$C(u_1, u_2) = exp[-\{(-\ln u_1)^{\delta} + (-\ln u_2)^{\delta}\}^{\frac{1}{\delta}}],$$

where $\delta \geq 1$ is similarly the parameter controlling the degree of dependence. Independence correspond to $\delta = 1$, while full dependence when $\delta \to \infty$.

- *Frank copula*

$$C(u_1, u_2) = -\frac{1}{\delta} \ln \left(\frac{1}{1 - e^{-\delta}} [(1 - e^{-\delta}) - (1 - e^{-\delta u_1})(1 - e^{-\delta u_2})] \right),$$

where the parameter δ is in $[-\infty, \infty] \setminus \{0\}$.

- *Joe copula*

$$C(u_1, u_2) = 1 - \left((1 - u_1)^{\delta} + (1 - u_2)^{\delta} - (1 - u_1)^{\delta}(1 - u_2)^{\delta} \right)^{\frac{1}{\delta}},$$

where the parameter is $\delta \geq 1$.

Example 1.21. (Independence copula)
In order to model independence between two uniformly distributed variables U_1 and U_2 we define the nonparametric (or parametric with parameter set to 0) independence copula (abbreviated by Indep. in later use)

$$C(u_1, u_2) = u_1 \times u_2.$$

Definition 1.22. (Rotated copulas)
In order to extend the range of dependence we use counterclockwise rotations of the copula density $c(\cdot, \cdot)$ by

- *90 degrees: $c_{90}(u_1, u_2) := c(1 - u_1, u_2)$,*
- *180 degrees: $c_{180}(u_1, u_2) := c(1 - u_1, 1 - u_2)$,*
- *270 degrees: $c_{270}(u_1, u_2) := c(u_1, 1 - u_2)$.*

Remark 1.23. *We also note that there is a one-to-one correspondence between the copula parameter and the Kendal's tau, as proved by Embrechts et al. (2003) for the elliptical copulas, and by Hürlimann (2003) for the Archimedean copulas.*

After defining many parametric bivariate copulas, we proceed with their estimation. Since we are working with parametric copulas, we also need a way to estimate the parameter for the copula. As a usual goodness of fit measure we will use log likelihood, Akaike information criterion (AIC) and the Bayesian information criterion (BIC). We define them as follows:

Definition 1.24. (Conditional log likelihood)
Let $C(\cdot,\cdot)$ be a bivariate parametric copula with density $c(\cdot,\cdot)$ and parameter $\boldsymbol{\theta}$, where $|\boldsymbol{\theta}|$ denotes the dimension of $\boldsymbol{\theta}$. Also, consider a copula data with size n $\boldsymbol{u} = (u_1^{(i)}, u_2^{(i)})$ for $i = 1, \ldots, n$. The log likelihood, with and without the penalizations AIC and BIC of the bivariate copula are defined as

- ***log likelihood:***

$$l(\boldsymbol{\theta}, \boldsymbol{u}) := \sum_{i=1}^{n} log(c(u_1^{(i)}, u_2^{(i)})),$$

- ***BIC penalized log likelihood:***

$$l_{BIC}(\boldsymbol{\theta}, \boldsymbol{u}) := -2l(\boldsymbol{\theta}, \boldsymbol{u}) + log(n)|\boldsymbol{\theta}|,$$

- ***AIC penalized log likelihood:***

$$l_{AIC}(\boldsymbol{\theta}, \boldsymbol{u}) := -2l(\boldsymbol{\theta}, \boldsymbol{u}) + 2|\boldsymbol{\theta}|.$$

Chapter 2

Introduction to C- and D-vine copulas

In this section using the building blocks we defined in the previous chapter, the bivariate copulas or the pair copulas, we continue with constructing multivariate distributions using conditioning. The first approach developed was introduced by Joe (1996) in which the construction of a multivariate copula was in terms of distribution functions, while Bedford and Cooke (2001) independently developed constructions in terms of densities. This method is known as pair copula construction (or PCC) and is the building block for defining vine copulas. To illustrate this method we start with an example, given in Czado (2019).

2.1 Regular vines

Example 2.1. (Pair copula construction in 3 dimensions)
Let X_1, X_2 and X_3 are random variables and our goal is to write their joint density $f(x_1, x_2, x_3)$ in terms of pair copulas and marginal densities.
We start with the recursive factorization formula for the joint density

$$f(x_1, x_2, x_3) = f_{3|12}(x_3|x_1, x_2)\ f_{2|1}(x_2|x_1)\ f_1(x_1). \tag{2.1}$$

We continue with decomposing each term in the above expression. Consider $f_{2|1}(x_2|x_1)$ and using the properties of conditional density and Sklar's Theorem 1.9 we get

$$\begin{aligned} f_{2|1}(x_2|x_1) &= \frac{f_{12}(x_1, x_2)}{f_1(x_1)} \\ &= \frac{c_{12}(F_1(x_1), F_2(x_2)\ f_1(x_1)\ f_2(x_2)}{f_1(x_1)} \\ &= c_{12}(F_1(x_1), F_2(x_2)\ f_2(x_2). \end{aligned} \tag{2.2}$$

Next, we derive $f_{3|12}(x_3|x_1,x_2)$*. In order to do so, we consider the bivariate conditional density* $f_{13|2}(x_1,x_3|x_2)$*. The copula density associated with the conditional distribution of* (X_1,X_3) *given* $X_2=x_2$ *is denoted by* $c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)$*. Thus, by Sklar's Theorem 1.9 we have*

$$f_{13|2}(x_1,x_3|x_2)=c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)\ f_{1|2}(x_1|x_2)\ f_{3|2}(x_3|x_2). \tag{2.3}$$

Plugging in equations (2.3) *and into the conditional density* $f_{3|12}(x_3|x_1,x_2)$ *we have*

$$\begin{aligned}
f_{3|12}(x_3|x_1,x_2) &= \frac{f(x_1,x_2,x_3)}{f(x_1,x_2)}\\
&= \frac{f_{13|2}(x_1,x_3|x_2)\ f_2(x_2)}{f_{12}(x_1,x_2)}\\
&= \frac{f_{13|2}(x_1,x_3|x_2)}{f_{1|2}(x_1|x_2)}\\
&= \frac{c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)\ f_{1|2}(x_1|x_2)\ f_{3|2}(x_3|x_2)}{f_{1|2}(x_1|x_2)}\\
&= c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)\ f_{3|2}(x_3|x_2)\\
&= c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)\ c_{23}(F_2(x_2),F_3(x_3))\ f_3(x_3).
\end{aligned} \tag{2.4}$$

Merging the results from equations (2.2)*) and* (2.4) *into equation* (2.1) *we finally get the pair copula decomposition in 3 dimensions*

$$\begin{aligned}
f(x_1,x_2,x_3) = {} & c_{13;2}(F_{1|2}(x_1|x_2),F_{3|2}(x_3|x_2);x_2)\ c_{12}(F_1(x_1),F_2(x_2))\\
& \times c_{23}(F_2(x_2),F_3(x_3))\ f_1(x_1)\ f_2(x_2)\ f_3(x_3).
\end{aligned} \tag{2.5}$$

However, this decomposition of the joint density is not unique, we can decompose $f(x_1,x_2,x_3)$ *also in two other ways, by reordering the variables* X_1,X_2 *and* X_3*. Namely, the following decompositions are also valid:*

$$f(x_1,x_2,x_3)=f_{2|13}(x_2|x_1,x_3)\ f_{1|3}(x_1|x_3)\ f_3(x_3),$$

and

$$f(x_1,x_2,x_3)=f_{1|23}(x_1|x_2,x_3)\ f_{3|2}(x_3|x_2)\ f_2(x_2).$$

Therefore, we have two additional valid decomposition of the joint density of X_1,X_2,X_3 *in terms of pair copulas as*

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

and

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

In the first given pair copula decomposition (2.5) *the copula density* $c_{13;2}(\cdot\ \cdot; x_2)$ *depends on the value* $X_2 = x_2$*. However, a standard assumption in vine copula theory is to ignore this dependence. In this case, we speak of making the so called* ***simplifying assumption****:*

$$c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2); x_2) = c_{13;2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)). \tag{2.6}$$

Assuming the simplifying assumption (2.6) *on the right hand side of equation* (2.5)*, this decomposition can also be used in a constructive manner. We transform the data on a copula level using the Probability integral transform, where* $F_i(x_i) := u_i$ *for* $i = 1, 2, 3$*. We specify the copula family and parameter for each of the copulas* $c_{13;2}$*,* c_{12} *and* c_{23}*. Since the margins* f_1, f_2 *and* f_3 *on a copula level are uniform, we define a three dimensional copula as*

$$c(u_1, u_2, u_3) = c_{13;2}(C_{1|2}(u_1|u_2), C_{3|2}(u_3|u_2)\ c_{12}(u_1, u_2)\ c_{23}(u_2, u_3),$$

where $C_{1|2}(\cdot|u_2)$ *and* $C_{3|2}(\cdot|u_2)$ *are the conditional distribution functions of* U_1 *given* $U_2 = u_2$ *and* U_3 *given* $U_2 = u_2$*, respectively.*

Furthermore, using Sklar's Theorem 1.9 the copula density corresponding to the distribution of (U_1, U_3) given $U_2 = u_2$ can be calculated as

$$c_{13;2}(u_1, u_3; u_2) = \frac{c_{123}\left(C^{-1}_{1|2}(u_1|u_2), u_2, C^{-1}_{3|2}(u_3|u_2)\right)}{c_{12}(C^{-1}_{1|2}(u_1|u_2), u_2)\ c_{23}\ (u_2, C^{-1}_{3|2}(u_3|u_2))},$$

where the inverse conditional functions are given as

$$C^{-1}_{1|2}(u_1|u_2) = F_1^{-1}(F^{-1}_{1|2}(u_1|x_2))$$

$$C^{-1}_{3|2}(u_3|u_2) = F_3^{-1}(F^{-1}_{3|2}(u_3|x_2)).$$

Following this manner we can use the pair copula construction (PCC) method to construct any d-dimensional multivariate distribution.

Definition 2.2. (Regular vine (R-vine) tree sequence)
The set of trees $\mathcal{V} = (T_1, ..., T_{d-1})$ *is a regular vine tree sequence on* d *elements if*

1. *Each tree* T_j *is connected, i.e. for all nodes* $a, b \in T_j, j = 1, ..., d-1$*, there exists a path* $(n_1, ..., n_k) \in N_j^k$ *with* $a = n_1$ *and* $b = n_k$*.*
2. T_1 *is a tree with node set* $N_1 = \{1, ..., d\}$ *and edge set* E_1*.*
3. *For* $j \geq 2$*,* T_j *is a tree with node set* $N_j = E_{j-1}$ *and edge set* E_j*.*
4. *For* $j = 2, ..., d-1$ *and* $\{a, b\} \in E_j$ *it must hold that* $|a \cap b| = 1$*.*

Remark 2.3. *The property* (4.) *is called the proximity condition. It ensures that if there is an edge* e *that connects nodes* a *and* b *in tree* T_j*, for* $j \geq 2$*, then the edges* a *and* b *in* T_{j-1} *must share a common node in* T_{j-1}*.*

Now, we define some notation that will help us to define the regular vine distribution and its properties.

Definition 2.4. (Complete union, conditioning and conditioned set)
Let $\mathcal{V} = (T_1, ..., T_{d-1})$ be a regular vine tree sequence. For an edge $e = \{a, b\} \in E_i$ we define the following sets

- ***complete union*** A_e *of the edge* e

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

- ***conditioning set*** D_e

$$D_e := A_a \cap A_b,$$

- ***conditioned sets*** $C_{e,a}$ *and* $C_{e,b}$

$$C_{e,a} := A_a \backslash D_e$$

$$C_{e,b} := A_b \backslash D_e$$

$$C_e := C_{e,a} \cup C_{e,b}.$$

We often abbreviate each edge $e = (C_{e,a}, C_{e,b}; D_e)$ in the vine tree sequence by

$$e = (e_a, e_b; D_e).$$

Definition 2.5. (Regular vine distribution)
A joint distribution F for the $d-$dimensional random vector $\boldsymbol{X} = (X_1, ..., X_d)$ is called regular vine distribution, if we can find a triplet $(\mathcal{F}, \mathcal{V}, \mathcal{B})$ such that:

1. ***Marginal distributions:*** *$\mathcal{F} = (F_1, ..., F_d)$ is a vector of continuous invertible distribution functions, representing the marginal distribution functions of the random variable $X_i, i = 1, ..., d$.*

2. ***Regular vine tree sequence:*** *$\mathcal{V}$ is an R-vine tree sequence on d elements.*

3. ***Bivariate copulas:*** *$\mathcal{B} = \{C_e | e \in E_i; i = 1, ..., d-1\}$ where C_e is a symmetric bivariate copula with density and E_i is the edge set of tree T_i in the R-vine sequence.*

4. ***Connection between tree sequence and bivariate copulas:*** *For each $e \in E_i$, $i = 1, ..., d-1$, $e = \{a, b\}$, C_e is the corresponding copula to the conditional distribution of $X_{C_{e,a}}$ and $X_{C_{e,b}}$ given $\boldsymbol{X}_{D_e} = \boldsymbol{x}_{D_e}$. Furthermore $C_e(\cdot, \cdot)$ does not depend on the specific value of $\boldsymbol{x}_{D_e}$, which is called the simplifying assumption.*

Definition 2.6. (Pair copula and copula density associated with edge e)
We will denote the copula C_e corresponding to the edge e by $C_{C_{e,a}C_{e,b};D_e}$ and the corresponding density by $c_{C_{e,a}C_{e,b};D_e}$. This copula is also called pair copula.

Definition 2.7. (Regular vine copula)
Let $(\mathcal{F}, \mathcal{V}, \mathcal{B})$ be a regular vine distribution on the $d-$dimensional random vector $\boldsymbol{U} = (U_1, ..., U_d)$ where $U_j \sim U(0,1)$ for $j = 1, ..., d$. Then the R vine copula can be identified by the triplet $(\mathcal{V}, \mathcal{B}(\mathcal{V}), \Theta(\mathcal{B}(\mathcal{V})))$ where

1. *$\mathcal{V}$ denotes the vine tree structure,*
2. *$\mathcal{B}(\mathcal{V})$ gives all pair copula families associated with each edge in $\mathcal{V}$,*
3. *$\Theta(\mathcal{B}(\mathcal{V}))$ denotes the associated pair copula parameters to each member in $\mathcal{B}(\mathcal{V})$.*

Remark 2.8. *Since we will consider only parametric pair copulas, we will denote the parameter associated with copula $C_{C_{e,a}C_{e,b};D_e}$ as $\boldsymbol{\theta}_{C_{e,a}C_{e,b};D_e}$. For example $\boldsymbol{\theta}_{1,2}$ is the associated parameter with the copula $C_{1,2}$ or $\boldsymbol{\theta}_{1,2;3}$ is the associated parameter with the copula $C_{1,2;3}$.*

2.2 C- and D-vine copulas

Next, we consider two important subclasses of regular vine tree sequence, drawable (D-) vine and canonical (C-) vine tree sequence.

Definition 2.9. (D-vine tree sequence)
A regular vine tree sequence $\mathcal{V} = (T_1, ..., T_{d-1})$ is called a D-vine tree sequence if for each node $n \in N_i$ we have $|\{e \in E_i | n \in e\}| \leq 2$.

Remark 2.10.

- *The proximity condition for the D-vine tree sequence implies that once the first tree T_1 in the sequence is defined, then all the other trees are determined, $T_2, ..., T_{d-1}$.*
- *From a graph theoretical perspective, in the D-vine tree sequence all trees are paths. The nodes whose degree equals 1, are called leaf nodes.*
- *Since the tree structure of a D-vine resembles a grape vine, Bedford and Cooke (2001) called the linked tree sequence of Definition 2.9 a vine.*

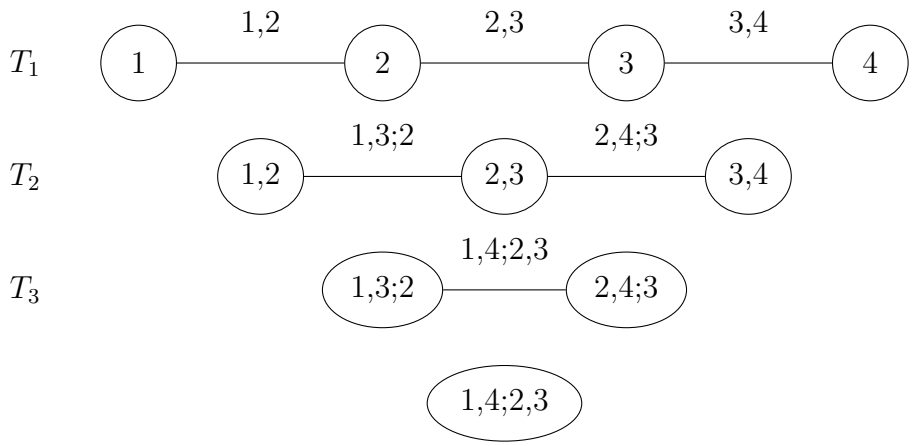

Figure 2.1: 4-dimensional D-vine.

Example 2.11. (Example of D-vine tree sequence in 4 dimensions)
Figure 2.1 shows the tree sequence of a 4-dimensional D-vine. As we already said before, the first tree defines the whole tree sequence for a D-vine, $\mathcal{V} = (T_1, T_2, T_3)$.

Definition 2.12. (C-vine tree sequence)
A regular vine tree sequence $\mathcal{V} = (T_1, ..., T_{d-1})$ *is called* $C-$*vine tree sequence if in each tree* T_i *there is one node* $n \in N_i$ *such that* $|\{e \in E_i | n \in e\}| = d - i$. *That node is called the root node of tree* T_i.

Remark 2.13.

- *The proximity condition for the C-vine tree sequence implies that we can choose the root node from* $d - i + 1$ *nodes in tree* T_i *for* $i = 1, ..., d - 1$.
- *From a graph theoretical perspective, in the C-vine tree sequence all trees are stars.*

Definition 2.14. (Simplifying assumption for D- and C-vines)
If it holds that

$$c_{ij;D}\left(F_{i|D}(x_i|\boldsymbol{x}_D), F_{j|D}(x_j|\boldsymbol{x}_D); \boldsymbol{x}_D\right) = c_{ij,D}\left(F_{i|D}(x_i|\boldsymbol{x}_D), F_{j|D}(x_j|\boldsymbol{x}_D)\right)$$

for all $\boldsymbol{x}_D$ *and* i, j *and* D *are chosen to occur in , then the corresponding D-vine (C-vine) is called simplified.*

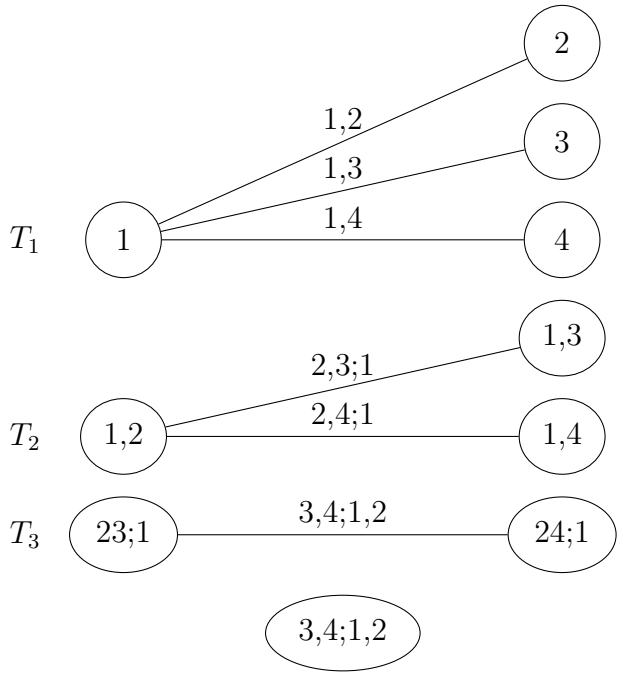

Figure 2.2: 4-dimensional C-vine.

Example 2.15. (Example of C-vine tree sequence in 4 dimensions)
Figure 2.2 shows the tree sequence of a 4-dimensional C-vine. In the tree sequence $\mathcal{V} = (T_1, T_2, T_3)$ *we can see that all trees are stars, where in each tree there a root node connected to all other nodes in that tree level.*

Theorem 2.16. (Drawable (D-vine) density)
The joint density $f_{1,\dots,d}$ *can be decomposed as*

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

The distribution associated with this density decomposition is called a drawable vine or shortened D-vine.

Next, using Bedford and Cooke (2001) we define the C-vine density.

Theorem 2.17. (Canonical (C-vine) density)
The joint density $f_{1,\dots,d}$ *can be decomposed as*

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

The distribution associated with this density decomposition is called a canonical vine or shortened C-vine.

Example 2.18. (Simplified 4-dimensional D-vine and C-vine density)
The density of the D-vine from the Example 2.11 will be given as

$$\begin{aligned} f_{1,2,3,4}(x_1,x_2,x_3,x_4) =& \left[\prod_{i=1}^{4} f_i(x_i)\right] \cdot c_{12}(x_1,x_2) \cdot c_{23}(x_2,x_3) \cdot c_{34}(x_3,x_4) \\ =& c_{13;2}\left(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)\right) \cdot c_{24;3}\left(F_{2|3}(x_2|x_3), F_{4|3}(x_4|x_3)\right) \\ =& c_{14;23}\left(F_{1|23}(x_1|x_2,x_3), F_{4|23}(x_4|x_2,x_3)\right). \end{aligned}$$

The density of the C-vine from Example 2.15 will be given as

$$\begin{aligned} f_{1,2,3,4}(x_1,x_2,x_3,x_4) =& \left[\prod_{i=1}^{4} f_i(x_i)\right] \cdot c_{12}(x_1,x_2) \cdot c_{13}(x_1,x_3) \cdot c_{14}(x_1,x_4) \\ =& c_{23;1}\left(F_{2|1}(x_2|x_1), F_{3|1}(x_3|x_1)\right) \cdot c_{24;1}\left(F_{2|1}(x_2|x_1), F_{4|1}(x_4|x_1)\right) \\ =& c_{34;12}\left(F_{3|12}(x_3|x_1,x_2), F_{4|12}(x_4|x_1,x_2)\right). \end{aligned}$$

Definition 2.19. (D- and C-vine copula)
If a regular vine copula is based on a C-vine tree sequence $\mathcal{V}$, we call it a C-vine copula. If it is based on a D-vine tree sequence, it is called a D-vine copula.

2.3 Conditional distribution functions

Definition 2.20. (General notation of h-functions of bivariate copulas)
Given the bivariate copula C_{ij} we define the h functions

$$h_{i|j} := \frac{\partial}{\partial u_j} C_{ij}(u_i,u_j),$$

$$h_{j|i} := \frac{\partial}{\partial u_i} C_{ij}(u_i,u_j).$$

Given the pair copula $C_{e_a,e_b;D_e}$ which corresponds to the edge $e_a, e_b; D_e$ in a regular vine tree sequence we use the following notation

$$h_{e_a|e_b;D_e}(w_1|w_2) := \frac{\partial}{\partial w_2} C_{e_a,e_b;D_e}(w_1|w_2),$$

$$h_{e_b|e_a;D_e}(w_2|w_1) := \frac{\partial}{\partial w_1} C_{e_b,e_a;D_e}(w_2|w_1).$$

Remark 2.21. *We are defining two versions of the h functions, because as a standard we use ordered notation in the index of the copula. For example, we would use C_{12} instead of C_{21} for the joint distribution of (U_1,U_2), and out of C_{12} we would calculate the conditional distribution functions $C_{1|2}$ and $C_{2|1}$.*

Theorem 2.22. (Recursion for conditional distribution functions)
Let $l \in D$ and $D_{-l} := D \setminus \{l\}$. Then,

$$C_{i|\mathbf{u}_D}(u_i|\mathbf{u}_D) = h_{i|l;D_{-l}}\left(C_{i|\mathbf{u}_{D_{-l}}}\left(u_i|\mathbf{u}_{D_{-l}}\right)|C_{l|\mathbf{u}_{D_{-l}}}\left(u_l|\mathbf{u}_{D_{-l}}\right)\right), \tag{2.7}$$

where for $i, j \notin D$, $i < j$, $h_{i|j;D}(u|v)$ and $h_{j|i;D}(v|u)$ are the h-functions associated with the pair copula $C_{ij;D}$.

Proof. The proof of Theorem 2.22 was first stated in Joe (1996) and is based on the chain rule of differentiation. □

In fact, Theorem 2.22 defines a recursion on the conditional distributions of a regular vine over its tree sequences. Namely, consider uniformly distributed random variables $\mathbf{U}_D$, where $D = \{1, \ldots, p\}$, and an R-vine copula on $\mathbf{U}_D$, $\mathcal{R} = (\mathcal{V}, \mathcal{B}, \Theta)$. To express the conditional distribution $C_{U_i|\mathbf{U}_{D_{-i}}}$ for $i = 1, \cdots p$, recursively following equation (2.7) we need the pair copulas

$$C_{ik_1}(\cdot,\cdot), \text{ and } C_{ik_j;k_1,\ldots k_{j-1}}(\cdot,\cdot) \text{ for } j = 2, \ldots, p-1,$$

where $(k_1, \ldots k_{p-1})$ is a permutation of D_{-i}. These pair copulas are essential in obtaining the required h-functions for the recursion defined by Theorem 2.22. Further, in order for the pair copulas of the form from equation (2.7) to be included in $\mathcal{B}(\mathcal{V})$ we must have the corresponding edges included in the tree sequence $\mathcal{V} = (T_1, \ldots, T_{p-1})$. Namely it must hold that

$$(U_i, U_{k_1}) \in T_1 \quad \text{and} \quad \left(U_i, U_{k_j}|U_{k_1}, \ldots U_{k_{j-1}}\right) \in T_j \quad \text{for } j = 2, \ldots, p-1.$$

This brings us to the following corollary:

Corollary 2.23. *Let $\mathbf{U}_D = (U_1, \ldots, U_p)$, where $D = \{1, \ldots, p\}$, be uniformly distributed random variables. Further, let $\mathcal{R} = (\mathcal{V}, \mathcal{B}(\mathcal{V}), \Theta(\mathcal{B}(\mathcal{V})))$ be an R-vine copula on $\mathbf{U}_D$. Then the conditional distribution $C_{U_i|\mathbf{U}_{D_{-i}}}$ can be derived using Theorem 2.22 and pair copulas from $\mathcal{B}(\mathcal{V})$ if and only if there exists a permutation $\left(U_{k_1}, \ldots U_{k_{p-1}}\right)$ of $\mathbf{U}_{D_{-i}}$ such that the edge $(U_i, U_{k_1}) \in T_1$ and the edges*

$$\left(U_i, U_{k_j}|U_{k_1}, \ldots U_{k_{j-1}}\right) \in T_j \quad \textit{for } j = 2, \ldots, p-1.$$

Example 2.24. *Consider a R-vine copula* $\mathcal{R} = (\mathcal{V}, \mathcal{B}(\mathcal{V}), \Theta(\mathcal{B}(\mathcal{V})))$ *with tree sequence,* $\mathcal{V} = (T_1, T_2, T_3, T_4)$*, defined as in Figure 2.3.*

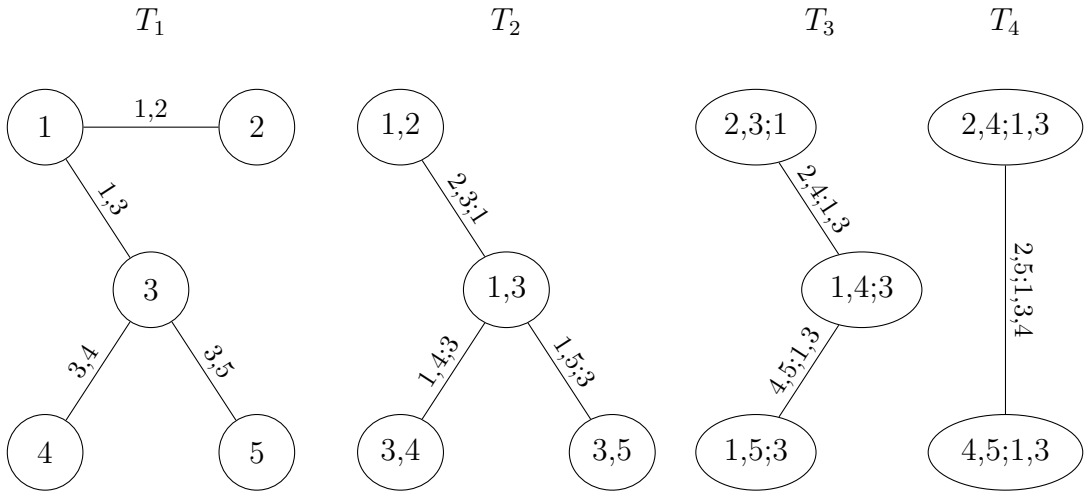

Figure 2.3: 5-dimensional R-vine tree sequence.

We would like to illustrate Corollary 2.23 in this example, so we are interested in expressing the conditional distributions of the copula C_{12345} *using the recursion from Theorem 2.22 and the pair copulas defined in* $\mathcal{B}(\mathcal{V})$*.*

Lets first derive the conditional distribution $C_{2|1345}$ *using the recursion defined in Theorem 2.22*

$$\begin{aligned} C_{2|1345}(u_2|u_1,u_3,u_4,u_5) &= h_{2|5;134}\left(C_{2|134}(u_2|u_1,u_3,u_4)\,|C_{5|134}(u_5|u_1,u_3,u_4)\right) \\ &= h_{2|5;134}\Big(h_{2|4;13}\left(C_{2|13}(u_2|u_1,u_3)\,|C_{4|13}(u_4|u_1,u_3)\right)| \\ &\qquad h_{5|4;13}\left(C_{5|13}(u_5|u_1,u_3)\,|C_{4|13}(u_4|u_1,u_3)\right)\Big) \\ &= h_{2|5;134}\Big(h_{2|4;13}\left(h_{2|3;1}\left(h_{2|1}|h_{3|1}\right)|h_{4|1;3}\left(h_{4|3}|h_{1|3}\right)\right)| \\ &\qquad h_{5|4;13}\left(h_{5|1;3}\left(h_{5|3}|h_{1|3}\right)|h_{4|1;3}\left(h_{4|3}|h_{1|3}\right)\right)\Big). \end{aligned}$$

Note that due to easier fitting in one page, in the very last equation the arguments of the $h_{i|j}$ *for* $i, j = 1, \ldots, 5$ *are not written, so instead of* $h_{i|j}(u_i|u_j)$ *we write* $h_{i|j}$*.*

Now, lets see whether we can derive all the h functions needed to calculate $C_{2|1345}$ *from the copulas defined in* $\mathcal{B}(\mathcal{V})$ *. Starting from the first h function* $h_{2|5;134}$*, to derive it we need the pair copula* $C_{25;134}$*. We can see that the edge*

$$(2,5;1,3,4) \in T_4,$$

which consequently implies that $C_{25;134} \in \mathcal{B}(\mathcal{V})$*.*

Next, for the $h_{2|4;13}$ and $h_{5|4;13}$ we require the copulas $C_{24;13}$ and $C_{45;13}$, respectively. We can see in the third tree of Figure 2.3 that

$$(2,4;1,3) \in T_3 \quad \textit{and} \quad (4,5;1,3) \in T_3,$$

so as these edges exist in the tree sequence, it follows $C_{24;13}, C_{45;13} \in \mathcal{B}(\mathcal{V})$.

For $h_{2|3;1}$, $h_{4|1;3}$ and $h_{5|1;3}$ we require the copulas $C_{23;1}$, $C_{14;3}$ and $C_{15;3}$, and since in the second tree of Figure 2.3 we have that edges

$$(2,3;1), (1,4;3), (1,5;3) \in T_2,$$

thus it follows $C_{23;1}, C_{14;3}, C_{15;3} \in \mathcal{B}(\mathcal{V})$.

Similarly, for $h_{2|1}$ we need C_{12} and edge $(1,2) \in T_1$, for $h_{3|1}$ and $h_{1|3}$ we require C_{13} and edge $(1,3) \in T_1$, thus $C_{12}, C_{13} \in \mathcal{B}(\mathcal{V})$.

For $h_{4|3}$ we need C_{34} and edge $(3,4) \in T_1$, for $h_{5|3}$ we need C_{35} and edge $(3,5) \in T_1$, so also $C_{34}, C_{35} \in \mathcal{B}(\mathcal{V})$.

Therefore, we can conclude that using Theorem 2.22 and the bivariate copulas given in $\mathcal{B}(\mathcal{V})$, we can derive the conditional distribution $C_{2|1345}$. In a similar manner, also $C_{5|1234}$ can be calculated, because of the existence of the edge $(2,5;1,3,4)$ and the copula $C_{2,5;1,3,4}$, and so on.

However, the conditional distributions $C_{1|2345}$, $C_{3|1245}$ and $C_{4|1235}$ cannot be derived using Theorem 2.22 and the copulas in $\mathcal{B}(\mathcal{V})$. Consider the conditional distribution $C_{1|2345}$. We can express it using four different h-functions, namely

$$C_{1|2345}\left(u_1|u_2,u_3,u_4,u_5\right) = \begin{cases} h_{1|2;345}\left(C_{1|345}\left(u_1|u_3,u_4,u_5\right)|C_{2|345}\left(u_2|u_3,u_4,u_5\right)\right) \\ h_{1|3;245}\left(C_{1|245}\left(u_1|u_2,u_4,u_5\right)|C_{3|245}\left(u_3|u_2,u_4,u_5\right)\right) \\ h_{1|4;235}\left(C_{1|235}\left(u_1|u_2,u_3,u_5\right)|C_{4|235}\left(u_4|u_2,u_3,u_5\right)\right) \\ h_{1|5;234}\left(C_{1|234}\left(u_1|u_2,u_3,u_4\right)|C_{5|234}\left(u_5|u_2,u_3,u_4\right)\right) \end{cases}.$$

The h functions $h_{1|2;345}$, $h_{1|3;245}$, $h_{1|4;235}$ and $h_{1|5;234}$ can be derived from the pair copulas $C_{12;345}$, $C_{13;245}$, $C_{14;235}$ and $C_{15;234}$, respectively. However, the edges

$$(1,2;3,4,5), (1,3;2,4,5), (1,4;2,3,5), (1,5;2,3,4) \notin T_i, \quad \textit{for } i = 1,2,3,4,$$

consequently, also for the pair copulas we then have

$$C_{12;345}, C_{13;245}, C_{14;235}, C_{15;234} \notin \mathcal{B}(\mathcal{V}).$$

Therefore, using Theorem 2.22 and the pair copulas in $\mathcal{B}(\mathcal{V})$, one can not derive the conditional distribution $C_{1|2345}$. In a similar matter, the same result holds for $C_{3|1245}$ and $C_{4|1235}$.

2.4 Simulating from C- and D-vines

Theorem 2.25. (Simulating from a multivariate copula)

$$
\begin{aligned}
&\textbf{\textit{First:}} \ \textit{Sample i.i.d.} \ w_j \sim U[0,1], \quad j = 1, \dots, d \\
&\textbf{\textit{Then:}} \ u_1 := w_1 \\
&\qquad u_2 := C_{2|1}^{-1}(w_2|u_1), \\
&\qquad \vdots \\
&\qquad u_d := C_{d|d-1,\cdots,1}^{-1}(w_d|u_{d-1}, \dots, u_1)
\end{aligned}
$$

Proof. The proof can be seen in Czado (2019). □

Now, we explain the general case of simulating from a C-vine copula in d dimensions. The vine tree sequence can be stored using matrix notation, and in this chapter we use the matrix notation from Czado (2019).

Consider a C-vine tree sequence in d dimensions, where the root node of tree T_1 is 1, the root node of tree T_2 is $1, 2, \dots$, of tree T_{d-1} is $d, d-1; 1, \dots, d-2$. Then, M_c will be a $d \times d$ dimensional matrix representation of this vine tree sequence as

$$
M_c = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & \cdots \\
 & 2 & 2 & 2 & 2 & \cdots \\
 & & 3 & 3 & 3 & \cdots \\
 & & & 4 & 4 & \cdots \\
 & & & & 5 & \cdots \\
 & & & & & \cdots
\end{bmatrix}.
$$

To simplify notation in the following part of this section we will use the shortened notation $i:j$, for $i, \dots, j$ when $j > i$ and $\boldsymbol{u}_{i:j}$ for $(u_i, \dots, u_j$.
Then we define the matrix in which the copula parameters are stored. Let Θ be a $d \times d$ dimensional, strict upper triangular matrix with entries $\{\eta_{i,k}\}$ for $i < k, k = 2, \dots, d$, where $\eta_{i,k} := \theta_{i,k;1:i-1}$ defined as:

$$
\Theta := \begin{bmatrix}
\theta_{1,2} & \theta_{1,3} & \theta_{1,4} & \cdots \\
 & \theta_{2,3;1} & \theta_{2,4;1} & \cdots \\
 & & \theta_{3,4;1,2} & \cdots \\
 & & & \cdots
\end{bmatrix}.
$$

Using the matrix Θ we can calculate the conditional distribution functions, stored in a upper triangular $d \times d$ dimensional matrix V with entries $\{v_{i,k}\}$ for $i < k, k = 1, \dots, d$, defined as

$$
V := \begin{bmatrix}
u_1 & u_2 & u_3 & u_4 & \cdots \\
 & C_{2|1}(u_2|u_1) & C_{3|1}(u_3|u_1) & C_{4|1}(u_4|u_1) & \cdots \\
 & & C_{3|1,2}(u_3|u_1, u_2) & C_{4|1,2}(u_4|u_1, u_2) & \cdots \\
 & & & C_{4|1,2,3}(u_4|u_1, u_2, u_3) & \cdots \\
 & & & & \cdots
\end{bmatrix}
$$

Now, we go back to how to calculate the conditional distribution functions. Using Theorem 2.22 in the case of C-vines, we can express the conditional distribution function $C_{i|1:i-k}(u_i|\boldsymbol{u}_{1:i-k})$ as

$$C_{i|1:i-k}(u_i|\boldsymbol{u}_{1:i-k}) = \frac{\partial C_{i,i-k;1:i-k-1}\left(C_{i|1:i-k-1}(u_i|\boldsymbol{u}_{1:i-k-1}), C_{i-k|1:i-k-1}(u_{i-k}|\boldsymbol{u}_{1:i-k-1})\right)}{\partial C_{i-k|1:i-k-1}(u_{i-k}|\boldsymbol{u}_{1:i-k-1})}, \tag{2.8}$$

for $i = 1, \ldots, d$ and $k = 1, \ldots, i-1$.
We use $C_{j|1:j-1}(u_j|\boldsymbol{u}_{1:j-1}) := w_j$, i.e., $v_{jj} = w_j$ for $j = 1, \ldots, d$, from the Theorem 2.25. Then we rewritte expression (2.8) in terms of h functions as

$$C_{i|1:i-k}(u_i|\boldsymbol{u}_{1:i-k}) = h_{i|i-k;1:i-k-1}\left(C_{i|1:i-k-1}(u_i|\boldsymbol{u}_{1:i-k-1})|C_{i-k|1:i-k-1}(u_{i-k}|\boldsymbol{u}_{1:i-k-1})\right).$$

Finally, inverting it for $i = 1, \ldots, d$ and $k = 1, \ldots, i = 1$ we have

$$C_{i|1:i-k}(u_i|\boldsymbol{u}_{1:i-k}) = h^{-1}_{i|i-k;1:i-k-1}\left(C_{i|1:i-k-1}(u_i|\boldsymbol{u}_{1:i-k-1})|C_{i-k|1:i-k-1}(u_{i-k}|\boldsymbol{u}_{1:i-k-1})\right). \tag{2.9}$$

By recursion on equation (2.9), we can calculate the entries of matrix V and obtain a sample $(U_1, \ldots, U_d)$. All of this is summarized in the following algorithm from Stoeber and Czado (2017).

Input: Strictly upper triangular matrix Θ of copula parameters with entries $\eta_{ik} = \theta_{ki;1:k-1}$ for $k < i$ and $i, k = 1, \ldots, d$ for the d-dimensional C-vine.
Output: Sample from the C-vine copula.
begin
  Sample i.i.d $w_i \sim U[0,1]$, $i = 1, \ldots, d$;
  $v_{1,1} := w_1$;
  for $i = 2, \ldots, d$ **do**
    $v_{i,i} := w_i$;
    for $k = i-1, \ldots, 1$ **do**
      $v_{k,i} := h^{-1}_{i|k;1:k-1}(v_{k+1,i}|v_{k,k}, \eta_{k,i})$;
    end
  end
end
return $u_i := v_{1,i}$ *for* $i = 1, \ldots, d$;

Algorithm 1: Sampling from a C-vine copula.

Next, we explain how we can simulate data from a $d-$dimensional D-vine copula. Consider a D-vine tree sequence in d dimensions, such that the ordering of the nodes in the first tree T_1 is $1, \cdots, d$ and it determines all further trees of the vine tree sequence. Then M_d will be a $d \times d$ dimensional matrix representation of this vine tree sequence as

$$M_d = \begin{bmatrix} 1 & 1 & 2 & 3 & 4 & \cdots \\ & 2 & 1 & 2 & 3 & \cdots \\ & & 3 & 1 & 2 & \cdots \\ & & & 4 & 1 & \cdots \\ & & & & 5 & \cdots \\ & & & & & \cdots \end{bmatrix}.$$

Let Θ be a $d \times d$ dimensional, strict upper triangular matrix with entries $\{\eta_{i,k}\}$ for $i > k$, where $\eta_{i,k} := \theta_{i-k,i;i-k+1:i-1}$, for $k = 1, \ldots, d-1$ and $i = k+1, \ldots, d$ defined as:

$$\Theta := \begin{bmatrix} \theta_{1,2} & \theta_{2,3} & \theta_{3,4} & \cdots \\ & \theta_{1,3;2} & \theta_{2,4;3} & \cdots \\ & & \theta_{1,4;2,3} & \cdots \\ & & & \cdots \end{bmatrix}.$$

Now, we define two $d \times d$ dimensional matrices, V_1, V_2 with entries $\left\{v_{i,k}^1\right\}$ and $\left\{v_{i,k}^2\right\}$ for $i \leq k, k = 1, \ldots, d$, as following

$$V_1 := \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & \cdots \\ & C_{2|1}(u_2|u_1) & C_{3|2}(u_3|u_2) & C_{4|3}(u_4|u_3) & \cdots \\ & & C_{3|1,2}(u_3|u_1,u_2) & C_{4|2,3}(u_4|u_2,u_3) & \cdots \\ & & & C_{4|1,2,3}(u_4|u_1,u_2,u_3) & \cdots \\ & & & & \cdots \end{bmatrix},$$

$$V_2 := \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & \cdots \\ & C_{1|2}(u_1|u_2) & C_{2|3}(u_2|u_3) & C_{3|4}(u_3|u_4) & \cdots \\ & & C_{1|2,3}(u_1|u_2,u_3) & C_{2|3,4}(u_2|u_3,u_4) & \cdots \\ & & & C_{1|2,3,4}(u_1|u_2,u_3,u_4) & \cdots \\ & & & & \cdots \end{bmatrix}.$$

Using Theorem 2.22 the conditional distribution function in the case of D-vines, $C_{i|k:i-1}(u_i|\boldsymbol{u}_{k:i-1})$, can be expressed as

$$C_{i|k:i-1}(u_i|\boldsymbol{u}_{k:i-1}) = \frac{\partial C_{i,k;k+1:i-1}\left(C_{i|k+1:i-1}(u_i|\boldsymbol{u}_{k+1:i-1}), C_{k|k+1:i-1}(u_k|\boldsymbol{u}_{k+1:i-1})\right)}{\partial C_{k|k+1:i-1}(u_k|\boldsymbol{u}_{k+1:i-1})}, \tag{2.10}$$

for $i = 3, \ldots, d$ and $k = 2, \ldots, i-1$, i.e., $i > k$. This is the reason why we need to define two matrices for the conditional distribution functions. In contrast to the C-vine simulation here the second argument in equation (2.10) is not automatically calculated. For example, in the case for D-vines we need to calculate both $C_{1|2}$ and $C_{2|1}$ from $C_{1,2}$, while for simulating from a C-vine, we only need to calculate one of $C_{1|2}$ and $C_{2|1}$, depending on the choice of a root node.

Recursive following equation (2.10) we can summarize this procedure in the following algorithm, from Stoeber and Czado (2017).

Input: Strictly upper triangular matrix Θ of copula parameters with entries $\eta_{ik} = \theta_{i-k,i;i-k+1:i-1}$ for $k = 1, \dots, d-1$ and $i = k+1, \dots, d$ for the d-dimensional D-vine.
Output: Sample from the D-vine copula.
begin
  Sample i.i.d $w_i \sim U[0,1]$, $i = 1, \dots, d$;
  $v_{1,1} := w_1$;
  $v_{1,1}^2 := w_1$;
  for $i = 2, \dots, d$ **do**
    $v_{i,i} := w_i$;
    for $k = i-1, \dots, 1$ **do**
      $v_{k,i} := h^{-1}_{i|i-k;i-k+1:i-1}\left(v_{k+1,i}|v^2_{k,i-1}, \eta_{k,i}\right)$;
      if $i < d$ **then**
        $v^2_{k+1,i} := h^{-1}_{i-k|i;i-k+1:i-1}\left(v^2_{k,i-1}|v_{k,i}, \eta_{k,i}\right)$;
      end
    end
    $v^2_{1,i} := v_{1,i}$;
  end
end
return $u_i := v_{1,i}$ *for* $i = 1, \dots, d$;

Algorithm 2: Sampling from a D-vine copula.

2.5 Estimation in regular vine copulas

Let there be given a i.i.d. d-dimensional sample of size n from a specified regular vine copula with parametric pair copula families, i.e.

$$\mathbf{u} := (\boldsymbol{u}_1, ..., \boldsymbol{u}_n)^T \quad \text{where} \quad \boldsymbol{u}_k := (u_1^{(k)}, \dots, u_d^{(k)}) \quad \text{for} \quad k = 1, \dots, n.$$

Let the parameter(s) of the pair copula $C_{i,i+j;i+1,\dots,i+j-1}$ are denoted by $\boldsymbol{\theta}_{i,i+j;i+1,\dots,i+j-1}$ for $j = 1, \dots, d-1$ and $i = 1, \dots, d-j$ and let all the parameters of the vine copula be collected in $\boldsymbol{\theta}$. Then the likelihood of a parametric D-vine and C-vine copula, using Theorems 2.16 and 2.17, will be given as:

Definition 2.26. (Likelihood of D- and C-vines)
The likelihood of a parametric D-vine is given as

$$L(\boldsymbol{\theta}, \boldsymbol{u}) = \prod_{k=1}^{n} \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} c_{i,i+j;i+1,\dots,i+j-1} \Big(C_{i|i+1,\dots,i+j-1}(u_i^{(k)}|\boldsymbol{u}^{(k)}_{i+1,\dots,i+j-1}), \\ C_{i+j|i+1,\dots,i+j-1}(u^{(k)}_{i+j}|\boldsymbol{u}^{(k)}_{i+1,\dots,i+j-1}) \Big).$$

The likelihood of a parametric C-vine is given as

$$L(\boldsymbol{\theta}, \boldsymbol{u}) = \prod_{k=1}^{n} \prod_{j=1}^{d-1} \prod_{i=1}^{d-j} \times c_{j,i+j;1,\ldots,j-1} \left(C_{i|1,\ldots,j-1}(u_i^{(k)}|\boldsymbol{u}_{1,\ldots,j-1}^{(k)}), C_{i+j|1,\ldots,j-1}(u_{i+j}^{(k)}|\boldsymbol{u}_{1,\ldots,j-1}^{(k)}) \right).$$

Having the likelihood defined, now we proceed with estimating the regular vine copula.

As seen in Definition 2.5 an R vine copula can be identified by the triplet $(\mathcal{V}, \mathcal{B}, \Theta)$. In our case let the vine tree structure $\mathcal{V}$ be given. In this section, we discuss how we can estimate the copula families $\mathcal{B}$ and the copula parameters Θ for each pair copula in the construction of the vine copula.
First, we consider the selection of copula family and parameters for pair copulas in the first tree $T_1 \in \mathcal{V}$. Let $e = (a_e, b_e)$ be an arbitrary edge in T_1. For the pair copula associated with the edge e we consider the copula data $u_{a_e}^{(k)}$ and $u_{b_e}^{(k)}$ for $k = 1, \ldots, n$. Let B_e be the set of possible parametric bivariate copula families for the edge e. For each element of B_e, we use the copula data $u_{a_e}^{(k)}$ and $u_{b_e}^{(k)}$ to fit a copula, either by maximazing the likelihood estimate only or including the AIC or BIC penalizations, with an associated parameter $\hat{\boldsymbol{\theta}}_{\boldsymbol{e}}$. In a similar matter, we estimate all the pair copulas in T_1.
Next, for the copula family selection and parameter estimation for the edges in trees T_i for $i > 1$ we use a sequential estimation approach. For an edge $e = (a_e, b_e; D_e)$ in tree T_i for $i > 1$, we have pseudo-observations for the bivariate copula distribution of (U_{a_e}, U_{b_e}) given U_{D_e} available, $\hat{u}_{a_e|D_e}^{(k)}$ and $\hat{u}_{b_e|D_e}^{(k)}$ for $k = 1, \ldots, n$ defined as:

$$u_{a_e|D_e}^{(k)} := C_{a_e|D_e}(u_{a_e}^{(k)}|\boldsymbol{u}_{D_e}^{(k)}) = h_{a_e|D_e}(u_{a_e}^{(k)}|\boldsymbol{u}_{D_e}^{(k)}),$$

$$u_{b_e|D_e}^{(k)} := C_{b_e|D_e}(u_{b_e}^{(k)}|\boldsymbol{u}_{D_e}^{(k)}) = h_{b_e|D_e}(u_{b_e}^{(k)}|\boldsymbol{u}_{D_e}^{(k)}).$$

Again consider a set of possible parametric bivariate copula families for the edge e, B_e. Using the pseudo data, we fit a copula, in the same matter as before, with an associated parameter $\hat{\boldsymbol{\theta}}_{\boldsymbol{e}}$. Similarly, we continue estimating all the pair copulas in T_i. Using the selected pair copulas in tree T_i, we estimate the pseudo observations needed for estimating the pair copula families and parameters in tree T_{i+1}.

Chapter 3

Quantile regression

3.1 Introduction

In this chapter we aim to provide a basic overview of the idea and the concepts behind quantile regression, a method disregarded up until the work of Koenker and Bassett Jr (1978). The basic ideas on quantile regression can be traced back to the mid-18th century and thus it predates the earliest work on the subject of least square estimation. However, due to the extremely appealing computational tractability and the optimality in case of a normally distributed noise of the least square methods they have influenced much of the applied statistics.

Intuitively one can compare the setup of quantile regression to linear regression in the sense that as linear regression provides models for estimation and inference about the conditional mean functions, quantile regression provides equivalent statistical tools for the full range of conditional quantile functions. The central and special case of quantile regression estimators is the median regression estimator, which is obtained in a similar framework as the linear regression estimators, with the difference that instead of minimizing the sum of squared errors one has to minimize a sum of absolute errors instead of squared errors. The other conditional quantile functions are obtained by minimizing an asymmetrically weighted sum of absolute errors.

> "What the regression curve does is give a grand summary for the averages of the distributions corresponding to the set of x's. We could go further and compute several different regression curves corresponding to the various percentage points of the distributions and thus get a more complete picture of the set. Ordinarily this is not done, and so regression often gives a rather incomplete picture. Just as the mean gives an incomplete picture of a single distribution, so the regression curve gives a correspondingly incomplete picture for a set of distributions." by Mosteller and Tukey (1977).

Despite the similarities between linear regression and quantile regression there are also substantial differences. First, as Mosteller and Tukey (1977) suggest, the obtained estimates of the conditional mean functions are rarely satisfactory results

themselves and computing a range of estimates for the conditional quantile functions will inevitably provide a more complete picture of the stochastic relationships among the random variables. On the other hand, while quantile regression gives a better insight it comes with the price of increased computational intensity compared to linear regression.

Nowadays quantile regression has found its way to a broad range of application settings and emerged as a complementary method to linear regression in many fields, from biostatistics to econometrics. In the rest of the chapter we follow Koenker (2005).

3.2 Quantiles and quantile regression

Before we go on to formal defining quantile regression we first introduce the concept of quantiles and ranks. Formally, given any univariate random variable X, it may be characterized by its distribution function

$$F(x) = P(X \leq x)$$

on the contrary, the "inverse" function of F, for any $0 < \alpha < 1$, is given as

$$F^{-1}(\alpha) = \inf\{x : F(x) \geq \alpha\} =: q_\alpha(x)$$

and the αth quantile of X is defined as the quantity $q_\alpha(x)$. To illustrate the concept of quantiles more intuitively we consider a simple example of a group of students taking an exam. A specific student scores the αth quantile of the exam distribution if he performs better then the proportion α of all students in the exam and consequently worse then the $(1-\alpha)$ proportion of all students. More specifically, the median, denoted by $q_{0.5}$, of the exam distribution is the student who performed better then half of the other students and worse then the other half.

Quantiles can also be defined from another perspective which turns out to be fundamental to quantile regression. Namely, quantiles arise from a rather simple optimization problem. Assume we want to solve a simple decision problem: a point estimate for a random variable X with distribution function F. We describe the loss function as a piecewise linear function, also know as check loss,

$$\rho_\alpha(x) = x(\alpha - I(x < 0)) = \begin{cases} x\alpha & x \geq 0 \\ x(\alpha - 1) & x < 0 \end{cases}$$

for some $\alpha \in (0, 1)$.

Our goal is to find $\hat{x}$ that will minimize the expected loss function:

$$\arg\min_{\hat{x}} E[\rho_\alpha(X - \hat{x})]$$

which can be further written down as

$$
\begin{aligned}
E[\rho_\alpha(X-\hat{x})] =& E\left[\alpha(X-\hat{x}) - I(X-\hat{x}<0)(X-\hat{x})\right] \\
=& \int_{-\infty}^{\infty} \left(\alpha(x-\hat{x}) - I(x-\hat{x}<0)(x-\hat{x})\right) f(x)\,dx \\
=& \int_{-\infty}^{\hat{x}} (\alpha-1)(x-\hat{x}) f(x)\,dx + \int_{\hat{x}}^{\infty} \alpha(x-\hat{x}) f(x)\,dx
\end{aligned}
$$

Differentiating $E[\rho_\alpha(X-\hat{x})]$ with respect to $\hat{x}$, under certain regularity conditions and following the formula for differentiation under the integral sign gives us

$$
\begin{aligned}
\frac{\partial}{\partial\hat{x}} E[\rho_\alpha(X-\hat{x})] =& \frac{\partial}{\partial\hat{x}} \int_{-\infty}^{\hat{x}} (\alpha-1)(x-\hat{x}) f(x)\,dx + \frac{\partial}{\partial\hat{x}} \int_{\hat{x}}^{\infty} \alpha(x-\hat{x}) f(x)\,dx \\
=& \frac{\partial}{\partial\hat{x}} \hat{x}(\hat{x}-\hat{x}) f(\hat{x}) + \int_{-\infty}^{\hat{x}} \frac{\partial}{\partial\hat{x}} (\alpha-1)(x-\hat{x}) f(x)\,dx \\
& - \frac{\partial}{\partial\hat{x}} \hat{x}(\hat{x}-\hat{x}) f(\hat{x}) + \int_{\hat{x}}^{\infty} \frac{\partial}{\partial\hat{x}} (\alpha(x-\hat{x}) f(x))dx \\
=& (1-\alpha) \int_{-\infty}^{\hat{x}} f(x)\,dx - \alpha \int_{\hat{x}}^{\infty} f(x)\,dx \\
=& (1-\alpha) \int_{-\infty}^{\hat{x}} dF(x) - \alpha \int_{\hat{x}}^{\infty} dF(x) \\
=& F(\hat{x}) - \alpha
\end{aligned}
$$

Now since the distribution function is monotone, any element of $\{x : F(x) = \alpha\}$ minimizes the expected loss. If the solution is unique, then $F(\hat{x}) - \alpha = 0$, thus $\hat{x} = F^{-1}(\alpha)$, otherwise we have an interval of αth quantiles, from which the smallest element is chosen, therefore the minimizer of this function is indeed how we defined our target function $q_\alpha(x) = inf\{x : F(x) \geq \alpha\}$.
Now, we continue with quantile regression, for which we first define conditional quantile function.

Definition 3.1. (Conditional quantile function for a continuous random variable) *The conditional quantile function for $\alpha \in (0,1)$, for a continuous response variable Y given the outcome of some predictor variables $X_1, ..., X_p$ for some number of predictors $p \geq 1$ is*

$$
q_\alpha(x_1, ..., x_p) := F^{-1}_{Y|X_1,...,X_p}(\alpha | X_1 = x_1, ..., X_p = x_p).
$$

Regression methods in general aim to characterize the behaviour of the response Y as a deterministic function of the predictors values $x_1, ..., x_p$, of the random $X_1, ..., X_p$, and some additive random noise ϵ, i.e. $Y = f(x_1, ..., x_p) + \epsilon$. Depending on the target function, we define a corresponding loss function, such that the the minimizer of the expected loss function is the target function. For example, if the target function is the conditional mean $E[Y|X_1 = x_1, ..., X_p = x_p]$ the corresponding loss function is the quadratic loss:

$$
L(y, f(x_1, ..., x_p)) = (y - f(x_1, ..., x_p))^2.
$$

The regression function minimizing the expected quadratic loss is indeed the conditional mean, i.e.

$$\underset{f(x_1,...,x_p)}{\arg\min}\, E[(y - f(x_1, ..., x_p))^2] = E[Y|X_1 = x_1, ..., X_p = x_p].$$

Another example is if the target function is the conditional quantile function, we defined above. As a loss function we use the check loss function, defined as

$$L\left(y, f(x_1, ..., x_p)\right) = \rho_\alpha\left(y - f(x_1, ..., x_p)\right).$$

The regression function minimizing the expected check loss is indeed the conditional quantile function, i.e.

$$\underset{f(x_1,...,x_p)}{\arg\min}\, E[\rho_\alpha\left(y - f(x_1, ..., x_p)\right)] = q_\alpha\left(x_1, ..., x_p\right),$$

for all $\alpha \in (0, 1)$, as proved in Koenker (2005).

3.3 Linear quantile regression

Linear quantile regression is the first quantile regression model introduced in Koenker and Bassett Jr (1978). In a similar manner as ordinary linear regression, they assume that the conditional quantiles to be linear in the predictors.

Definition 3.2. (Linear quantile regression model)
Let Y be a response variable depending on the set of predictors $X_1, ..., X_p$ for some $p \geq 1$. For every α the conditional quantiles of Y given $X_1 = x_1, ..., X_p = x_p$ are given as

$$Q_{Y|X_1,...,X_p}\left(\alpha|x_1, ..., x_p\right) := \beta_0(\alpha) + \sum_{j=1}^{p} \beta_j(\alpha)x_j.$$

Further, we can estimate the coefficients $\hat{\boldsymbol{\beta}}(\alpha) = (\hat{\beta}_0(\alpha), \hat{\beta}_1(\alpha), ..., \hat{\beta}_p(\alpha)) \in \mathbb{R}^{p+1}$ for every $\alpha \in (0, 1)$ by minimization of the empirical risk, given as an empirical mean of the check losses.

Definition 3.3. (Parameter estimation for linear quantile regression)
Let $\boldsymbol{y} = (y_1, ..., y_n)^t$, $\boldsymbol{x_i} = (x_{1i}, ..., x_{ni})^t \in \mathbb{R}^n$ be observations of the response Y and the predictors $X_1, ..., X_p$. For every $\alpha \in (0, 1)$, the estimated regression coefficients are defined as

$$\hat{\boldsymbol{\beta}}(\alpha) := \underset{\beta(\alpha)}{\arg\min} \sum_{i=1}^{n} \rho_\alpha\left(y_i - \boldsymbol{x_i}\boldsymbol{\beta}(\alpha)\right),$$

where $\boldsymbol{x_i} := (x_{i1}, ..., x_{ip}.)$

This optimization problem might be solved by linear programming techniques, as the simplex method or interior point methods, for example discussed in Vanderbei et al. (2015).

However, linear quantile regression has drawn criticism for a few of its properties/assumptions. A major drawback that Koenker himself points out is that the regression lines of different quantile levels may cross, since they may have differing slopes, also known as quantile crossing.

Definition 3.4. (Quantile crossing)
Let $\hat{Q}_{Y|X_1,...,X_p}(\cdot|\cdot)$ be the estimated quantile function for a response variable Y given the predictors $X_1, ..., X_p$. A quantile crossing in a point $\boldsymbol{x} := (1, x_1, ..., x_p)^t \in \mathbb{R}^{p+1}$ occurs, if

$$\hat{q}_{\alpha_1}(x_1, ..., x_p) > \hat{q}_{\alpha_2}(x_1, ..., x_p)$$

for some $0 < \alpha_1 < \alpha_2$.

Another drawback is its strong assumption of linear conditional quantiles. Bernard and Czado (2015) highlight that the linearity assumption is almost never fulfilled. As an example, assuming normal margins the only copula resulting in linear conditional quantiles is the Gaussian copula, which is very restrictive.

In addition to this, this method also has other typical shortfalls of linear models, such as multicollinearity and the potential need to include transformed variables and interactions.

Part II

C- and D-vine based quantile regression

Chapter 4

Vine based quantile regression

For the purpose of quantile regression one is interested in modeling the response variable Y given the outcome of a set of predictors $X_1, X_2, \ldots, X_p$, where $Y \sim F_Y$ and $X_j \sim F_{X_j},\ j = 1, \ldots, p$. More precisely, we want to estimate the conditional quantile function

$$q_\alpha(x_1, \ldots, x_p) := F^{-1}_{Y|X_1,\ldots,X_p}(\alpha|x_1, \ldots, x_p) \quad \text{for } \alpha \in (0,1). \tag{4.1}$$

In contrast to linear quantile regression, which models the conditional quantile function q_α directly, the aim of regular vine based quantile regression models is to estimate the conditional distribution of Y given $X_1, X_2, \ldots, X_p$. Namely, we are interested in estimating the inverse of the right hand side of (4.1) defined as

$$F_{Y|X_1,\ldots,X_p}(y|x_1, \ldots, x_p) = P(Y \leq y|X_1 = x_1, \ldots, X_p = x_p). \tag{4.2}$$

We propose, similarly as Kraus and Czado (2017), to estimate this conditional distribution with the help of C-vine and D-vine copulas. For this we need to select the C-vine (D-vine) tree sequence, the parametric pair copula families and their parameters. The parametric pair copulas are only defined on the unit square and therefore we need to transform all variables to the u-scale. This is accomplished by defining the random probability integral transforms (PIT) as

- $V := F_Y(Y)$ and $U_j := F_{X_j}(X_j)$,

together with their corresponding observed PIT values as

- $v := F_Y(y)$ and $u_j := F_{X_j}(x_j)$.

Now we can rewrite equation (4.2) as

$$\begin{aligned} F_{Y|X_1,\ldots,X_p}(y|x_1, \ldots, x_p) = \\ &= P\left(F_Y(Y) \leq F_Y(y) \,|F_{X_1}(X_1) = F_{X_1}(x_1), \ldots, F_{X_p}(X_p) = F_{X_p}(x_p)\right) \\ &= P(V \leq v|U_1 = u_1, \ldots, U_p = u_p) \\ &= C_{V|U_1,\ldots,U_p}(v|u_1, \ldots u_p). \end{aligned}$$

Here $C_{V|U_1,\ldots,U_p}$ is the conditional distribution of V given $U_j = u_j$ and $C_{V,U_1,\ldots,U_p}$ is defined as the $(p+1)$-dimensional copula associated with the joint distribution of $(Y, X_1, \ldots, X_p)$. Then the desired conditional quantile function can be derived by inverting the expression, i.e

$$F^{-1}_{Y|X_1,\ldots,X_p}(\alpha|x_1,\ldots,x_p) = F^{-1}_Y\left(C^{-1}_{V|U_1,\ldots,U_p}(\alpha|u_1,\ldots u_p)\right).$$

Assuming the margins F_Y, F_{X_j}, for $j = 1,\ldots,p$ are known, we can obtain an estimate of the conditional quantile function $F_{Y|X_1,\ldots,X_p}$ by only estimating the copula $C_{V,U_1,\ldots,U_p}$.
Let us denote with $\hat{C}_{V|U_1,\ldots,U_p}$ the estimate of the conditional distribution $C_{V|U_1,\ldots,U_p}$ based on $C_{V,U_1,\ldots,U_p}$. Then the estimate $\hat{q}_\alpha$ of the conditional quantile function q_α can then be expressed as

$$\hat{q}_\alpha(x_1,\ldots,x_p) = F^{-1}_Y\left(\hat{C}^{-1}_{V|U_1,\ldots,U_p}(\alpha|u_1,\ldots u_p)\right).$$

In fact, $C_{V,U_1,\ldots,U_p}$ can be any multivariate copula which is a $p+1$ dimensional function to be estimated (and thus till now we have not reduced the complexity of the initial problem). Estimating $C_{V,U_1,\ldots,U_p}$ can be made more tractable by focusing on a specific class of copulas. For the purpose of quantile regression we would like to restrict $C_{V,U_1,\ldots,U_p}$ to a flexible class of vine copulas, which are able to capture asymmetric dependencies, heavy tails and tail dependencies between the variables. As mentioned before, we chose to restrict $C_{V,U_1,\ldots,U_p}$ to a subclass of C-vines and D-vines. This way we still obtain a class of flexible vine copulas, since each bivariate copula of the pair-copula construction can be modeled separately and the order of predictors in the tree sequence remains a free parameter.

It is important to remember that while we focus on estimating the copula $C_{V,U_1,\ldots,U_p}$, our quantile regression target function remains the conditional distribution $C_{V|U_1,\ldots,U_p}$. For regular vine copulas analytical expressions for conditional distributions in terms of pair copulas are available as a result of Theorem 2.22. Ideally we would like to use this result to calculate the conditional distribution $C_{V|U_1,\ldots,U_p}$. However, a closer look at Theorem 2.22 reveals that the conditional distributions are provided as a recursion indexed by regular vine tree sequences. Further, calculating $C_{V|U_1,\ldots,U_p}$ from the recursion of Theorem 2.22 is only feasible if the regular vine tree sequence of the copula $C_{V,U_1,\ldots,U_p}$ is chosen appropriately. A more detailed view on this topic is provided by the following theorem:

Theorem 4.1. *The conditional distribution $C_{V|U_1,\ldots,U_p}$ can be derived using the recursion defined in Theorem 2.22 if the copula $C_{V,U_1,\ldots,U_p}$ with tree sequence $\mathcal{V} = \{T_1,\ldots,T_p\}$ is*

a) a D-vine copula such that V is a leaf in T_1,

b) a C-vine copula such that for every $T_i \in \mathcal{V}$, $i = 1,\ldots,p-1$ it holds that the node containing the response variable V in the conditioned set is not the root node of T_i.

Proof. Suppose $C_{V,U_1,\ldots,U_p}$ is a D-vine copula with tree sequence $\mathcal{V} = \{T_1, \ldots, T_p\}$, such that V is a leaf in T_1. Since $\mathcal{V}$ is a D-vine tree sequence there exists a permutation $(k_1, \ldots, k_p)$ of $(1, \ldots, p)$ such that the order of nodes in T_1 is

$$V - U_{k_1} - \ldots - U_{k_p}. \tag{4.3}$$

Further, equation (4.3) and the properties of a D-vine imply that

$$(V, U_{k_1}) \in T_1 \quad \text{and} \quad \left(V, U_{k_j} | U_{k_1}, \ldots U_{k_{j-1}}\right) \in T_j \quad \text{for } j = 2, \ldots, p. \tag{4.4}$$

Similarly, if $C_{V,U_1,\ldots,U_p}$ is a C-vine copula such that for every $T_i \in \mathcal{V}$, $i = 1, \ldots, p-1$ it holds that the node containing the response variable V in the conditioned set is not the root node of T_i, there exists a permutation $(k_1, \ldots, k_p)$ of $(1, \ldots, p)$ such that $U_{k_1}U_{k_2}$ is the root node in T_1 and

$$U_{k_j}U_{k_{j+1}}; U_{k_1}, \ldots, U_{k_{j-1}} \text{ is the root node of tree } T_j \text{ for } j = 2, \ldots p.$$

Consequently, this implies also that $\mathcal{V}$ of the C-vine copula $C_{V,U_1,\ldots,U_p}$ will satisfy equation (4.4).
Thus, from Corollary 2.23 follows directly that, since $C_{V,U_1,\ldots,U_p}$ satisfies (4.4), the conditional distribution $C_{V|U_1,\ldots,U_p}$ can be expressed using the recursion from Theorem 2.22. □

Example 4.2. *Let $\mathcal{C}_D = (\mathcal{V}_D, \mathcal{B}_D, \Theta_D)$ be a D-vine with node order in the first tree given as $V - U_1 - U_2 - U_3$. Then the conditional distribution of V given (U_1, U_2, U_3), using Theorem 2.22, can be recursively expressed as*

$$\begin{aligned}
C_{V|U_1,U_2,U_3}(v|u_1,u_2,u_3) &= h_{V|U_3;U_1,U_2}\left(C_{V|U_1,U_2}(v|u_1,u_2)\,|C_{U_3|U_1,U_2}(u_3|u_1,u_2)\right) \\
&= h_{V|U_3;U_1,U_2}\left(h_{V|U_2;U_1}\left(C_{V|U_1}(v|u_1)\,|C_{U_2|U_1}(u_2|u_1)\right)\,| \right. \\
&\qquad \left. h_{U_3|U_1;U_2}\left(C_{U_3|U_2}(u_3|u_2)\,|C_{U_1|U_2}(u_1|u_2)\right)\right) \\
&= h_{V|U_3;U_1,U_2}\left(h_{V|U_2;U_1}\left(h_{V|U_1}(v|u_1)\,|h_{U_2|U_1}(u_2|u_1)\right)\,| \right. \\
&\qquad \left. h_{U_3|U_1;U_2}\left(h_{U_3|U_2}(u_3|u_2)\,|h_{U_1|U_2}(u_1|u_2)\right)\right).
\end{aligned}$$

A simple inversion then yields the desired conditional quantile function

$$\begin{aligned}
&C^{-1}_{V|U_1,U_2,U_3}(\alpha|u_1,u_2,u_3) = \\
&h^{-1}_{V|U_1}\left\{h^{-1}_{V|U_2;U_1}\left[h^{-1}_{V|U_3;U_1,U_2}\left(\alpha|h_{U_3|U_1;U_2}\left(h_{U_3|U_2}(u_3|u_2)\,|h_{U_1|U_2}(u_1|u_2)\right)\right)|h_{U_2|U_1}(u_2|u_1)\right]|u_1\right\}.
\end{aligned}$$

Example 4.3. *Let $\mathcal{C}_C = (\mathcal{V}_C, \mathcal{B}_C, \Theta_C)$ be a C-vine with U_1 as the root node in the first tree, U_2U_1 the root node in the second tree and $U_3U_2|U_1$ root node in the third tree. Similarly as in Example 4.2, we can obtain the conditional distribution of V*

given (U_1, U_2, U_3) *as*

$$\begin{aligned} C_{V|U_1,U_2,U_3}(v|u_1,u_2,u_3) &= h_{V|U_3;U_1,U_2}\left(C_{V|U_1,U_2}(v|u_1,u_2)\,|C_{U_3|U_1,U_2}(u_3|u_1,u_2)\right) \\ &= h_{V|U_3;U_1,U_2}\left(h_{V|U_2;U_1}\left(C_{V|U_1}(v|u_1)\,|C_{U_2|U_1}(u_2|u_1)\right)|\right. \\ &\qquad \left. h_{U_3|U_2;U_1}\left(C_{U_3|U_1}(u_3|u_1)\,|C_{U_2|U_1}(u_2|u_1)\right)\right) \\ &= h_{V|U_3;U_1,U_2}\left(h_{V|U_2;U_1}\left(h_{V|U_1}(v|u_1)\,|h_{U_2|U_1}(u_2|u_1)\right)|\right. \\ &\qquad \left. h_{U_3|U_2;U_1}\left(h_{U_3|U_1}(u_3|u_1)\,|h_{U_2|U_1}(u_2|u_1)\right)\right). \end{aligned}$$

Again by inverting the conditional distribution we get the conditional quantile function as:

$$\begin{aligned} &C^{-1}_{V|U_1,U_2,U_3}(\alpha|u_1,u_2,u_3) = \\ &h^{-1}_{V|U_1}\left\{h^{-1}_{V|U_2;U_1}\left[h^{-1}_{V|U_3;U_1,U_2}\left(\alpha|h_{U_3|U_2;U_1}\left(h_{U_3|U_1}(u_3|u_1)\,|h_{U_2|U_1}(u_2|u_1)\right)\right)|h_{U_2|U_1}(u_2|u_1)\right]|u_1\right\}. \end{aligned}$$

In the Examples 4.2 and 4.3, taken from Czado (2019), we can see the consequences of Corollary 4.1. In neither the C-vine $\mathcal{C}_C$ nor in the D-vine $\mathcal{C}_D$ we can derive the conditional distribution $C_{U_1|V,U_2,U_3}$ using only the pair copulas defined by this specific tree sequences $\mathcal{V}_C$ or $\mathcal{V}_D$. For $C_{U_1|V,U_2,U_3}$ the first step of the recursion from Theorem 2.22 would require an h-function of the form $h_{U_1|W_1;W_2,W_3}$ where (W_1, W_2, W_3) is any permutation of (V, U_2, U_3). In order to derive $h_{U_1|W_1;W_2,W_3}$ we need the pair copula $C_{U_1,W_1;W_2,W_3}$. However, a pair copula of the form $C_{U_1,W_1;W_2,W_3}$ is not included in neither $\mathcal{B}_C$ nor $\mathcal{B}_D$.

Following the results of Corollary (4.1) we define the C-vine and D-vine classes for quantile regression as:

Definition 4.4. (C-vine and D-vine classes for quantile regression)
Let $V, U_1, \ldots, U_p$ *be uniformly distributed random variables. Then the copula* $\mathcal{C} = (\mathcal{V}, \mathcal{B}, \Theta)$ *belongs to the class of C-vine copulas for quantile regression on p predictors, denoted as* $\mathfrak{C}^p$*, if*

(i) $\mathcal{C}$ *is a* $p+1$ *dimensional C-vine copula on the elements:* $V, U_1, \ldots, U_p$*,*

(ii) for every $T_i \in \mathcal{V}$, $i = 1, \ldots, p-2$ *it holds that the node containing the response variable* V *in the conditioned set is not the root node of* T_i*.*

On the other side, $\mathcal{C}$ *belongs to the class of D-vine copulas for quantile regression on p predictors, denoted as* $\mathfrak{D}^p$*, if*

(i) $\mathcal{C}$ *is a* $p+1$ *dimensional D-vine copula on the elements:* $V, U_1, \ldots, U_p$*,*

(ii) V *is a leaf node in the first tree of* $\mathcal{V}$*.*

For the sake of simplicity, in the rest of this chapter we suppose without loss of generality that if $\mathcal{C} \in \mathfrak{D}^p$ then V is always the left leaf node of the first tree in $\mathcal{V}$.

Definition 4.5. (Order of C- and D-vine)
Let $\mathcal{C} = (\mathcal{V}, \mathcal{B}, \Theta)$ be a vine copula.

(i) Let $\mathcal{C} \in \mathfrak{C}^p$, then we say that the C-vine $\mathcal{C}$ has order

$$\mathcal{O}_C(\mathcal{C}) = \left(V, U_{i_1}, \ldots, U_{i_p}\right),$$

if U_{i_1} is the root node in T_1, $U_{i_2}U_{i_1}$ is the root node in T_2, and $U_{i_k}U_{i_{k-1}}|U_{i_1}, \ldots, U_{i_{k-2}}$ is the root node in T_k for $k = 3, \ldots, p-1$.

(ii) Let $\mathcal{C} \in \mathfrak{D}^p$ then we say that the D-vine $\mathcal{C}$ has order

$$\mathcal{O}_D(\mathcal{C}) = \left(V, U_{i_1}, \ldots, U_{i_p}\right),$$

if V is the first node of T_1 and U_{i_k} is the $(k+1)-th$ node of T_1.

Corollary 4.6. *Let $\mathcal{C} = (\mathcal{V}, \mathcal{B}, \Theta)$ belong to the class $\mathfrak{C}^p$ or the class $\mathfrak{D}^p$ with order $\mathcal{O}(\mathcal{C}) = (V, U_1, \ldots, U_p)$, then we can express the conditional distribution of V given U_1 as*

$$C_{V|U_1}(v|u_1) = h_{V|U_1}(v|u_1).$$

The conditional distributions of V given $U_1, \ldots, U_k$ for $k = 2, \ldots, p$ can be expressed with the recursion

$$\begin{aligned} C_{V|U_1,\ldots,U_k}(v|u_1 \ldots, u_k) =& h_{V|U_k;U_1,\ldots,U_{k-1}}\left(C_{V|U_1,\ldots,U_{k-1}}(v|u_1, \ldots, u_{k-1}) \,|\right. \\ & \left. C_{U_k|U_1,\ldots,U_{k-1}}(u_k|u_1, \ldots, u_{k-1})\right). \end{aligned}$$

Proof. Since $\mathcal{C} \in \mathfrak{D}^p$ (or $\mathfrak{C}^p$) with order $\mathcal{O}(\mathcal{C}) = (V, U_1, \ldots, U_p)$ it holds that

$$(V, U_1) \in T_1 \quad \text{and} \quad (V, U_j|U_1, \ldots U_{j-1}) \in T_j \quad \text{for } j = 2, \ldots, p.$$

Therefore,

$$C_{VU_1}(\cdot, \cdot) \in \mathcal{B}(\mathcal{V}) \quad \text{and} \quad C_{VU_1|U_1,\ldots U_{j-1}}(\cdot, \cdot) \in \mathcal{B}(\mathcal{V}) \quad \text{for } j = 2, \ldots, p.$$

Thus, $C_{V|U_1}(v|u_1)$ can be derived using the pair copula C_{V,U_1}. Similarly, using Theorem 2.22 one can also recursively obtain $C_{V|U_1,\ldots,U_j}(v|u_1, \ldots, u_j)$ using the pair copulas $C_{VU_j|U_1,\ldots U_{j-1}}$. □

With the conditional distributions defined in Corollary 4.6, we can formulate the conditional density $c_{V|U_1,\ldots,U_p}(v|u_1, \ldots, u_p)$ as

Corollary 4.7. *Let $\mathcal{C} = (\mathcal{V}, \mathcal{B}, \Theta)$ belong to the class $\mathfrak{C}^p$ or the class $\mathfrak{D}^p$ with order $\mathcal{O}(\mathcal{C}) = (V, U_1, \ldots, U_p)$. Then, the conditional density $c_{V|U_1,\ldots,U_p}$ can be expressed as:*

$$\begin{aligned} c_{V|U_1,\ldots,U_p}(v|u_1, \ldots, u_p) =& c_{V,U_1}(v, u_1) \\ & \times c_{V,U_2|U_1}\left(h_{V|U_1}(v|u_1), h_{U_2|U_1}(u_2|u_1)\right) \\ & \times \prod_{i=3}^{p} c_{V,U_i|U_1,\ldots,U_{i-1}}\left(C_{V|U_1,\ldots,U_{i-1}}(v|u_1, \ldots, u_{i-1}),\right. \\ & \left. C_{U_i|U_1,\ldots,U_{i-1}}(u_i|u_1, \ldots, u_{i-1})\right). \end{aligned} \tag{4.5}$$

Proof. If $\mathcal{C} \in \mathfrak{C}^p$, then, from Theorem 2.17 we can write the density of $\mathcal{C}$

$$\begin{aligned} c_{V,U_1,\ldots,U_p} &= \left[\prod_{j=1}^{p} c_{V,U_j;U_1,\ldots,U_{j-1}}\right] \cdot \left[\prod_{i=1}^{p-1}\prod_{j=1}^{p-i} c_{U_j,U_{j+i};U_{j+1},\ldots,U_{j+i-1}}\right] \\ &= \left[\prod_{j=1}^{p} c_{V,U_j;U_1,\ldots,U_{j-1}}\right] \cdot c_{U_1,\ldots,U_p}. \end{aligned}$$

Thus, we have

$$\begin{aligned} c_{V|U_1,\ldots,U_p} =& \frac{c_{V,U_1,\ldots,U_p}}{c_{U_1,\ldots,U_p}} \\ =& \prod_{j=1}^{p} c_{V,U_j;U_1,\ldots,U_{j-1}}, \end{aligned} \tag{4.6}$$

which is equivalent to (4.5). On the other hand, if $\mathcal{C} \in \mathfrak{D}^p$, then, from Theorem 2.16 we can write the density of $\mathcal{C}$

$$\begin{aligned} c_{V,U_1,\ldots,U_p} &= \left[\prod_{j=1}^{p} c_{V,U_j;U_1,\ldots,U_{j-1}}\right] \cdot \left[\prod_{i=1}^{p-1}\prod_{j=i+1}^{p} c_{U_j,U_{j+i};U_{j+1},\ldots,U_{i-1}}\right] \\ &= \left[\prod_{j=1}^{p} c_{V,U_j;U_1,\ldots,U_{j-1}}\right] \cdot c_{U_1,\ldots,U_p}, \end{aligned}$$

and similarly as in (4.6) we obtain that the conditional density is equivalent to (4.5). □

4.1 Model setup

Our goal is to estimate the vine copula $C_{V,U_1,\ldots,U_p}$ for a given set of observed n data points. Let

- $\mathbf{v} := \left(v^{(1)}, \ldots, v^{(n)}\right)$,

and

- $\mathbf{u}_j := \left(u_j^{(1)}, \ldots, u_j^{(n)}\right)$, for $j = 1, \ldots, p$

be n independent identically distributed observations from the random vector $(V, U_1, \ldots, U_p)^T$. As already stated, for the purpose of C- and D-vine quantile regression we restrict the vine copula $C_{V,U_1,\ldots,U_p}$ to the classes $\mathfrak{C}^p$ and $\mathfrak{D}^p$, respectively. Still, the order

$$\mathcal{O}\left(C_{V,U_1,\ldots,U_p}\right) = \left(V, U_{i_1}, \ldots, U_{i_p}\right),$$

where $\left(U_{i_1}, \ldots, U_{i_p}\right)$ is any permutation of $(U_1, \ldots, U_p)$, remains a free parameter. It is clear from the Definition 4.4 that we can chose $\mathcal{O}\left(C_{V,U_1,\ldots,U_p}\right)$ arbitrarily. However,

the explanatory power of the fit will greatly depend on the chosen order. Therefore we would like to be able to compare and quantify the explanatory powers of models with different orders. For that purpose, we define the conditional log likelihood function as a goodness of fit measure.

Definition 4.8. (Conditional log likelihood function)
Let $\mathcal{C} = (\mathcal{V}, \mathcal{B}, \Theta)$ *belong to the class* $\mathfrak{C}^p$ *or the class* $\mathfrak{D}^p$ *with order* $\mathcal{O}(\mathcal{C}) = (V, U_1, \ldots, U_p)$. *Additionally, assume that we are given* n *observations* $\mathbf{v}$ *and* $\mathbf{u}_j$ *for* $j = 1, \ldots, p$. *Then the conditional log likelihood function is given as*

$$cll\left(\mathcal{C}, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_p)\right) = \sum_{i=1}^{n} \log c_{V|U_1,\ldots,U_p}\left(v^{(i)}|u_1^{(i)}, \ldots, u_p^{(i)}\right).$$

Corollary 4.9. *The conditional log likelihood of Definition 4.8 can be rewritten as*

$$\begin{aligned} cll\left(\mathcal{C}, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_p)\right) = \sum_{i=1}^{n} \Bigg[& \log c_{V,U_1}\left(v^{(i)}, u_1^{(i)}\right) \\ & + \log c_{V,U_2|U_1}\left(h_{V|U_1}\left(v^{(i)}|u_1^{(i)}\right), h_{U_2|U_1}\left(u_2^{(i)}|u_1^{(i)}\right)\right) \\ & + \sum_{i=3}^{p} \log c_{V,U_i|U_1,\ldots,U_{i-1}}\Big(C_{V|U_1,\ldots,U_{i-1}}\left(v^{(i)}|u_1^{(i)}, \ldots, u_{i-1}^{(i)}\right), \\ & \qquad C_{U_i|U_1,\ldots,U_{i-1}}\left(u_i^{(i)}|u_1^{(i)}, \ldots, u_{i-1}^{(i)}\right)\Big)\Bigg] \end{aligned}$$

Proof. This follows directly from Corollary 4.7. Namely, taking the logarithm of equation (4.5) we obtained the density in the desired form. □

Corollary 4.10. *Let* $\mathcal{C}_1 = (\mathcal{V}_1, \mathcal{B}_1, \Theta_1)$ *belong to the class* $\mathfrak{C}^p$ *or the class* $\mathfrak{D}^p$ *with order* $\mathcal{O}(\mathcal{C}) = (V, U_1, \ldots, U_p)$. *Further, let* $\mathcal{C}_2 = (\mathcal{V}_2, \mathcal{B}_2, \Theta_2)$ *belong to* $\mathfrak{C}^{p-1}$ *if* $\mathcal{C}_1 \in \mathfrak{C}^p$, *or* $\mathfrak{D}^{p-1}$ *if* $\mathcal{C}_1 \in \mathfrak{D}^p$ *with order* $\mathcal{O}(\mathcal{C}) = (V, U_1, \ldots, U_{p-1})$. *Then*

$$cll\left(\mathcal{C}_1, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_p)\right) = cll\left(\mathcal{C}_2, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_{p-1})\right) + \sum_{i=1}^{n} \log c_{V,U_p|U_1,\ldots,U_{p-1}}(\cdot, \cdot)$$

Proof. Writing out the $cll\left(\mathcal{C}_1, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_p)\right)$ as in Corollary 4.9 we obtain

$$\begin{aligned} cll\left(\mathcal{C}_1, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_p)\right) &= \sum_{i=1}^{n} \left[\log c_{V,U_1}(\cdot, \cdot) + \sum_{i=2}^{p} \log c_{V,U_i|U_1,\ldots,U_{i-1}}(\cdot, \cdot)\right] \\ &= \sum_{i=1}^{n} \Bigg[\log c_{V,U_1}(\cdot, \cdot) + \sum_{i=2}^{p-1} \log c_{V,U_i|U_1,\ldots,U_{i-1}}(\cdot, \cdot) \\ &\qquad + \log c_{V,U_p|U_1,\ldots,U_{p-1}}(\cdot, \cdot)\Bigg] \\ &= cll\left(\mathcal{C}_2, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_{p-1})\right) + \sum_{i=1}^{n} \log c_{V,U_p|U_1,\ldots,U_{p-1}}(\cdot, \cdot) \end{aligned}$$

□

However, using the conditional log likelihood as a goodness of fit measure, there is a possibility of overfitting. Overfitted model is a statistical model that contains more parameters than can be justified by the data. Therefore, as another goodness of fit measure we will introduce penalized conditional log likelihood. A penalization that is suitable for vine copula models is the Bayesian information criterion (or shortly BIC).

Definition 4.11. (AIC/BIC penalized conditional log likelihood function)
Let $\mathcal{C} = (\mathcal{V}, \mathcal{B}, \Theta)$ belong to the class $\mathfrak{C}^p$ or the class $\mathfrak{D}^p$ with order $\mathcal{O}(\mathcal{C}) = (V, U_1, \ldots, U_p)$. Additionally, assume that we are given n observations $\mathbf{v}$ and $\mathbf{u}_j$ for $j = 1, \ldots, p$. Let the number of parameters of the vine copula models be $|\Theta|$. Then the AIC- and BIC-penalized conditional log likelihood functions are given as

$$cll_{AIC}(\mathcal{C}, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_p)) = -2cll(\mathcal{C}, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_p)) + 2|\Theta|,$$

$$cll_{BIC}(\mathcal{C}, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_p)) = -2cll(\mathcal{C}, \mathbf{v}, (\mathbf{u}_1, \ldots, \mathbf{u}_p)) + log(n)|\Theta|.$$

In order to compare nested models and thus, decide whether adding a predictor to a model will yield a significant improve in the fit, we introduce the following statistical test:

Definition 4.12. (Conditional likelihood ratio test)
Let $\mathcal{C}_1 = (\mathcal{V}_1, \mathcal{B}_1, \Theta_1)$ belong to the class $\mathfrak{C}^p$ or the class $\mathfrak{D}^p$ with order $\mathcal{O}(\mathcal{C}) = (V, U_1, \ldots, U_p)$. Further, let $\mathcal{C}_2 = (\mathcal{V}_2, \mathcal{B}_2, \Theta_2)$ belong to $\mathfrak{C}^{p-1}$ if $\mathcal{C}_1 \in \mathfrak{C}^p$, or $\mathfrak{D}^{p-1}$ if $\mathcal{C}_1 \in \mathfrak{D}^p$ with order $\mathcal{O}(\mathcal{C}) = (V, U_1, \ldots, U_{p-1})$. Additionally, assume that we are given n observations on each of the considered variables, i.e. $\mathbf{v}, \mathbf{u}_j$ for $j = 1, \ldots p$. Then we define the conditional likelihood ratio test between the vine copula models $\mathcal{C}_1$ and $\mathcal{C}_2$ as the test which rejects the null hypothesis

$$H_0 \ : \ \textit{Adding } U_p \textit{ to the model } \mathcal{C}_2 \textit{ does not improve the } fit$$

at level $\alpha \in (0, 1)$, if

$$cll(\mathcal{C}_1, \mathbf{v}, (\mathbf{u}_1, \ldots \mathbf{u}_p)) - cll(\mathcal{C}_2, \mathbf{v}, (\mathbf{u}_1, \ldots \mathbf{u}_{p-1})) > \chi^2_{1-\alpha, |\Theta_1|-|\Theta_2|}.$$

where $\chi^2_{1-\alpha, |\Theta_1|-|\Theta_2|}$ denotes the $(1-\alpha)-$quantile of a χ^2-distribution with $|\Theta_1| - |\Theta_2|$ degrees of freedom.

For more details about Definition 4.12, we refer to Shao (2003) and Cheng (2017).

4.2 Estimation of margins

Up until now, we always supposed that the margins F_Y and F_{X_i} are known. However, in a nonsimulated data set the margins are almost always unknown. For a copula based approach to quantile regression knowing the margins is crucial for two reasons:

- First, F_Y or an estimate of F_Y are needed in order to be able to obtain the conditional quantile function q_α from the conditional distribution $C_{V|U_1,\ldots,U_p}$.
- Second, to estimate $C_{V|U_1,\ldots,U_p}$ from a non uniform data set, one needs the margins to transform the data on the $u-$scale.

There is a vast literature about the estimation of univariate conditional distribution functions and generally we have two choices of how to fit the marginal distribution, either parametrically or nonparametrically. Given that the bivariate copulas are to be fitted parametrically, we can either have a full parametric or a semiparametric estimate of q_α. Modeling the marginals and the copulas parametrically might cause the resulting fully parametric estimator to be biased and inconsistent, if one of the parametric models is misspecified (a detailed discussion in Noh et al. (2013)). Therefore, for the purpose of quantile regression, we will use nonparametric estimators $\hat{F}_Y$ and $\hat{F}_{X_i}$ of the margins. One of the simplest nonparametric estimation methods for marginal distributions is the empirical distribution function, but due to its discrete nature and the fact that we need inverses for calculating q_α, we opt against it. Instead we use the univariate local polynomial kernel density estimators.

Definition 4.13. (Univariate kernel density estimator)
Given a sample $(x_1, \ldots, x_n)$ from a random variable X, the univariate local polynomial kernel density estimator is defined as

$$\hat{F}(x) = \frac{1}{nb} \sum_{i=1}^{n} K\left(\frac{x - x_i}{b}\right), \quad x \in \mathbb{R},$$

where $K(x) := \int_{-\infty}^{x} k(t)\, dt$ with $k(\cdot)$ being a symmetric probability density function and $b > 0$ is a bandwidth parameter.

We chose K to be a Gaussian kernel, i.e.

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2},$$

and use the optimal bandwidth parameter b developed in Sheather and Jones (1991).

The bandwidth b controls the smoothness or complexity of the estimate $\hat{F}(x)$. Thus, b plays the main role in the bias-variance trade off of the estimator. To find the optimal bandwidth parameter cross validation is used, and for more details we refer to Sheather and Jones (1991). The univariate local polynomial kernel density estimator is implemented in the R library `kde1d-package` as the function `kde1d`.

Chapter 5

Forward selection algorithms

Given a response variable V and p possible predictors, one would optimally want to use the $cll-$optimal $\mathfrak{C}^p$ or $\mathfrak{D}^p$ vine model for the purpose of quantile regression. However, that would mean estimating all possible $\mathfrak{C}^p(\mathfrak{D}^p)$ vine models and thus, also estimating the C- or D-vine copulas associated with the models. Although, the number of possible regular vine models is drastically reduced by focusing only on models from the classes $\mathfrak{C}^p$ or $\mathfrak{D}^p$ computing the $cll-$ optimal model still remains intractable for larger number of predictors.

The classes $\mathfrak{C}^p$ or $\mathfrak{D}^p$ contain exactly $p!$ regular vine models with different orders each. Consequently, since the order of a model uniquely defines the underlying tree sequence and vice versa, there are also $p!$ different vine copulas associated with the different models from a class. Thus, we would like to have an algorithm that will automatically choose the order of the predictors, so that the resulting model for the prediction of the conditional quantiles will have the highest explanatory power.

Therefore, in the following section we introduce two algorithms that automatically select the order of covariates. The first algorithm is based on the work of Kraus and Czado (2017), who made an algorithm for the class of D-vines, $\mathfrak{D}^p$, and looked sequentially which covariates to add, based on which covariate in each step yield the highest improvement in the conditional log likelihood, i.e., they looked one step ahead in the tree sequence. Following this idea, we implement the algorithm to the class of C-vines, $\mathfrak{C}^p$. We present the implementation for both classes as a single algorithm, in which we can choose whether we want a D-vine, or C-vine model, based on the data set we and background knowledge of dependency structures.

The second algorithm we introduce, is an "improvement" to the first algorithm, in a sense that we look two steps ahead in search of the next covariate at each step. Basically, at every step we try each of the remaining covariates, and on that expanded model, we look one more step ahead, at the second new tree showing up. Then, we choose the covariate that has the greatest improvement in the next two trees. This way, step by step the covariates are ordered, based on their power to predict the response.

We also adapt this algorithm for large data sets, with many covariates. As imple-

menting the algorithm on a large data set is computationally expensive, we include some randomization, so that our models generalize better and are computationally tractable. Furthermore, as done by Kraus and Czado (2017) an automated forward selection procedure to choose a subset of the most influential covariates in the model is introduced.

5.1 Algorithm 1: Forward selection of C- and D-vine quantile regression models

Input: We start with a given data set where

$$\mathbf{y} := \left(y^{(1)}, \ldots, y^{(n)}\right),$$

and

$$\mathbf{x}_j := \left(x_j^{(1)}, \ldots, x_j^{(n)}\right), \quad for\ j = 1, \ldots, p$$

are n independent identically distributed observations from the random vector $(Y, X_1, \ldots, X_p)^T$.

Data preprocessing:

- As already proposed in Section 4.2 we estimate the margins $\hat{F}_Y$ and $\hat{F}_{X_j}$ for $j = 1, \ldots, p$.
- With the margins estimated, then we compute the pseudo copula data by applying the probability integral transform:

$$\hat{v}^{(i)} := \hat{F}_Y\left(y^{(i)}\right) \quad \text{and} \quad \hat{u}_j^{(i)} := \hat{F}_{X_j}\left(x_j^{(i)}\right), \quad \text{for } i = 1, \ldots, n, \;\; j = 1, \ldots, p.$$

Initialization:

- We have to decide beforehand with which vine copula we will model the data. Namely, we have to choose between the classes $\mathfrak{D}^p$ or $\mathfrak{C}^p$.
- The selection criteria for the estimation of pair copulas must be chosen and the penalization method for the *cll*, if desired.
- The maximal number of predictors that can be included L has to be predefined.

Step 1:

- For all $k = 1, \ldots, p$ we calculate the conditional log likelihood of the bivariate copulas $\mathcal{C}_k \in \mathfrak{D}^1$ (or $\mathfrak{C}^1$) with order $\mathcal{O}(\mathcal{C}_k) = (V, U_k)$, i.e.

$$cll\left(\mathcal{C}_k, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_k)\right) = \sum_{i=1}^{n} \log c_{V,U_k}\left(\hat{v}^{(i)}, \hat{u}_k^{(i)}\right).$$

- We define t_1 as
$$t_1 := \underset{k=1,\dots,p}{\arg\max}\, cll\left(\mathcal{C}_k, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_k)\right). \tag{5.1}$$
When a penalization from Definition 4.11 is desired, the arg max is taken over the cll_{BIC} or cll_{AIC}, rather then over the cll of the copulas $\mathcal{C}_k$ in equation (5.1).

- As our current optimal fit, denoted by $\mathcal{C}^*$, we choose the vine copula $\mathcal{C}_{t_1}$. Thus the order of $\mathcal{C}^*$ after the first step is given by
$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_{t_1}\right).$$

Step r:

- At the $r-$step our current optimal fit has order
$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_{t_1}, \dots, U_{t_{r-1}}\right).$$

- Further, for all $k \in S := \{1,\dots,p\} \setminus \{t_1,\dots,t_{r-1}\}$ we calculate the conditional log likelihood of the vine copulas $\mathcal{C}_k \in \mathfrak{D}^r$ (or $\mathfrak{C}^r$) with order $\mathcal{O}\left(\mathcal{C}_k\right) = \left(V, U_{t_1}, \dots, U_{t_{r-1}}, U_k\right)$, i.e.
$$\begin{aligned} cll\left(\mathcal{C}_k, \hat{\mathbf{v}},\ \left(\hat{\mathbf{u}}_{t_1} \dots \hat{\mathbf{u}}_{t_{r-1}}, \hat{\mathbf{u}}_k\right)\right) &= cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, \left(\hat{\mathbf{u}}_{t_1} \dots \hat{\mathbf{u}}_{t_{r-1}}\right)\right) \\ &+ \sum_{i=1}^{n} \log c_{VU_k;U_{t_1},\dots,U_{t_{r-1}}}\left(C_{V|U_{t_1},\dots,U_{t_{r-1}}}\left(v^{(i)}|u_{t_1}^{(i)},\dots,u_{t_{r-1}}^{(i)}\right),\right. \\ &\left. C_{U_k|U_{t_1},\dots,U_{t_{r-1}}}\left(u_k^{(i)}|u_{t_1}^{(i)},\dots,u_{t_{r-1}}^{(i)}\right)\right). \end{aligned}$$
Here we used the result of Corollary 4.10.

- We define t_r as
$$t_r := \underset{k\in S}{\arg\max}\, cll\left(\mathcal{C}_k, \hat{\mathbf{v}}, \left(\hat{\mathbf{u}}_{t_1} \dots \hat{\mathbf{u}}_{t_{r-1}}, \hat{\mathbf{u}}_k\right)\right). \tag{5.2}$$
Similarly as in step 1, when a penalization from Definition 4.11 is desired, the arg max is taken over the cll_{BIC} or cll_{AIC}, rather then over the cll of the copulas $\mathcal{C}_k$ in equation (5.2).

- We update the optimal fit $\mathcal{C}^*$ with the vine copula $\mathcal{C}_{t_r}$. Thus, the order of $\mathcal{C}^*$ after the $r-$step is given as
$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_{t_1}, \dots U_{t_r}\right).$$

Stopping: We continue the iteration as described above until we select a preferred number L of covariates or until the conditional log likelihood no longer yields a significant increase compared to the previous step, for which we utilize the conditional log likelihood ratio test from Definition 4.12.

Output: A vine copula $\mathcal{C}^* \in \mathfrak{D}^d$ $\left(\text{or } \mathfrak{C}^d\right)$ with order
$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_{t_1}, \dots, U_{t_d}\right),$$
where $d \leq L$ and represents the number of significant predictors found by the algorithm.

5.1.1 Illustration of Algorithm 1

In this section, we give a step by step illustration of the algorithm introduced in Section 5.1 on a simulated 6-dimensional non-gaussian data set. Further, detailed estimates of pair copulas and their corresponding log likelihoods are provided, as well as the intermediate optimal models at each step.

Input:

We consider a six dimensional data set $\left(v^{(i)}, u_1^{(i)}, u_2^{(i)}, u_3^{(i)}, u_4^{(i)}, u_5^{(i)}\right)^T$, $i = 1, \ldots, 500$, sampled from $(V, U_1, U_2, U_3, U_4, U_5)$ which follows a six dimensional D-vine copula distribution $C_{V,U_1,U_2,U_3,U_4,U_5}$.

The pair copulas of the D-vine $C_{V,U_1,U_2,U_3,U_4,U_5}$ are given in Table 5.1.

Tree	Edge	Conditioned	;	Conditioning	Family	Tau
1	1	V, U_1	;		Clayton	**0.60**
1	2	U_1, U_2	;		Joe	0.80
1	3	U_2, U_3	;		Gumbel	0.50
1	4	U_3, U_4	;		Gauss	0.13
1	5	U_4, U_5	;		Indep.	0
2	1	V, U_2	;	U_1	Gumbel	**0.80**
2	2	U_1, U_3	;	U_2	Frank	0.65
2	3	U_2, U_4	;	U_3	Joe	0.49
2	4	U_3, U_5	;	U_4	Gauss	0.13
3	1	V, U_3	;	U_1, U_2	Joe	**0.60**
3	2	U_1, U_4	;	U_2, U_3	Frank	0.55
3	3	U_2, U_5	;	U_3, U_4	Gauss	0.19
4	1	V, U_4	;	U_1, U_2, U_3	Clayton	**0.50**
4	2	U_1, U_5	;	U_2, U_3, U_4	Gauss	0.06
5	1	V, U_5	;	U_1, U_2, U_3, U_4	Indep.	0

Table 5.1: Pair copulas of the D-vine $C_{V,U_1,U_2,U_3,U_4,U_5}$.

Data preprocessing:

The data is already on the copula scale, thus estimation of margins and transforming the data is not necessary.

Initialization:

To model the given data we decide to use a vine copula model from the D-vine class $\mathfrak{D}^p$ with $p \leq 5$.

Step 1:

To add the first predictor to the model the conditional log likelihood of the candidate models $\mathcal{C}_j$ with order

$$\mathcal{O}(V, U_j), \quad \text{for } j = 1, \ldots, 5,$$

has to be calculated.

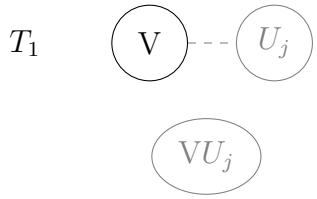

Figure 5.1: We are searching for a model from the class $\mathfrak{D}^p$, thus V is already set as the first node (black) in T_1. In the 1-st step of the algorithm we add a new predictor U_j (gray). The gray copula VU_j has to be estimated.

To calculate the *cll* of $\mathcal{C}_j$ we first estimate the pair copulas C_{VU_j} for $j = 1, \ldots, 5$. For this we use the function `bicop` from the package `rvinecopulib`, using the default selection criteria for bivariate copula estimation BIC.

Pair copulas	$\hat{C}_{VU_1}$	$\hat{C}_{VU_2}$	$\hat{C}_{VU_3}$	$\hat{C}_{VU_4}$	$\hat{C}_{VU_5}$
Family	Joe(180)	Gumbel(180)	Clayton	Gauss	Gauss
Parameter	3.87	2.79	1.29	0.54	0.26
Loglik.	322.58	**327.78**	126.61	86.90	18.97

Table 5.2: Estimated bivariate copulas with their parameters.

In the first iteration step the conditional log likelihood of the candidate models $\mathcal{C}_j$ is equal to the log likelihood of the pair copula $\hat{C}_{VU_j}$.

Since the candidate model $\mathcal{C}_2$ has the greatest conditional log likelihood, i.e. the pair copula $\hat{C}_{VU_2}$ has the greatest log likelihood, we update the current optimal fit order with U_2. Namely, we set the current optimal fit to be the D-vine copula model for quantile regression $\mathcal{C}^* \in \mathfrak{D}^1$ with order

$$\mathcal{O}(\mathcal{C}^*) = (V, U_2).$$

The conditional log likelihood of the model $\mathcal{C}^*$ is 327.78.

Before we continue with the second step, we estimate the pseudo copula data needed for fitting the pair copulas in the second tree

$$\hat{u}^{(i)}_{V|U_2} = h_{V|U_2}\left(\hat{v}^{(i)}|\hat{u}^{(i)}_2\right).$$

This is done using the function `hbicop` from the package `rvinecopulib`.

Step 2:

In order to estimate the condition log likelihood of the vine copulas $\mathcal{C}_j \in \mathfrak{D}^2$ with order $\mathcal{O}(\mathcal{C}_j) = (V, U_2, U_j)$, for $j = 1, 3, 4, 5$, we first have to fit the pair copulas $\hat{C}_{U_2U_j}$ and $\hat{C}_{VU_j;U_2}$.

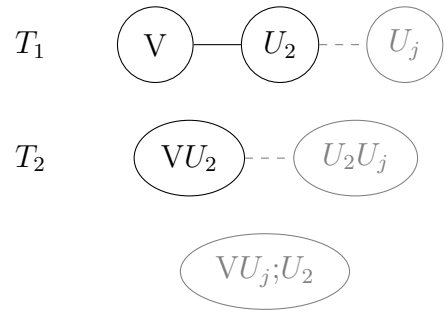

Figure 5.2: Extending the current D-vine (black) by adding U_j in the 2-nd step of the algorithm. Thus, the gray pair-copulas need to be estimated.

The pair copulas $\hat{C}_{U_2U_j}$ have to be estimated in order to obtain the pseudo copula data

$$\hat{u}^{(i)}_{U_j|U_2} = h_{U_j|U_2}\left(\hat{u}^{(i)}_j|\hat{u}^{(i)}_2\right)$$

We give the estimates of these pair copula in Table 5.3.

Pair copulas	$\hat{C}_{U_2U_1}$	$\hat{C}_{U_2U_3}$	$\hat{C}_{U_2U_4}$	$\hat{C}_{U_2U_5}$
Family	Joe	Gumbel	Gumbel	Gauss
Parameter	8.91	2.04	1.60	0.29

Table 5.3: Estimated bivariate copulas with their parameters.

After estimating of the pair copulas $\hat{C}_{U_2U_j}$ we can estimate $\hat{C}_{VU_j;U_2}$ using the pseudo copula data $\hat{u}^{(i)}_{V|U_2}$ and $\hat{u}^{(i)}_{U_j|U_2}$. For the pair copulas $\hat{C}_{VU_j;U_2}$ we also give their corresponding log likelihoods since they contribute to the *cll*, while the pair copulas $\hat{C}_{U_2U_j}$ do not contribute to the *cll* directly.

Pair copulas	$\hat{C}_{VU_1;U_2}$	$\hat{C}_{VU_3;U_2}$	$\hat{C}_{VU_4;U_2}$	$\hat{C}_{VU_5;U_2}$
Family	t	t	Gauss	Indep.
Parameter	-0.49, 2	-0.11, 5.67	0.16	0
Loglik.	123.97	12.25	6.24	0

Table 5.4: Estimated bivariate copulas with their parameters.

Now we can calculate the conditional log likelihood of the copulas $\mathcal{C}_j$ which is given as

$$cll\left(\mathcal{C}_j, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_j)\right) = cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2)\right) + l\left(\hat{C}_{VU_j;U_2}\right)$$

Candidate models	$\mathcal{C}_1$	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$
Conditional log likelihood	**451.75**	340.02	334.02	327.78

Table 5.5: Conditional log likelihood of candidate D-vine models.

Further, we apply the conditional likelihood ratio test from Definition 4.12 to the copula model with the greatest conditional log likelihood. That is we check

$$cll\left(\mathcal{C}_j, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_j)\right) - cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2)\right) > \chi^2_{0.95,1}.$$

Since copula $\mathcal{C}_1$ has the greatest conditional log likelihood it follows that

$$cll\left(\mathcal{C}_1, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_1)\right) - cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2)\right) = 123.97 > 5.99 = \chi^2_{0.95,2}.$$

Thus, it is justified to add U_1 to the model.

We update our optimal fit $\mathcal{C}^*$ to be the candidate model $\mathcal{C}_1$. This means that we add U_1 as the next predictor, thus our estimated D-vine copula model after the second step is the $\mathcal{C}^* \in \mathfrak{D}^2$ with order

$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_2, U_1\right).$$

The conditional log likelihood of the model $\mathcal{C}^*$ is 451.78.

Again we calculate the pseudo copula data needed for the next step,

$$\hat{u}^{(i)}_{V|U_2,U_1} = h_{V|U_1;U_2}\left(\hat{u}^{(i)}_{V|U_2}|\hat{u}^{(i)}_{U_1|U_2}\right).$$

Step 3:

For $j = 3, 4, 5$, in order to estimate the condition log likelihood of the candidate models $\mathcal{C}_j \in \mathfrak{D}^3$ with order $\mathcal{O}(\mathcal{C}_j) = (V, U_2, U_1, U_j)$, we first have to fit the pair copulas $\hat{C}_{U_jU_1}$, $\hat{C}_{U_jU_2;U_1}$ and $\hat{C}_{VU_j;U_2,U_1}$.

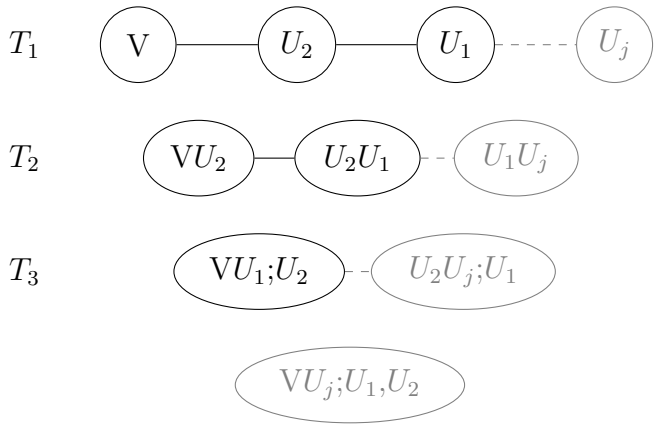

Figure 5.3: Extending the current D-vine (black) by adding U_j in the 3-rd step of the algorithm. Thus, the gray pair-copulas need to be estimated.

Similarly as in Step 2, the pair copulas $\hat{C}_{U_jU_1}$ and $\hat{C}_{U_jU_1;U_2}$ are fitted only to obtain the pseudo copula data

$$\hat{u}^{(i)}_{U_j|U_2,U_1} = h_{U_j|U_2;U_1}\left(\hat{u}^{(i)}_{U_j|U_1}|\hat{u}^{(i)}_{U_2|U_1}\right), \tag{5.3}$$

and their estimates are given in Table 5.6.

Pair copulas	$\hat{C}_{U_3U_1}$	$\hat{C}_{U_3U_1;U_2}$	$\hat{C}_{U_4U_1}$	$\hat{C}_{U_4U_1;U_2}$	$\hat{C}_{U_5U_1}$	$\hat{C}_{U_5U_1;U_2}$
Family	t	Frank	Gauss	Clayton(180)	Gauss	Frank
Parameter	0.86, 5.14	-4.41	0.58	0.39	0.24	1.23

Table 5.6: Estimated bivariate copulas with their parameters.

With the pair copulas from Table 5.6 the pseudo copula data from equation (5.3) can be calculated. Then, we can estimate $\hat{C}_{VU_j;U_2,U_1}$ with the pseudo data $\hat{u}^{(i)}_{V|U_2,U_1}$ and $\hat{u}^{(i)}_{U_j|U_2,U_1}$.

We calculate the conditional log likelihood for the candidate models $\mathcal{C}_j$ as

$$cll\left(\mathcal{C}_j, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_1, \hat{\mathbf{u}}_j)\right) = cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_1)\right) + l\left(\hat{C}_{VU_j;U_2,U_1}\right).$$

Pair copulas	$\hat{C}_{VU_3;U_2,U_1}$	$\hat{C}_{VU_4;U_2,U_1}$	$\hat{C}_{VU_5;U_2,U_1}$
Family	t	Frank	Indep.
Parameter	0.20, 6.33	-0.76	0
Loglik.	16.10	3.47	0

Table 5.7: Estimated bivariate copulas with their parameters.

Candidate models	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$
Conditional log likelihood	**467.84**	455.22	451.75

Table 5.8: Conditional log likelihood of candidate D-vine models.

Obviously $\mathcal{C}_3$ has the greatest conditional log likelihood. Again, we have to apply the conditional likelihood ratio test which yields

$$cll\left(\mathcal{C}_3, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_1, \hat{\mathbf{u}}_3)\right) - cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_1)\right) = 16.10 > 5.99 = \chi^2_{0.95,2}.$$

Thus, it is justified to add U_3 to the model.

We update our optimal fit $\mathcal{C}^*$ to the candidate model $\mathcal{C}_3$. Thus, our estimated D-vine copula model after the third step is the $\mathcal{C}^* \in \mathfrak{D}^3$ with order

$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_2, U_1, U_3\right).$$

The conditional log likelihood of the model $\mathcal{C}^*$ is 467.84.

Again, we calculate the pseudo copula data needed for the next step,

$$\hat{u}^{(i)}_{V|U_2,U_1,U_3} = h_{V|U_3;U_2,U_1}\left(\hat{u}^{(i)}_{V|U_2,U_1}|\hat{u}^{(i)}_{U_3|U_2,U_1}\right).$$

Step 4:

Similarly as in the previous steps, in order to estimate the condition log likelihood of the candidate models $\mathcal{C}_j \in \mathfrak{D}^4$, for $j = 4, 5$, there is a number of pair copulas to be estimated. Namely, in this step, those pair copulas are $\hat{C}_{U_jU_3}$, $\hat{C}_{U_jU_1;U_3}$, $\hat{C}_{U_jU_2;U_1,U_3}$ and $\hat{C}_{VU_j;U_2,U_1,U_3}$.

The pair copulas $\hat{C}_{U_jU_3}$, $\hat{C}_{U_jU_1;U_3}$ and $\hat{C}_{U_jU_2;U_1,U_3}$ are fitted to obtain the pseudo copula data

$$\hat{u}^{(i)}_{U_j|U_2,U_1,U_3} = h_{U_j|U_2;U_1,U_3}\left(\hat{u}^{(i)}_{U_j|U_1,U_3}|\hat{u}^{(i)}_{U_2|U_1,U_3}\right).$$

Further, we estimate $\hat{C}_{VU_j;U_2,U_1,U_3}$ for $j = 4, 5$ with the pseudo copula data $\hat{u}^{(i)}_{V|U_2,U_1,U_3}$ and $\hat{u}^{(i)}_{U_j|U_2,U_1,U_3}$.

Pair copulas	$\hat{C}_{U_4U_3}$	$\hat{C}_{U_4U_1;U_3}$	$\hat{C}_{U_4U_2;U_1,U_3}$	$\hat{C}_{U_5U_3}$	$\hat{C}_{U_5U_1;U_3}$	$\hat{C}_{U_5U_2;U_1,U_3}$
Family	Gauss	Gumbel	Frank	Gauss	Frank	Gauss
Parameter	0.24	2.56	-2.14	0.21	0.80	0.22

Table 5.9: Estimated bivariate copulas with their parameters.

Pair copulas	$\hat{C}_{VU_4;U_2,U_1,U_3}$	$\hat{C}_{VU_5;U_2,U_1,U_3}$
Family	Indep.	Indep.
Parameter	0	0
Loglik.	0	0

Table 5.10: Estimated bivariate copulas with their parameters.

Both pair copulas, $\hat{C}_{VU_4;U_2,U_1,U_3}$ and $\hat{C}_{VU_5;U_2,U_1,U_3}$, are estimated to be the Independence copula.

Thus, adding either of the covariates, U_4 or U_5, to the model will not change the conditional log likelihood and therefore we do not include any of them in the model. Not including any new predictors also stops the iteration process.

Output: The optimal D-vine copula model $\mathcal{C}^* \in \mathfrak{D}^3$, lastly updated at Step 3 and has order

$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_2, U_1, U_3\right),$$

with conditional log likelihood

$$cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, \left(\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_1, \hat{\mathbf{u}}_3\right)\right) = \mathbf{467.84}. \tag{5.4}$$

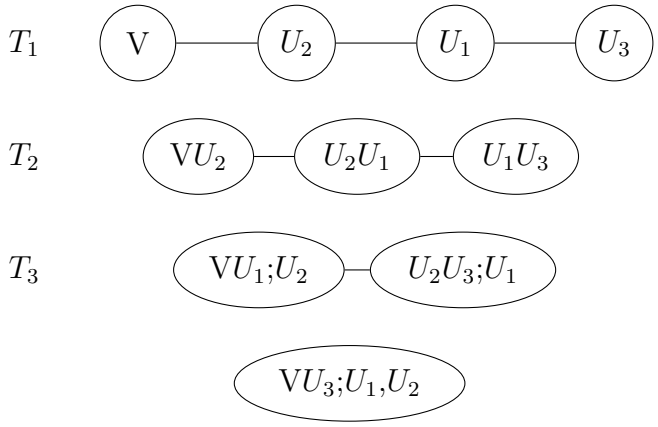

Figure 5.4: D-vine tree sequence of the optimal fit

5.2 Algorithm 2: Two step ahead forward selection of C- and D-vine quantile regression models

Input: Again as with Algorithm 1 from Section 5.1, we start with a given data set where

$$\mathbf{y} := \left(y^{(1)}, \ldots, y^{(n)}\right),$$

and

$$\mathbf{x}_j := \left(x_j^{(1)}, \ldots, x_j^{(n)}\right), \quad for\ j = 1, \ldots, p$$

are n independent identically distributed observations from the random vector $(Y, X_1, \ldots, X_p)^T$.

Data preprocessing:

- Following Section 4.2 we estimate the margins $\hat{F}_Y$ and $\hat{F}_{X_j}$ for $j = 1, \ldots, p$.
- With the margins estimated, the pseudo copula data is computed by applying the probability integral transform:

$$\hat{v}^{(i)} := \hat{F}_Y\left(y^{(i)}\right) \quad \text{and} \quad \hat{u}_j^{(i)} := \hat{F}_{X_j}\left(x_j^{(i)}\right) \quad \text{for } i = 1, \ldots, n, \;\; j = 1, \ldots, p.$$

Initialization:

- We have to decide beforehand with which vine copula we will model the data. Namely, we have to choose between the classes $\mathfrak{D}^p$ and $\mathfrak{C}^p$.
- The selection criteria for the estimation of pair copulas must be chosen and the penalization method for the *cll*, if desired.
- Further, the number of candidates k and the maximal number of predictors which can be included in the model L have to be predefined.

Step 1:

- Obtain estimates of Kendall's τ values τ_{VU_j} for $j = 1, \ldots p$ based on

$$\left\{v^{(i)}, u_j^{(i)} | i = 1, \ldots, n\right\}.$$

- Choose $k \leq p$ highest estimates $|\hat{\tau}_{VU_j}|$ and identify the indices $p_1, \ldots, p_k$, where

$$|\hat{\tau}_{VU_{p_1}}| \geq |\hat{\tau}_{VU_{p_2}}| \geq \ldots \geq |\hat{\tau}_{VU_{p_k}}|.$$

  We take the covariates $U_{p_1}, \ldots, U_{p_k}$ as our candidate predictors at step 1 and we define the set of candidate indices as $K = \{p_1, \ldots, p_k\}$.

- For all candidates, i.e., $c \in K$, we consider the $p-1$ two step ahead copula models. That is, for $j \in \{1,\dots,p\}\backslash\{c\}$ we consider the models $\mathcal{C}_{c,j} \in \mathfrak{D}^2$ (or $\mathfrak{C}^2$) with order $\mathcal{O}\left(\mathcal{C}_{c,j}\right) = (V, U_c, U_j)$ and we calculate the conditional log likelihood

$$cll\left(\mathcal{C}_{c,j}, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j)\right) = \sum_{i=1}^{n} \Big[\log c_{V,U_c}\left(\hat{v}^{(i)}, \hat{u}_c^{(i)}\right) + \log c_{V,U_j|U_c}\left(h\left(\hat{v}^{(i)}|\hat{u}_c^{(i)}\right), h\left(\hat{u}_j^{(i)}|\hat{u}_c^{(i)}\right)\right)\Big].$$

- We define as $m\mathcal{C}_c$ the maximal two step ahead cll for each of the candidate predictors, i.e.,

$$m\mathcal{C}_c := \max_{j\in\{1,\dots,p\}\backslash\{c\}} cll\left(\mathcal{C}_{c,j}, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j)\right), \ \forall c \in K. \tag{5.5}$$

- Then we define t_1 as

$$t_1 := \arg\max_{c\in K} \ m\mathcal{C}_c.$$

- When a penalization from Definition 4.11 is desired, the max is taken over the cll_{BIC} or cll_{AIC}, rather then over the cll of the copulas $\mathcal{C}_k$ in equation (5.5).

- As our current optimal fit, denoted by $\mathcal{C}^*$, we choose the vine copula $\mathcal{C}_{t_1}$ with order

$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_{t_1}\right).$$

Step r:

- At the $r-$step, the current optimal fit has order

$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_{t_1}, \dots, U_{t_{r-1}}\right).$$

- To include the next covariate we first compute the empirical partial correlations $\rho_{V,U_j;U_{t_1},\dots,U_{t_{r-1}}}$ for $j \in \{1,2,\dots,p\} \setminus \{t_1,\dots,t_{r-1}\}$.

- We choose the k indices $p_1,\dots,p_k$ with the largest values of $|\hat{\rho}_{V,U_j;U_{t_1},\dots,U_{t_{r-1}}}|$ as the candidate set K for the r-th covariate node.

- We are interested in the conditional log likelihood of the $(r+2)-$dimensional D-vine or C-vine model. More precisely, we are considering the conditional log likelihood of the two step ahead models $\mathcal{C}_{c,j}$ with order

$$\mathcal{O}\left(V, U_{t_1}, \dots, U_{t_{r-1}}, U_c, U_j\right),$$

  for all $c \in \{p_1,\dots,p_k\}$ and $j \in \{1,2,\dots,p\} \setminus \{t_1,\dots,t_{r-1},c\}$. This yields $k\,(p-r)$ candidate models in the r-th step of the algorithm.

- Following Corollary 4.10 we compute the conditional log likelihoods for the copulas $\mathcal{C}_{c,j}$ as

$$\begin{aligned} cll\left(\mathcal{C}_{c,j},\ \hat{\mathbf{v}}, \left(\hat{\mathbf{u}}_{t_1} \dots \hat{\mathbf{u}}_{t_{r-1}}, \hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j\right)\right) = &\ cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, \left(\hat{\mathbf{u}}_{t_1} \dots \hat{\mathbf{u}}_{t_{r-1}}\right)\right) \\ &+ \sum_{i=1}^{n} \log c_{VU_c;U_{t_1},\dots,U_{t_{r-1}}} \left(C_{V|U_{t_1},\dots,U_{t_{r-1}}}\left(v^{(i)}|u_{t_1}^{(i)},\dots,u_{t_{r-1}}^{(i)}\right),\right. \\ &\qquad \left. C_{U_c|U_{t_1},\dots,U_{t_{r-1}}}\left(u_c^{(i)}|u_{t_1}^{(i)},\dots,u_{t_{r-1}}^{(i)}\right)\right) \\ &+ \sum_{i=1}^{n} \log c_{VU_j;U_{t_1},\dots,U_{t_{r-1}},U_c} \left(C_{V|U_{t_1},\dots,U_{t_{r-1}},U_c}\left(v^{(i)}|u_{t_1}^{(i)},\dots,u_{t_{r-1}}^{(i)},u_c^{(i)}\right),\right. \\ &\qquad \left. C_{U_j|U_{t_1},\dots,U_{t_{r-1}},U_c}\left(u_j^{(i)}|u_{t_1}^{(i)},\dots,u_{t_{r-1}}^{(i)},u_c^{(i)}\right)\right). \end{aligned}$$

- We define as $m\mathcal{C}_c$ the maximal two step ahead *cll* for each of the candidate predictors, i.e.,

$$m\mathcal{C}_c := \max_{j\in\{1,2,\dots,p\}\setminus\{t_1,\dots,t_{r-1},c\}} cll\left(\mathcal{C}_{c,j}, \hat{\mathbf{v}}, \left(\hat{\mathbf{u}}_{t_1} \dots \hat{\mathbf{u}}_{t_{r-1}}, \hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j\right)\right),\ \forall c \in K. \quad (5.6)$$

- Further, we define t_r as

$$t_r := \arg\max_{c\in K}\ m\mathcal{C}_c.$$

- Again, when a penalization from Definition 4.11 is desired, the max is taken over the cll_{BIC} or cll_{AIC}, rather then over the *cll* of the copulas $\mathcal{C}_k$ in equation (5.6).

- We update the optimal fit $\mathcal{C}^*$ with the predictor U_{t_r}. Thus, the order of $\mathcal{C}^*$ after the $r-$step is given by

$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_{t_1}, \dots U_{t_r}\right).$$

Stopping: We continue the iteration as described above until we select a preferred number L of covariates or until the conditional log likelihood no longer yields a significant increase compared to the previous step, for which we utilize the conditional log likelihood ratio test from Definition 4.12.

Output: A vine copula $\mathcal{C}^* \in \mathfrak{D}^d$ $\left(\text{or } \mathfrak{C}^d\right)$ with order

$$\mathcal{O}\left(\mathcal{C}^*\right) = \left(V, U_{t_1}, \dots, U_{t_d}\right).$$

where $d \leq L$ and represents the number of significant predictors found by the algorithm.

5.2.1 Illustration of Algorithm 2

In this section, we give a step by step illustration of the algorithm introduced in Section 5.2 on a simulated 6-dimensional non-gaussian data set. Further, detailed estimates of pair copulas and their corresponding log likelihoods are provided, as well as the intermediate optimal models at each step.

Input:

We consider the same 6-dimensional data set as in Subsection 5.1.1.

Data preprocessing:

The data is already on the copula scale, thus estimation of margins and transforming the data is not necessary.

Further, we set the number of candidates $k = 2$ and the maximal number of predictors that can be included to $L = 5$.

No penalization of the conditional log likelihood is used and the pair copula selection criteria is set to "BIC".

Initialization:

To model the given data we decide to use a vine copula model from the D-vine class $\mathfrak{D}^p$ with $p \leq 5$.

Step 1:

To obtain the candidates in the first step, we estimate first the Kendall's tau values between the response and all possible predictors, denoted by $\hat{\tau}_{VU_j}$ for $j = 1, \ldots 5$ based on

$$\left\{ v^{(i)}, u_j^{(i)} | i = 1, \ldots, 500 \right\}.$$

For this we use the function `cor` from the package `stats`.

$\hat{\tau}_{VU_1}$	$\hat{\tau}_{VU_2}$	$\hat{\tau}_{VU_3}$	$\hat{\tau}_{VU_4}$	$\hat{\tau}_{VU_5}$
0.62	**0.71**	0.39	0.37	0.17

Table 5.11: Estimated Kendall's tau values.

$k = 2$ predictors with the greatest estimated absolute Kendall's tau values are chosen as candidate predictors. In this case, that are U_2 and U_1 as can be seen from Table 5.11, and the set of candidate indices is then $K = \{2, 1\}$.

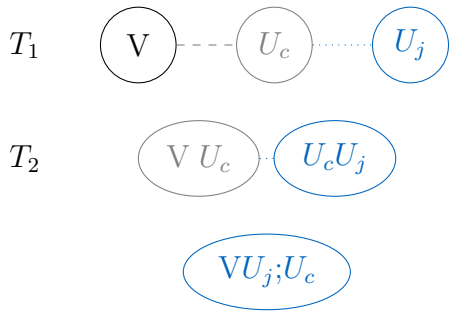

Figure 5.5: We are searching for a model from the class $\mathfrak{D}^p$, thus V is already set as the first node (black) in T_1. In the 1-st step of the algorithm we are extending it for candidate U_c (gray) and predictor U_j (blue). The gray and blue pair-copulas need to be estimated.

For $c \in K$ the conditional log likelihoods of interest are those of the two-step ahead copulas $\mathcal{C}_{c,j} \in \mathfrak{D}^2$ for $j \in \{1,\dots,5\} \setminus \{c\}$ with order $\mathcal{O}(\mathcal{C}_{c,j}) = (V, U_c, U_j)$. Those *cll*'s are obtained as

$$cll\left(\mathcal{C}_{c,j}, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j)\right) = l\left(\hat{C}_{VU_c}, (\hat{\mathbf{v}}, \hat{\mathbf{u}}_j)\right) + l\left(\hat{C}_{VU_j;U_c}, \left(\hat{\mathbf{u}}_{V|U_c}, \hat{\mathbf{u}}_{U_j|U_c}\right)\right), \quad (5.7)$$

where $\hat{\mathbf{u}}_{V|U_c}$ and $\hat{\mathbf{u}}_{U_j|U_c}$ are the pseudo copula data defined as

$$\hat{\mathbf{u}}_{V|U_c} := h_{V|U_c}(\hat{\mathbf{v}}|\hat{\mathbf{u}}_c) \quad \text{and} \quad \hat{\mathbf{u}}_{U_j|U_c} := h_{U_j|U_c}(\hat{\mathbf{u}}_j|\hat{\mathbf{u}}_c).$$

In order to obtain the desired conditional log likelihoods equation (5.7) implies that we have to estimate the pair copulas C_{VU_c} and $C_{VU_j;U_c}$ because their log likelihoods contribute to the *cll*. However, we have also to estimate the pair copulas $C_{U_cU_j}$, since they are needed to obtain the pseudo copula data $\hat{\mathbf{u}}_{U_j|U_c}$ which is used in the estimation of $C_{VU_j;U_c}$.

Pair copulas	$C_{U_2U_1}$	$C_{U_2U_3}$	$C_{U_2U_4}$	$C_{U_2U_5}$
Family	Joe	Gumbel	Gumbel	Gauss
Parameter	8.91	2.04	1.60	0.29

Pair copulas	C_{VU_2}	$C_{VU_1;U_2}$	$C_{VU_3;U_2}$	$C_{VU_4;U_2}$	$C_{VU_5;U_2}$
Family	Gumbel(180)	t	t	Gauss	Indep.
Parameter	2.79	-0.49, 2	-0.11, 5.67	0.16	0
Loglik.	327.78	123.97	12.25	6.24	0

Table 5.12: Estimated pair copulas for $c = 2$.

Pair copulas	$C_{U_1U_2}$	$C_{U_1U_3}$	$C_{U_1U_4}$	$C_{U_1U_5}$
Family	Joe	t	Gauss	Gauss
Parameter	8.91	0.86, 5.14	0.58	0.24

Pair copulas	C_{VU_1}	$C_{VU_2;U_1}$	$C_{VU_3;U_1}$	$C_{VU_4;U_1}$	$C_{VU_5;U_1}$
Family	Joe(180)	Gumbel	Frank	Frank	Frank
Parameter	3.87	4.83	-2.36	0.89	1.10
Loglik.	322.58	592.46	33.35	5.44	8.34

Table 5.13: Estimated pair copulas for $c = 1$.

The loglikehoods of the copulas $C_{U_cU_j}$ were omitted since they are not contributing to the *cll* of the models.

Following equation (5.7) we calculate then the conditional log likelihoods of the two step ahead vine models

Candidate models	$\mathcal{C}_{2,1}$	$\mathcal{C}_{2,3}$	$\mathcal{C}_{2,4}$	$\mathcal{C}_{2,5}$
cll	451.75	340.03	334.02	327.78

Candidate models	$\mathcal{C}_{1,2}$	$\mathcal{C}_{1,3}$	$\mathcal{C}_{1,4}$	$\mathcal{C}_{1,5}$
cll	**915.04**	355.93	328.02	330.92

Table 5.14: Conditional log likelihood of the candidate models.

Since the greatest *cll* is associated with the candidate model $\mathcal{C}_{1,2}$ the corresponding candidate predictor, U_1, is added first to the D-vine model. Consequently, the current optimal fit $\mathcal{C}^* \in \mathfrak{D}^1$ is the D-vine copula with order

$$\mathcal{O}(\mathcal{C}^*) = (V, U_1).$$

The *cll* of $\mathcal{C}^*$ is 322.58. Note that the *cll* of the current optimal fit $\mathcal{C}^*$ is not equal to the *cll* of the candidate model $\mathcal{C}_{1,2}$. This is due to the fact that $\mathcal{C}_{1,2}$ is the two step ahead model from the class $\mathfrak{D}^2$ and has order $\mathcal{O}(V, U_1, U_2)$, while $\mathcal{C}^*$ has only one predictor included in the order and consequently one pair copula less contributing to its *cll*.

Step 2:

In the second step, the candidates are obtained using the empirical partial correlations. Namely, we choose the $k = 2$ candidates based on $|\hat{\rho}_{V,U_j;U_1}|$ for $j = 2, 3, 4, 5$.

$\hat{\rho}_{V,U_2;U_1}$	$\hat{\rho}_{V,U_3;U_1}$	$\hat{\rho}_{V,U_4;U_1}$	$\hat{\rho}_{V,U_5;U_1}$
0.68	**-0.35**	0.13	0.12

Table 5.15: Estimated partial correlations .

The partial correlations are calculated using the function `pcor` from the package `ppcor`.

Since U_2 and U_3 have the greatest absolute partial correlations with V, they are the candidates predictors at step two and the set of candidate indices is consequently $K = \{2, 3\}$.

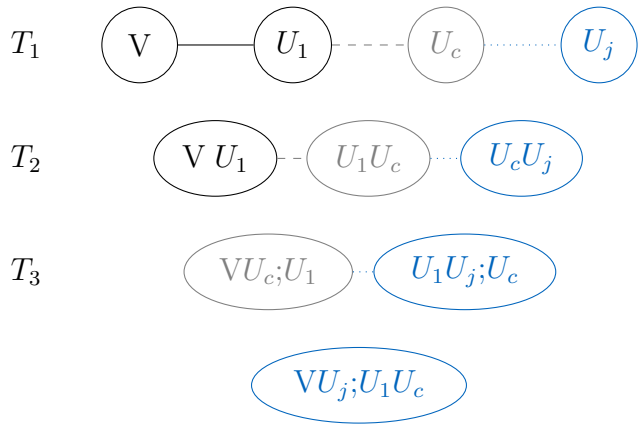

Figure 5.6: Extending the current D-vine (black) by adding candidate U_c (gray) and U_j (blue) in the 2-nd step of the algorithm. Thus, the gray and blue pair-copulas need to be estimated.

The conditional log likelihoods of interest now are those of the two-step ahead copulas $\mathcal{C}_{c,j}$ for $c \in K$ and $j \in \{2, 3, 4, 5\} \setminus \{c\}$, with order

$$\mathcal{O}\left(\mathcal{C}_{c,j}\right) = \left(V, U_1, U_c, U_j\right).$$

The *cll*'s of interest are then calculated as

$$\begin{aligned} cll\left(\mathcal{C}_{c,j}, \hat{\mathbf{v}}, \left(\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j\right)\right) = cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, \left(\hat{\mathbf{u}}_1\right)\right) + l\left(\hat{C}_{VU_c;U_1}, \left(\hat{\mathbf{u}}_{V|U_1}, \hat{\mathbf{u}}_{U_c|U_1}\right)\right) \\ + l\left(\hat{C}_{VU_j;U_1,U_c}, \left(\hat{\mathbf{u}}_{V|U_c;U_1}, \hat{\mathbf{u}}_{U_j|U_1;U_c}\right)\right). \end{aligned} \tag{5.8}$$

The pseudo copula data is obtained as

$$\hat{\mathbf{u}}_{V|U_1} := h_{V|U_1}\left(\hat{\mathbf{v}}|\hat{\mathbf{u}}_1\right), \quad \hat{\mathbf{u}}_{U_c|U_1} := h_{U_c|U_1}\left(\hat{\mathbf{u}}_c|\hat{\mathbf{u}}_1\right), \quad \hat{\mathbf{u}}_{U_j|U_c} := h_{U_j|U_c}\left(\hat{\mathbf{u}}_j|\hat{\mathbf{u}}_c\right), \tag{5.9}$$

and

$$\hat{\mathbf{u}}_{V|U_c;U_1} := h_{V|U_c;U_1}\left(\hat{\mathbf{u}}_{V|U_1}|\hat{\mathbf{u}}_{U_c|U_1}\right), \quad \hat{\mathbf{u}}_{U_j|U_1;U_c} := h_{U_j|U_1;U_c}\left(\hat{\mathbf{u}}_{U_j|U_c}|\hat{\mathbf{u}}_{U_1|U_c}\right). \tag{5.10}$$

In order to be able to estimate the pseudo copula data from equation (5.9) and (5.10) and consequently the *cll*, the pair copulas associated with the h-functions need to be estimated. Namely, the associated pair copulas with the h-functions of equation (5.9) are C_{VU_1}, $C_{U_cU_1}$ and $C_{U_jU_c}$, respectively. The associated pair copulas with the $h-$functions of equation (5.10) are $C_{VU_c;U_1}$ and $C_{U_jU_1;U_c}$ respectively.

Additionally, also the pair copulas whose log likelihood contributes to the *cll* need to be estimated. Those pair copulas are $C_{VU_j;U_1,U_c}$ and $C_{VU_c;U_1}$, as can be seen from equation (5.8).

The pair copulas C_{VU_1} and $C_{U_cU_1}$ are already estimated in step 1 and can be found in the Table 5.13. The pair copulas which need to be additionally estimated we give in the Table 5.16 and Table 5.17.

Pair copulas	$C_{U_2U_3}$	$C_{U_2U_4}$	$C_{U_2U_5}$	$C_{U_3U_1;U_2}$	$C_{U_4U_1;U_2}$	$C_{U_5U_1;U_2}$
Family	Gumbel	Gumbel	Gauss	Frank	Indep.	Indep.
Parameter	2.04	1.60	0.29	9.55	0	0

Pair copulas	$C_{VU_2;U_1}$	$C_{VU_3;U_1,U_2}$	$C_{VU_4;U_1,U_2}$	$C_{VU_5;U_2}$
Family	Gumbel	Joe	Clayton(90)	Indep.
Parameter	4.83	3.95	0.52	0
Loglik.	592.46	347.88	38.00	0

Table 5.16: Estimated pair copulas for $c = 2$.

Pair copulas	$C_{U_3U_2}$	$C_{U_3U_4}$	$C_{U_3U_5}$	$C_{U_2U_1;U_3}$	$C_{U_4U_1;U_3}$	$C_{U_5U_1;U_3}$
Family	Gumbel	Gauss	Gauss	Gumbel	Gumbel	Gauss
Parameter	2.04	0.24	0.21	3.63	2.56	0.80

Pair copulas	$C_{VU_3;U_1}$	$C_{VU_2;U_1,U_3}$	$C_{VU_4;U_1,U_3}$	$C_{VU_5;U_1,U_3}$
Family	Frank	Frank	Frank	Gumbel(180)
Parameter	-2.36	9.96	-1.30	1.10
Loglik.	33.35	303.66	10.09	7.95

Table 5.17: Estimated pair copulas for $c = 3$.

The conditional log likelihood of the two step ahead models are calculated and given in Table 5.18.

Candidate models	$\mathcal{C}_{2,3}$	$\mathcal{C}_{2,4}$	$\mathcal{C}_{2,5}$
cll	**940.33**	630.46	592.46

Candidate models	$\mathcal{C}_{3,2}$	$\mathcal{C}_{3,4}$	$\mathcal{C}_{3,5}$
cll	337.01	43.44	41.30

Table 5.18: Conditional log likelihood of the candidate models.

The greatest *cll* is associated with the candidate model $\mathcal{C}_{2,3}$. Thus, we check whether updating the current optimal fit $\mathcal{C}^* \in \mathfrak{D}^1$ with order $\mathcal{O}(\mathcal{C}^*) = (V, U_1)$ to $\mathcal{C}_2 \in \mathfrak{D}^2$ with order $\mathcal{O}(\mathcal{C}_2) = (V, U_1, U_2)$ will increase the *cll* significantly. Namely, we employ the conditional log likelihood ratio test, and since

$$cll\left(\mathcal{C}_2, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2)\right) - cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1)\right) = 592.46 > 3.84 = \chi^2_{0.95,1},$$

we reject H_0 at level 0.05. Thus, updating the optimal fit to $\mathcal{C}_2$ will improve the models fit significantly.

We update our optimal fit $\mathcal{C}^*$ to the candidate model $\mathcal{C}_2$. Thus, our estimated D-vine copula model after the second iteration step is the $\mathcal{C}^* \in \mathfrak{D}^2$ with order

$$\mathcal{O}(\mathcal{C}^*) = (V, U_1, U_2).$$

The *cll* of $\mathcal{C}^*$ is 915.04.

Step 3:

In the third step the candidates are obtained using the empirical partial correlations. Namely, we choose the $k = 2$ candidates based on $|\hat{\rho}_{V,U_j;U_1,U_2}|$ for $j = 3, 4, 5$.

$\hat{\rho}_{V,U_3;U_1,U_2}$	$\hat{\rho}_{V,U_4;U_1,U_2}$	$\hat{\rho}_{V,U_5;U_1,U_2}$
-0.11	**0.02**	-0.01

Table 5.19: Estimated partial correlations .

Since U_3 and U_4 have the greatest absolute partial correlations, they are chosen as the candidates predictors at step three and the set of candidate indices is consequently $K = \{3, 4\}$.

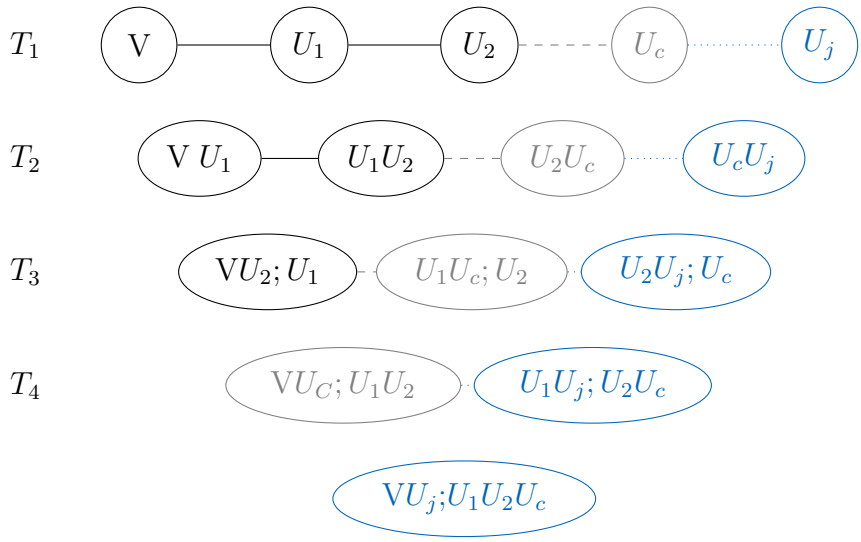

Figure 5.7: Extending the current D-vine (black) by adding candidate U_c (gray) and U_j (blue) in the 3-rd step of the algorithm. Thus, the gray and blue pair-copulas need to be estimated.

The conditional log likelihoods of interest now are those of the two-step ahead copulas $\mathcal{C}_{c,j}$ for $c \in K$ and $j \in \{3,4,5\} \setminus \{c\}$, with order

$$\mathcal{O}\left(\mathcal{C}_{c,j}\right) = \left(V, U_1, U_2, U_c, U_j\right).$$

The *cll*'s of interest are then calculated as

$$\begin{aligned} cll\left(\mathcal{C}_{c,j}, \hat{\mathbf{v}}, \left(\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j\right)\right) = {} & cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, \left(\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2\right)\right) \\ & + l\left(\hat{C}_{VU_c;U_1,U_2}, \left(\hat{\mathbf{u}}_{V|U_2;U_1}, \hat{\mathbf{u}}_{U_c|U_1;U_2}\right)\right) \\ & + l\left(\hat{C}_{VU_j;U_1,U_2,U_c}, \left(\hat{\mathbf{u}}_{V|U_c;U_1,U_2}, \hat{\mathbf{u}}_{U_j|U_1;U_2,U_c}\right)\right), \end{aligned} \tag{5.11}$$

where

$$\hat{\mathbf{u}}_{V|U_2;U_1} := h_{V|U_2;U_1}\left(\hat{\mathbf{u}}_{V|U_1}|\hat{\mathbf{u}}_{U_2|U_1}\right) \quad \text{and} \quad \hat{\mathbf{u}}_{U_c|U_1;U_2} := h_{U_c|U_1;U_2}\left(\hat{\mathbf{u}}_{U_c|U_2}|\hat{\mathbf{u}}_{U_1|U_2}\right) \tag{5.12}$$

and

$$\begin{aligned} \hat{\mathbf{u}}_{V|U_c;U_1,U_2} &:= h_{V|U_c;U_1,U_2}\left(\hat{\mathbf{u}}_{V|U_2;U_1}|\hat{\mathbf{u}}_{U_c|U_1;U_2}\right), \\ \hat{\mathbf{u}}_{U_j|U_1;U_2,U_c} &:= h_{U_j|U_1;U_2,U_c}\left(\hat{\mathbf{u}}_{U_j|U_2;U_c}|\hat{\mathbf{u}}_{U_1|U_c;U_2}\right). \end{aligned} \tag{5.13}$$

Again, similarly to step 2, the pair copulas needed to estimate the pseudo data from equations (5.12) and (5.13) as well as the pair copulas needed to calculate the *cll* from equation (5.11) have to be estimated.

Pair copulas	$C_{U_3U_2}$	$C_{U_3U_4}$	$C_{U_3U_5}$	$C_{U_3U_1;U_2}$	$C_{U_2U_4;U_3}$	$C_{U_2U_5;U_3}$
Family	Gumbel	Gauss	Gauss	Frank	Joe	Gauss
Parameter	2.04	0.24	0.21	9.55	2.68	0.23

Pair copulas	$C_{U_1U_4;U_2,U_3}$	$C_{U_1U_5;U_2,U_3}$
Family	Frank	Frank
Parameter	6.86	-0.81

Pair copulas	$C_{VU_3;U_1,U_2}$	$C_{VU_4;U_1,U_2,U_3}$	$C_{VU_5;U_1,U_2,U_3}$
Family	Joe	Clayton	Gauss
Parameter	3.95	1.72	-0.17
Loglik.	347.88	197.90	7.68

Table 5.20: Estimated pair copulas for $c = 3$.

Pair copulas	$C_{U_4U_2}$	$C_{U_4U_3}$	$C_{U_4U_5}$	$C_{U_4U_1;U_2}$	$C_{U_2U_3;U_4}$	$C_{U_2U_5;U_4}$
Family	Indep.	Gauss	Indep	Clayton(180)	Gumbel	Gauss
Parameter	0	0.24	0	0.14	2.22	0.32

Pair copulas	$C_{U_1U_3;U_2,U_4}$	$C_{U_1U_5;U_2,U_4}$
Family	Frank	Frank
Parameter	8.72	-0.69

Pair copulas	$C_{VU_4;U_1,U_2}$	$C_{VU_3;U_1,U_2,U_4}$	$C_{VU_5;U_1,U_2,U_4}$
Family	Clayton(90)	Frank	Gauss
Parameter	0.52	1.20	5.00
Loglik.	38.00	124.24	3.28

Table 5.21: Estimated pair copulas for $c = 4$.

The conditional log likelihood of the two step ahead models are calculated and given in Table 5.22.

Candidate models	$\mathcal{C}_{3,4}$	$\mathcal{C}_{3,5}$
cll	**545.78**	355.56

Candidate models	$\mathcal{C}_{4,3}$	$\mathcal{C}_{4,5}$
cll	162.24	41.28

Table 5.22: Conditional log likelihood of the candidate models.

The greatest *cll* is associated with the candidate model $\mathcal{C}_{3,4}$. Thus, we check whether updating the current optimal fit $\mathcal{C}^* \in \mathfrak{D}^2$ with order $\mathcal{O}(\mathcal{C}^*) = (V, U_1, U_2)$ to $\mathcal{C}_3 \in \mathfrak{D}^3$ with order $\mathcal{O}(\mathcal{C}_3) = (V, U_1, U_2, U_3)$ will increase the *cll* significantly. Namely, we employ the conditional log likelihood ratio test, and since

$$cll\left(\mathcal{C}_3, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_3)\right) - cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2)\right) = 347.88 > 3.84 = \chi^2_{0.95,1},$$

we reject H_0 at level 0.05. Thus, updating the optimal fit to $\mathcal{C}_3$ will improve the models fit significantly.

We update our optimal fit $\mathcal{C}^*$ to the candidate model $\mathcal{C}_3$. Thus, the current optimal fit is D-vine copula model $\mathcal{C}^* \in \mathfrak{D}^3$ with order

$$\mathcal{O}(\mathcal{C}^*) = (V, U_1, U_2, U_3).$$

The *cll* of $\mathcal{C}^*$ is 1262.92.

Step 4:

Now only two more possible predictors are left, and since the number of possible predictors is less or equal to k we do not calculate the partial correlation. Instead we take all remaining possible predictors as candidates. Thus, now $K = \{4, 5\}$.

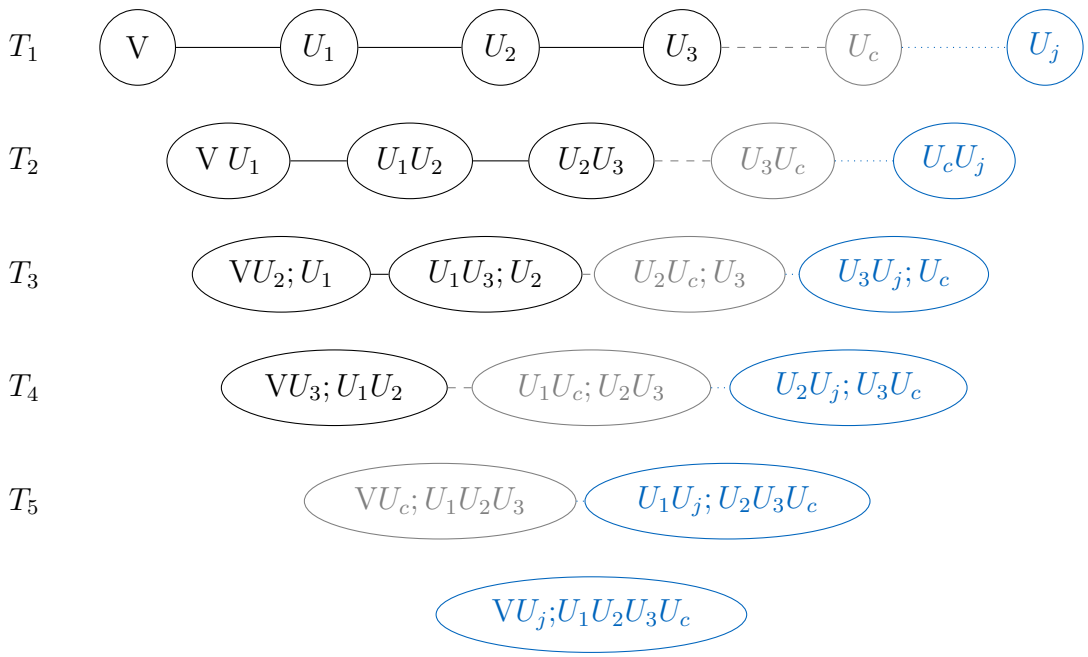

Figure 5.8: Extending the current D-vine (black) by adding candidate U_c (gray) and U_j (blue) in the 4-th step of the algorithm. Thus, the gray and blue pair-copulas need to be estimated.

The conditional log likelihoods of interest now are those of the two-step ahead copulas $\mathcal{C}_{c,j}$ for $c \in K$ and $j \in \{4,5\} \setminus \{c\}$, with order

$$\mathcal{O}\left(\mathcal{C}_{c,j}\right) = \left(V, U_1, U_2, U_3, U_c, U_j\right).$$

The *cll*'s of interest are then calculated as

$$\begin{aligned} cll\left(\mathcal{C}_{c,j}, \hat{\mathbf{v}}, \ (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_3, \hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j)\right) = \ & cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_3)\right) \\ & + l\left(\hat{C}_{VU_c;U_1,U_2,U_3}, \left(\hat{\mathbf{u}}_{V|U_3;U_1,U_2}, \hat{\mathbf{u}}_{U_c|U_1;U_2,U_3}\right)\right) \\ & + l\left(\hat{C}_{VU_j;U_1,U_2,U_3,U_c}, \left(\hat{\mathbf{u}}_{V|U_c;U_1,U_2,U_3}, \hat{\mathbf{u}}_{U_j|U_1;U_2,U_3,U_c}\right)\right). \end{aligned}$$

As in the previous steps, numerous pair copulas have to be estimated in order to obtain the *cll* despite only two pair copulas contributing to the *cll*. Again, most of the pair copulas are estimated only to obtain the needed pseudo observations. We give all additionally estimated pair copulas in Table 5.23.

Pair copulas	$C_{U_4U_2;U_3}$	$C_{U_5U_3;U_4}$	$C_{U_4U_1;U_2,U_3}$	$C_{U_5U_2;U_3,U_4}$	$C_{U_5U_1;U_2,U_3,U_4}$
Family	Joe	Gauss	Frank	Gumbel	Indep.
Parameter	2.68	0.21	6.86	1.23	0

Pair copulas	$C_{VU_4;U_1,U_2,U_3}$	$C_{VU_5;U_1,U_2,U_3,U_4}$
Family	Clayton	Indep.
Parameter	1.72	0
Loglik.	197.90	0

Table 5.23: Estimated pair copulas for $c = 4$.

Pair copulas	$C_{U_5U_2;U_3}$	$C_{U_4U_3;U_5}$	$C_{U_5U_1;U_2,U_3}$	$C_{U_4U_2;U_3,U_5}$	$C_{U_4U_1;U_2,U_3,U_5}$
Family	Gauss	Gauss	Frank	Clayton(180)	Frank
Parameter	0.23	0.24	-0.81	1.86	6.17

Pair copulas	$C_{VU_5;U_1,U_2,U_3}$	$C_{VU_4;U_1,U_2,U_3,U_4}$
Family	Gauss	Frank
Parameter	-0.17	-2.82
Loglik.	7.68	48.36

Table 5.24: Estimated pair copulas for $c = 5$.

The conditional log likelihoods of the candidate models are given in Table 5.25.

Candidate models	$\mathcal{C}_{4,5}$	$\mathcal{C}_{5,4}$
cll	**197.90**	56.04

Table 5.25: Conditional log likelihood of the candidate models.

Once more we employ the conditional log likelihood ratio test. We check whether updating the current optimal fit to the candidate model $\mathcal{C}_4 \in \mathfrak{D}^4$ with order $\mathcal{O}(\mathcal{C}_4) = (V, U_1, U_2, U_3, U_4)$. Since it holds that

$$\begin{aligned} cll\left(\mathcal{C}_4, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_3, \hat{\mathbf{u}}_4)\right) - cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_3)\right) &= 197.90 \\ &> 3.84 = \chi^2_{0.95,1}, \end{aligned}$$

the test implies that U_4 should be added to the model.

We update our optimal fit $\mathcal{C}^*$ to the candidate model $\mathcal{C}_4$. Thus, the current optimal fit is D-vine copula model $\mathcal{C}^* \in \mathfrak{D}^4$ with order

$$\mathcal{O}(\mathcal{C}^*) = (V, U_1, U_2, U_3, U_4).$$

The *cll* of $\mathcal{C}^*$ is 1460.82.

Step 5:

At this point only one predictor is remaining. Thus, we can not calculate any two step ahead *cll*'s and the only thing left is to decide whether to add U_5 to the model or not.

The copula $\hat{C}_{VU_5;U_1,U_2,U_3,U_4}$ is estimated to be the Independence copula as can be seen in Table 5.23.

Thus, adding U_5 to the model will not change the conditional log likelihood and therefore we do not include U_5 in the model.

Output: The optimal D-vine copula model $\mathcal{C}^* \in \mathfrak{D}^4$, lastly updated at Step 4 and has order

$$\mathcal{O}(\mathcal{C}^*) = (V, U_1, U_2, U_3, U_4),$$

with conditional log likelihood

$$cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_3, \hat{\mathbf{u}}_4)\right) = \mathbf{1460.82}. \tag{5.14}$$

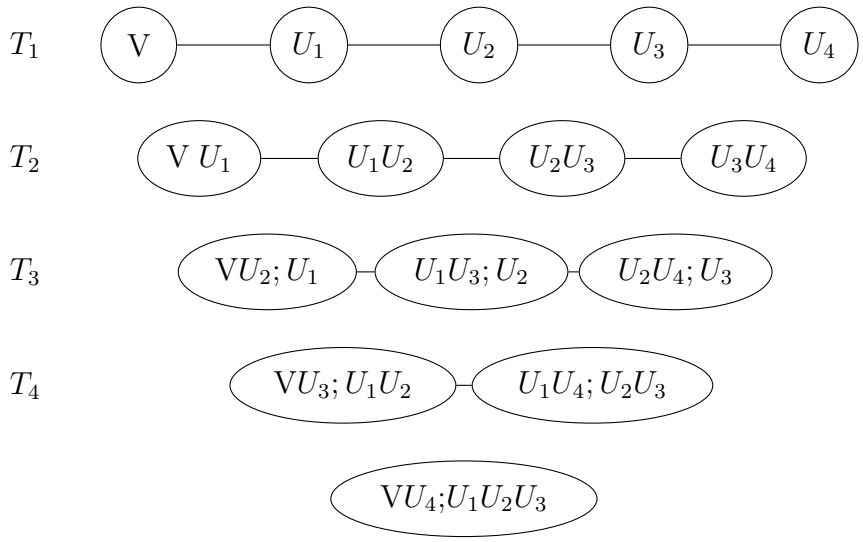

Figure 5.9: D-vine tree sequence of the optimal fit.

5.3 Note on the Illustrations of Algorithm 1 and 2

Let us denote the optimal fit of the D-vine quantile regression model, from subsection 5.1.1, obtained by Algorithm 1 ("Forward selection of C- and D-vine quantile regression models" Section 5.1) as

$$\mathcal{C}_{A1} \in \mathfrak{D}^3, \text{ with order } \mathcal{O}(\mathcal{C}_{A1}) = (V, U_2, U_1, U_3),$$

and the optimal fit, from subsections 5.2.1, obtained by Algorithm 2 ("Two step ahead forward selection of C- and D-vine quantile regression models" Section 5.2) from subsections 5.2.1 as

$$\mathcal{C}_{A2} \in \mathfrak{D}^4, \text{ with order } \mathcal{O}(\mathcal{C}_{A2}) = (V, U_1, U_2, U_3, U_4). \tag{5.15}$$

The data was simulated from the six dimensional D-vine copula, $\mathcal{C}_6$, whose pair copulas given in Table 5.1. The order of the nodes in the first tree of $\mathcal{C}_6$ is

$$V - U_1 - U_2 - U_3 - U_4 - U_5.$$

From Table 5.1 we see that the first four covariates, U_1, U_2, U_3 and U_4, have a strong influence on V and thus, should be included in a good model. On the other hand the influence of U_5 is rather small, especially conditioned on the first four covariates, and can be left out.

Algorithm 2 follows exactly what we described about the D-vine copula $\mathcal{C}_6$. The four important predictors, U_1, U_2, U_3 and U_4, in the data were identified by Algorithm 2

and U_5 was left out as it was not increasing the *cll* significantly. Even the order of the predictors in the first tree of the D-vine quantile regression model $\mathcal{C}_{A2}$ is identical to the order of covariates in the D-vine copula $\mathcal{C}_6$, as indicated by its order in equation (5.15).

On the other hand, Algorithm 1 failed to identify one of the important predictors. From its order it is also apparent that the order of predictors included in the D-vine quantile regression model $\mathcal{C}_{A1}$ is different from the order of covariates in the first tree of $\mathcal{C}_6$.

Clearly Algorithm 2 has produced a better fit of the D-vine copula $\mathcal{C}_6$ than Algorithm 1. That is also confirmed by the conditional log likelihoods of the respective models, given in equations (5.4) and (5.14),

$$cll\left(\mathcal{C}_{A1}\right) = 467.84 \text{ and } cll\left(\mathcal{C}_{A2}\right) = 1460.82.$$

Algorithm 2 achieved an almost 3 times better conditional log likelihood.

In Appendix A detailed illustrations of the estimation procedures for both proposed forward selection methods for C-vine copulas are provided on five dimensional Gaussian data sets.

We conclude that there are data sets on which the two step ahead forward selection of C- and D-vine quantile regression models works better that the forward selection of C- and D-vine quantile regression models. We will further investigate the performance of both methods in the simulation study of Part III.

Chapter 6

Forward selection algorithms and large data sets

6.1 Computational complexity

The earlier proposed algorithms both work on a principle of sequentially growing the models at each step. This means, instead of estimating regular vine copulas at each step, the algorithms are just extending the already existing vine copula from the step before. The extensions at each step only require additional estimation of the bivariate copulas. In order to be able to quantify and compare the computational complexity, we express it as the number of pair copulas that need to be estimated during each approach. A valid representation of the computational complexity of each algorithm is obtained since the complexity of bivariate copula estimation, given in Section 2.5, remains the same regardless of the algorithm used. Now, we express the computational complexity of finding the *cll* optimal fit, as well as those of "Algorithm 1: Forward selection of C- and D-vine quantile regression models" (Section 5.1) and "Algorithm 2: Two step ahead forward selection of C- and D-vine quantile regression models" (Section 5.2) , in terms of the number of pair copulas to be estimated. For this purpose, we consider a C- or D-vine quantile regression model on p predictors.

- ***cll*-optimal fit:** As already stated, to compute the *cll*-optimal fit one needs to estimate $p!$ C- or D-vine copulas. Since for each p dimensional vine copula, $\frac{p(p-1)}{2}$ pair copulas are estimated, the overall computational complexity is then

$$p! \cdot \frac{p\,(p-1)}{2}.$$

- **Algorithm 1:** The first algorithm computes at the r-th step the *cll*-optimal submodel on r predictors given that the first $r-1$ predictors are chosen already. In order to do that the model needs to be extended by one predictor for each of the remaining predictors. To obtain the desired extension for one of the remaining predictors r pair copulas need to be estimated. Since there are $p-(r-1)$ choices for the extension, one has to estimate in total $r \cdot (p-r+1)$

at step r. All together the computation complexity is

$$\sum_{r=1}^{p} r \cdot (p - r + 1), \tag{6.1}$$

which is equivalent to:

$$\frac{1}{6}p^3 + \frac{1}{2}p^2 + \frac{1}{3}p. \tag{6.2}$$

- **Algorithm 2:** The second algorithm on the other hand computes at each step r the *cll*-optimal submodel on $r+1$ predictors given that the first $r-1$ predictors are chosen already. For this first k candidates are chosen based on partial correlations. We suppose that the estimation of margins is computationally negligible compared to the bivariate copula selection. Further, for each of the k candidates the model is extended one step further for each of the remaining non-chosen predictors. That means, to estimate a two step ahead model one needs a total of

$$\sum_{r=1}^{p-1} (2r + 1) \cdot \min\{k, p - r + 1\} \cdot (p - r) \quad + \quad 1 \tag{6.3}$$

bivariate pair copula selections. In equation (6.3), r represents the number of bivariate copulas required to be selected at step r. The term $\min\{k, p - r + 1\}$ represents the number of candidates at each step. The minimum is taken to avoid instabilities in the case when the number of remaining covariates is less than the number of candidates and one can not choose k candidates. Finally, $p - r$ corresponds to the number of remaining predictors at step r, without the candidate for which the two step ahead extension is being estimated for. Equation (6.3) is equivalent to:

$$\frac{p^3k}{3} + \frac{p^2k}{2} - \frac{pk^3}{3} + pk^2 - \frac{3pk}{2} + \frac{k^4}{6} - \frac{5k^3}{6} + \frac{4k^2}{3} - \frac{2k}{3}. \tag{6.4}$$

From equations (6.2) and (6.4) we can see that the estimation complexity of Algorithms 1 and 2, in terms of bivariate copula selection, are $O(p^3)$ and $O(p^3k + k^4)$, respectively. This means that, in terms of the number of predictors, p, both algorithms are of order p^3. However, an increase in the number of predictors will more heavily impact Algorithm 2, due to the presence of k multiplying the term p^3 in its computational complexity. On the other hand, a full maximum likelihood approach will have complexity $O(p!)$. As already discussed before, such an approach would only be feasible if $p < 10$, or similar.

Num. of predictors p	5	10	15	20	50	100
cll-optimal fit	1200	163 296 000	-	-	-	-
Algorithm 1	35	220	680	1 540	22 100	171 700
Algorithm 2	180	1 705	5 855	13 880	213 405	1 689 280

Table 6.1: Number of estimations of bivariate copulas for $p = 5, 10, 15, 20, 50, 100$, $k = 5$.

In Table 6.1 we see how quickly the number of pair copula estimations rises with the number of predictors for both algorithms. Despite the low computational intensity of the pair copula estimation process, the high number of estimations needed, especially for Algorithm 2, can become a problem when the number of predictors is big. As already discussed before, efficient methods for the estimation of pair copulas are developed and implemented. However their computational complexity depends on the sample size n. An increase in the sample size will inevitably lead to a proportional increase in the computational time of the pair copula selection. This implies that also the computational complexity of our algorithms will greatly depend on the sample size. As it is the case with the *cll*-optimal fit, also Algorithms 1 and 2 eventually become computationally too intensive for big values of p and n.

As an example, fitting a pair copula for the data vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ with 300 rows each requires approximately 0.3 seconds on a commercial machine. On a data set with 300 data points this means, disregarding everything but the pair copula estimation, that Algorithm 1 would take at least 15 minutes to estimate a model on 20 predictors, while for 50 predictors the estimation process would last about 2 hours. It gets even worse for Algorithm 2, where one needs 1 hour for 20 predictors and almost 10 hours for 50 predictors. Data sets with more than 100 predictors would as a reference take more than 24 hours to compute the model.

6.2 Batch algorithms

Modern day big data sets contain hundreds of predictors and up to a million data points. Obviously, in such scenarios the proposed algorithms are computationally too intensive. Therefore, we want propose an alternative way of estimating models from the classes $\mathfrak{D}^p$ and $\mathfrak{C}^p$, when p is big. Ideally we would like to make the computational complexity independent from p. This would allow us to chose L out of p predictors in a time independent of p, but only dependent on the number of candidates k, sample size n and possibly other hyper-parameters.

With the intent to optimize the computational complexity of such algorithms, we consider the two step ahead forward selection algorithm from Part 5. Considering an iteration step r of the two step ahead forward selection, we have the set of candidates $\{U_i\}_{i \in K}$ and the set of remaining predictors $\{U_i\}_{i \in R}$ where

$$K := \{p_1, \ldots, p_k\} \quad \text{and} \quad R := \{1, \ldots, p\} \setminus \{t_1, \ldots, t_{r-1}\},$$

where $\{t_1, \ldots, t_{r-1}\}$ represent the indexes of the predictors chosen up to step r. Additionally, we define the set R_{-c} as

$$R_{-c} := R \setminus \{c\}.$$

To include the next predictor in the model, the *cll*'s of the two step ahead models $\mathcal{C}_{c,j}$ with order

$$\mathcal{O}\left(V, U_{t_1}, \ldots, U_{t_{r-1}}, U_c, U_j\right) \quad c \in K, \quad j \in R_{-c}$$

are computed. The next predictor included in the model is chosen based on the *cll* of the two step ahead models $\mathcal{C}_{c,j}$. To extend the current optimal fit of step r, $\mathcal{C}^*$, to $\mathcal{C}_{c,j}$ for given c and j, additional $r+2$ pair copulas need to be fitted. Thus, to compute the *cll* for all two step ahead models $\mathcal{C}_{c,j}$ at step r one has to fit a total

$$(2r+1) \cdot |K| \cdot |R_{-c}|$$

pair copulas, as already indicated by equation (6.3). Obviously, the number of pair copulas needed for the extensions can not be changed. Furthermore, since we already control the size of K, the only way to reduce the computational complexity of the algorithm is to reduce the size of R_{-c}. With the ideas to approximate the two step ahead algorithm while reducing the computational complexity we introduce two new algorithms:

- **Batch maximum algorithm:** Here the idea is instead of computing all possible models $\mathcal{C}_{c,j}$ for $c \in K$ and $j \in R_{-c}$ at a step r, to compute the models

  $$\mathcal{C}_{c,i}, \quad \text{for} \quad c \in K, \quad i \in B_c,$$

  where B_c is a random subsample of R_{-c} of size b. We define $m\mathcal{C}_c$ as the maximal two step ahead *cll* for each of the candidate predictors, i.e.,

  $$m\mathcal{C}_c := \max_{i \in B_c} cll\left(\mathcal{C}_{c,i}\right), \ \forall c \in K.$$

  Then, the index of the next predictor to be added, t_r, is obtained as

  $$t_r := \arg\max_{c \in K} \ m\mathcal{C}_c.$$

- **Batch average algorithm:** Similarly as with the batch maximum approach, instead of all possible models $\mathcal{C}_{c,j}$ at step r, only the models

  $$\mathcal{C}_{c,i}, \quad \text{for} \quad c \in K, \quad i \in B_c,$$

  where B_c is a random subsample of R_{-c} of size b, are computed. Further, we define the average conditional log likelihood of a candidate predictor U_c as

  $$acll\left(U_c\right) = \frac{1}{b} \sum_{i \in B_c} cll\left(\mathcal{C}_{c,i}\right).$$

  Finally, the index of the next predictor to be added, t_r, is obtained as

  $$t_r := \arg\max_{c \in K} \ acll\left(U_c\right).$$

The above defined Batch maximum and Batch average algorithms estimate the *cll* optimal fit for step r on the space $K \times R_{-c}$ with the *cll* optimal fit on the subspace $K \times B_c$ for each candidate. Furthermore, these algorithms also solve the problem of computational complexity to some extend. The number of pair copulas to be estimated at step r, for both of the algorithms, is given by

$$(2r+1) \cdot |K| \cdot |B| = (2r+1) \cdot k \cdot b.$$

This means, to estimate a C- or D-vine model for quantile regression on p covariates with the above defined algorithm, a total of

$$\sum_{r=1}^{p-1} (2r+1) \cdot b \cdot \min\{k, p-r+1\} \;\; + \;\; 1 \;\; \approx \sum_{r=1}^{p-1} (2r+1) \cdot b \cdot k \tag{6.5}$$

pair copula fits is needed. With this approach the computational complexity, in terms of the number of pair copula fits, reduces to $O\left(p^2 k b\right)$.

At first, this is not a big reduction of computational complexity since for huge data sets it will still be hardly tractable. However, up until now we only considered the case when we want to model the response with all predictors. Often this is not the case, especially when several hundreds of possible predictors are available, and one want to model the response only with the most influential L predictors. In this case, the computational complexity will be appropriately smaller. The exact number of pair copula estimations needed if there is L out of p predictors to be chosen can be obtained by changing the upper summation limits in equations (6.1), (6.3) and (6.5) to L. This yields computational complexities of

$$O\left(L^2 p\right), \;\; O\left(L^2 p k + k^4\right) \;\; \text{and} \;\; O\left(L^2 k b\right) \tag{6.6}$$

for Algorithm 1, Algorithm 2 and the batch algorithms, respectively. From equation (6.6) we can see that the complexity of the Batch algorithms is the only one independent of p. Thus, both of the above define Batch algorithms give an approximation of the Two step ahead forward selection of C- and D-vine quantile regression models (Section 5.2). For the values of $p < 100$ one can apply Algorithm 1 and Algorithm 2 with enough resources. But for larger values of p Algorithms 1 and 2 can quickly become intractable. On the other hand, if one wants to chose only L predictors out of p one can do this approximately with the same complexity for arbitrary values of p, when keeping k and b constant.
However, the Batch algorithms are approximating Algorithm 2 and the quality of the approximation depends on the subsample sizes b chosen. But, if L and b can be kept small the Batch algorithms can handle, in terms of computational complexity, data sets with p up to 500.

Suppose we have a data set with up to 500 possible predictors, a sample size of $n < 2000$, further we set the batch size to $b = 50$ and the number of candidates to $k = 25$. Additionally, suppose a pair copula estimation time of 0.1 seconds, which is available on commercial machines. The Batch algorithms could efficiently determine the 20 to 30 most important predictors with a computational time of about

12 hours. If p is smaller then one can decrease b and increase n while keeping the same computational complexity. With the rapid increase of computational power and cluster computing becoming more popular, both the Batch algorithms as well as the Forward selection algorithms, can be parallelized and with enough resources computed in hours for an even bigger set up then the one described before.

It is clear that in terms of computational complexity the Batch algorithms are preferable to Algorithm 2 and even Algorithm 1 with bigger data sets. We have not yet discussed the performance in terms of cll, order and other goodness of fit measures, of the Algorithms 1 and 2, which is further discussed in the simulation studies in Part III.

Part III

Simulation study

Chapter 7

Introduction to simulation study

After defining the two algorithms for an automated forward selection of C- and D-vine quantile regression models from Chapter 5 and the batch algorithms from Section 6.2, we continue with comparing the performance of the algorithms through a simulation study. Before we formally define the set up of our simulation studies, we introduce the notation for the algorithms considered.

1. **Algorithm 1**. Forward selection of C- and D-vine quantile regression models (Section 5.1).

   Indices: $D - A1$ for D-vine models, $C - A1$ for C-vine models.

2. **Algorithm 2**. Two step ahead forward selection of C- and D-vine quantile regression models (Section 5.2).

   Indices: $D - A2$ for D-vine models, $C - A2$ for C-vine models.

3. **Batch maximum algorithm**. Two step ahead forward selection of C- and D-vine quantile regression models with maximum batch (Section 6.2).

   Indices: $D - Bmax$ for D-vine models, $C - Bmax$ for C-vine models.

4. **Batch average algorithm**. Two step ahead forward selection of C- and D-vine quantile regression models with average batch (Section 6.2).

   Indices: $D - Bavg$ for D-vine models, $C - Bavg$ for C-vine models.

5. **Linear quantile regression** (introduces in Section 3.3).

   Index: $QLin$.

The simulation study is divided in 2 different case studies, in which we compare the algorithms performance based on several measures. In the first case study, we simulate the data from a D-vine copula distribution and apply the D-vine based algorithms, while in the second case study, we simulate data from a C-vine copula distribution and apply the corresponding C-vine based algorithms. Detailed description of the simulation methods is given in Section 2.4.

Given a data set D of size n, within a simulation scenario, we divide it into two subsets, training and testing set. The training set D_{train} is used for model selection. On the testing set, D_{test}, the best chosen model for the training set, is being applied. The performance on the testing set is an indication how good the models estimated on the training set can generalize, i.e., how good the models will perform on new data. Given a C- or D-vine copula distribution $\mathcal{C}$ on p possible continuous predictors $X_1, \ldots, X_p$ for continuous response variable Y, a replication $k = 1, \ldots r$ within a scenario consists of simulating a data set $D^{(k)}$, of size n, from the copula distribution $\mathcal{C}$. Further, we split $D^{(k)}$ into

$$D_{train}^{(k)} = \left(\left\{ x_{i,j,train}^{(k)} \right\}_{i=1,\ldots,p;\, j=1,\ldots,n_{train}}, \left\{ y_{j,train}^{(k)} \right\}_{j=1,\ldots,n_{train}} \right),$$

where $n_{train} = |D_{train}|$ and

$$D_{test}^{(k)} = \left(\left\{ x_{i,j,test}^{(k)} \right\}_{i=1,\ldots,p;\, j=1,\ldots,n_{test}}, \left\{ y_{j,test}^{(k)} \right\}_{j=1,\ldots,n_{test}} \right),$$

with $n_{test} = |D_{test}^{(k)}|$. We test the performance of the competing algorithms based on several goodness of fit measures.

Conditional log likelihood is used as a goodness of fit measure of the models obtained from different algorithms as defined in Definition 4.8.

Order of the models obtained from the vine based algorithms is compared with the order of the nodes in the vine copula from which the data was simulated. The order of a C- or D-vine model is defined in Definition 4.5.

Out of sample mean square error (OMSE) Given an estimated quantile regression model $\hat{\mathcal{C}}$, its OMSE is computed as

$$OMSE(\hat{\mathcal{C}}, \alpha) = \frac{1}{r} \sum_{k=1}^{r} \left[\frac{1}{n_{test}} \sum_{j=1}^{n_{test}} \left(\hat{q}_\alpha(x_{1,j,test}^{(k)}, \ldots, x_{p,j,test}^{(k)}) - q_\alpha(x_{1,j,test}^{(k)}, \ldots, x_{p,j,test}^{(k)}) \right)^2 \right],$$

where $\hat{q}_\alpha$ is the α-level prediction function of the quantile regression model $\hat{\mathcal{C}}$ and q_α is the true quantile function.

In sample mean square error (IMSE) Given an estimated quantile regression model $\hat{\mathcal{C}}$, its IMSE is computed as

$$IMSE(\hat{\mathcal{C}}, \alpha) = \frac{1}{r} \sum_{k=1}^{r} \left[\frac{1}{n_{train}} \sum_{j=1}^{n_{train}} \left(\hat{q}_\alpha(x_{1,j,train}^{(k)}, \ldots, x_{p,j,train}^{(k)}) - q_\alpha(x_{1,j,train}^{(k)}, \ldots, x_{p,j,train}^{(k)}) \right)^2 \right],$$

where $\hat{q}_\alpha$ is the α-level prediction function of the quantile regression model $\hat{\mathcal{C}}$ and q_α is the true quantile function.

Computational time of the estimation procedures of every algorithm are compared.

All computations in the following case studies are conducted using the statistical language `R`. The one step ahead forward selection of D-vine regression models is already implemented in the library `vinereg`. For the purpose of the following case studies we developed special libraries for both the two step ahead and one step ahead C- and D-vine algorithms, following their corresponding definitions from Sections 5.1 and 5.2. The linear quantile regression implementation can be found in the package `quantreg`.

Chapter 8

Simulation studies

8.1 Case Study I: D-vine regression models

In the first case study, we sample a data set D of $n = 400$ data points from a random vector $(V, U_1, \ldots, U_{20})$ following a 21-dimensional D-vine copula distribution $\mathcal{C}_{21}$ with order

$$\mathcal{O}\left(V, U_1, U_2, U_3, \ldots U_{20}, U_{21}\right). \tag{8.1}$$

To make sure there is significant dependence between the response variable and the predictors we define the following pair copulas of the D-vine $\mathcal{C}_{21}$

Pair copula	Family	Kendall's τ
C_{V,U_1}	Frank	0.7
$C_{V,U_2;U_1}$	Clayton	0.7
$C_{V,U_3;U_2,U_1}$	Gumbel	0.7
$C_{V,U_4;U_3,U_2,U_1}$	Frank	0.3
$C_{V,U_5;U_4,U_3,U_2,U_1}$	Clayton	0.5
$C_{V,U_6;U_5,U_4,U_3,U_2,U_1}$	Clayton	0.4
$C_{V,U_7;U_6,U_5,U_4,U_3,U_2,U_1}$	Frank	0.4

Table 8.1: Pair copulas defining the dependence between the response and the first seven predictors.

For the dependence in-between the first ten predictors of the order, the pair copulas were sampled from a set of families containing the Clayton, Gumbel and Frank copula with parameters corresponding to Kendall's τ between 0.2 and 0.6, except the pair copulas already defined in Table 8.1 . All other pair copulas were sampled from a set containing Gaussian copulas with $\tau < 0.15$ and the independence copula. These choices were maintained for all replications

After obtaining the data set D we split it into a training set, D_{train} and a testing set, D_{test}, of sizes 300 and 100, respectively. In this case study, we employ the four algorithms developed in Chapter 7 and linear quantile regression from Section 3.3. All algorithms, from Chapter 7, were set to choose a model from the D-vine quantile regression class $\mathfrak{D}^{21}$ and include all possible predictors. The bivariate pair copula

selection criteria is set to maximum log likelihood and no penalization on the condition log likelihood is used. The number of candidates for the two step ahead forward selection is set to $k = 5$ and the batch size for the maximum and average batch method is $b = 5$. All together $r = 20$ replication of case study one were run with seeds from 1 to 20 and all the results are averaged over the 20 iterations.

In Figure 8.1, a pairs plot is given of the train data set $D^{(9)}_{train}$, where k indicates the number of iteration and the seed number, as they are the same.

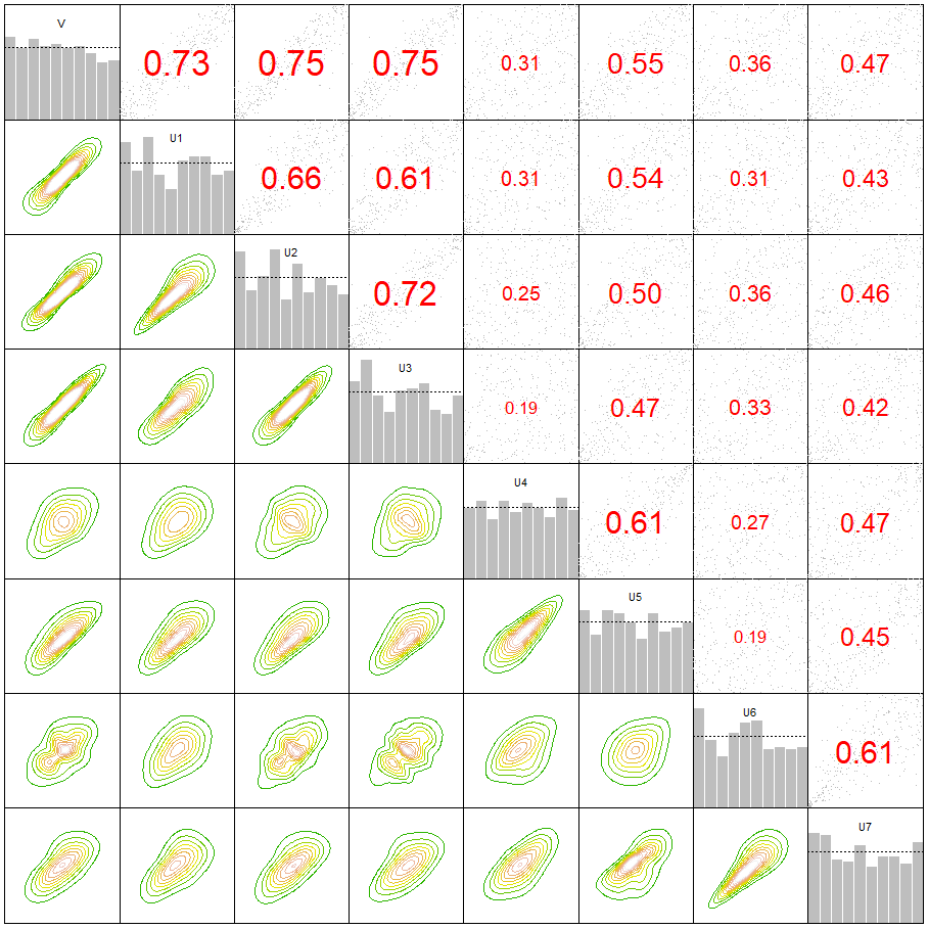

Figure 8.1: Pairs copula plot between the response and the predictors $U_1, \ldots, U_7$ of data set $D^{(9)}_{train}$, with contour plots of the pair copulas, Kendall's tau values and u-plots.

First, we look at the conditional log likelihood of the four competing algorithms for D-vine quantile regression. The *cll* of each individual model is computed at each of the 20 iterations and an average is provided in Table 8.2. A higher value of the *cll* indicates a better fit of the model.

Algorithm	$D-A1$	$D-A2$	$D-Bmax$	$D-Bavg$
Max *cll*	537.98	**778.45**	738.11	549.11
Min *cll*	416.68	**524.88**	424.76	399.89
Mean *cll*	495.85	**695.68**	589.80	485.75
Standard deviation of *cll*	**33.06**	60.26	120.94	41.26

Table 8.2: *cll* statistics of Algorithms 1-4 from Case study I.

From Table 8.2 we can see that in this set up, regarding the conditional log likelihood, the two step ahead forward selection algorithm, labeled $D-A2$, has shown the best result. We see that the $D-A2$ algorithm has on average the greatest *cll* as well as the greatest maximal and greatest minimal *cll* over the 20 iterations. For the other algorithms, the batch maximal algorithm has the next best *cll* statistic. The worst models, regarding the *cll* are the one step ahead forward selection and the batch average algorithm. Regarding the standard deviations of the conditional log likelihoods, the algorithms $D-A1$, $D-A2$ and $D-Bavg$ showed small values and are consistently achieving a similar *cll* on average. Algorithm $D-Bmax$ showed a considerably bigger standard deviation than the other three algorithms and nevertheless, scored a better *cll* statistics than two of the tree more stable algorithms.

The AIC penalized *cll* values show a similar behavior as the *cll*. From Table 8.3 we see that, according to the AIC penalized *cll* values as the goodness of fit measure, the two step ahead forward selection has delivered the best performing models. For the cll_{AIC}, smaller values indicate better performing models. One additional inter-

Algorithm	$D-A1$	$D-A2$	$D-Bmax$	$D-Bavg$
Max cll_{AIC}	-181.35	**-398.38**	-177.25	-133.35
Min cll_{AIC}	-441.35	**-894.07**	-809.84	-437.64
Mean cll_{AIC}	-336.20	**-721.13**	-511.68	-298.68
Standard deviation of cll_{AIC}	**70.47**	119.89	247.84	84.92

Table 8.3: cll_{AIC} statistics of Algorithms 1-4 from Case study I.

esting thing is that according to both measures the batch maximum algorithm was able to provide some of the best performing fits in certain iterations, while in other iterations it delivered some of the worst models. With best and worst performing models we refer to the models with highest and lowest *cll* values. The one step ahead algorithm and the batch average algorithm were not able to find these fits, as can be seen from the maximum and minimum *cll* and cll_{AIC} values in Tables 8.2 and

8.3.

Next, we look at the order of the estimated D-vine regression models obtained from algorithms 1-5, respectively. We know from the underlying distribution that the predictors U_1, U_2, U_3 have an approximate dependence on V of 0.75, in terms of Kendall tau values, and thus, should be included relatively early in the order of a good fit. Additionally, U_1, U_2, U_3 are the next connecting nodes to V in the true underlying D-vine tree sequence, which is also an indicator of strong dependence and predicting power. The next of importance, considering the same criteria as for U_1, U_2, U_3, is the group U_4, U_5, U_6, U_7. To asses how good the order of each fit is, we will count for each algorithm how many times the group $G_1 = (U_1, U_2, U_3)$ was within the first 3 elements of the true order, given in equation (8.1), for each iteration. For the group $G_2 = (U_4, U_5, U_6, U_7)$ we will count how many times the members of G_2 are among the first 7 predictors in the order obtained by the algorithms. Additionally, we count the number of times the true order of the underlying D-vine tree sequence of $\mathcal{C}_{21}$, i.e. $(V, U_1, U_2, \ldots, U_{19}, U_{20})$, has been reproduced by the models up to the first 3 elements. Finally, the number of different orders on the first 4, 5 and 6 predictors of the models has been counted.

Algorithm	$D-A1$	$D-A2$	$D-Bmax$	$D-Bavg$
G_1 among first 3 predictors	14	20	15	19
G_2 among first 7 predictors	12	6	6	1
# misspecified first predictors	20	2	9	20
# of true orders < 3 predictors	0	2	8	0
# of true orders > 3 predictors	0	16	3	0
Different orders up to 4	4	5	8	5
Different orders up to 5	12	9	13	7
Different orders up to 6	13	15	18	17

Table 8.4: Order statistics of the fits obtained from Algorithms 1-4 from Case study I.

Considering only the first 7 elements from the orders of the fitted models is motivated by two reasons. First, from the underlying D-vine distribution of the simulated data we know that only 7 of the 20 possible predictors have a significant dependence with the response. The second source of motivation comes from Definition 4.8, which defines a split of the *cll* into summands of log likelihoods of the conditional distributions $c_{V|U_1,\ldots,U_i}$ for $i = 1, \ldots 20$. As an example, the *cll* of the two step ahead model from iteration 14 can be split up into log likelihoods of the corresponding pair copulas as

$$cll(M^{(14)}_{D-A2}) = 208.95 + 180.68 + 154.15 + 155.56 + 12.26 + 12.37 + 17.48 + 2.38 + \ldots$$

where the first summand corresponds to the log likelihood of the conditional distribution $c_{V|U_1}$, the second to the log likelihood of $c_{V|U_1,U_2}$ and so on, details can be found in Table B.1. In this particular example, the first 8 log likelihoods contributed to over 97% of the total 752.12 *cll*. A similar behavior is seen across all models in this set up which indicates that most models are giving the highest weight to the first few predictors chosen in the order. Since the predictors after the seventh or eight place in the order have no significant effect on the *cll* for most models, we can conclude that there is almost no information loss when considering only the first seven predictors of the order of each model.

Additionally, the orders of the fitted models also indicate how consistent the algorithm is. Similar orders reproduced many times in the iterations are a strong sign that the algorithm is very consistent. From Table 8.4 we see that for the first 4 predictors all algorithms have only 4 or 5 different orders, except the batch maximum algorithm which had already 8 different orders on the four predictors. Further, from Table 8.4 we see that all algorithms have included almost every time the predictors of group G_1 as the first three predictors in the models. Thus, all algorithms are very consistent in including the most important predictors into the models very early and therefore giving them the greatest importance in the model. The orders between algorithms were very different, but for a given algorithm and for the first four predictors the orders of the obtained models were consistent and had minimal variation. Next, the order of the nodes in the first tree of the underlying D-vine distribution is compared to the order of predictors in the obtained models. The one step ahead regression model did not match the true order at any iteration due to the fact that either U_2 or U_3 was chosen as the first predictor of the order. On the other hand, the two step ahead matched the order of the first 5 predictors to the true order in 12 out of the 20 iterations. The batch algorithms have more inconsistencies in their orders, which was expected due to the additional randomness added in the estimation procedure. For further analysis, the first seven elements of the orders from the models in each iteration together with additional summary statistics are given in Appendix B.

We conclude the analysis of the order, by considering the relationship between the order of a model and its conditional log likelihood. From the summary statistics of the batch maximum algorithm from Table B.3, which had the biggest variations in terms of order and *cll*, we see that the models of iterations 1, 3, 7, 11, and 17 had the correct order of the first five predictors and their *cll* was over 700. On the other hand, the models at iterations 4, 8, 13, and 19 have more deviating orders than the true order and a *cll* of below 500. Similar trends can be observed in other algorithms as well. The two step ahead algorithm has the most accurate orders and the best *cll*. Additionally, the best *cll*'s were achieved in models with correct order on more than five predictors.

After the order, we now take a look at the out of sample mean square error (OMSE) and the in sample mean square error (IMSE). The OMSE characterizes how well the model predicts on new data sets with the same distribution as the training data. To calculate the OMSE for the specific models we use the test set D_{test} and the

formula defined in Chapter 7. The IMSE is computed on the training set D_{train} and characterizes how well the model has fitted the training data. Tables 8.5 and 8.6 give the OMSE and IMSE for the competitor models.

α	0.05	0.1	y_{true}	0.9	0.95
$D-A1$	0.0635	0.0445	0.0208	0.0528	0.0727
$D-A2$	**0.0182**	**0.0128**	**0.0052**	**0.0125**	**0.0182**
$D-Bmax$	0.0452	0.0317	0.0140	0.0349	0.0491
$D-Bavg$	0.0711	0.0501	0.0222	0.0543	0.0765
$QLin$	0.0250	0.0251	0.0254	0.0258	0.0259

Table 8.5: Out of sample mean square errors.

α	0.05	0.1	y_{true}	0.9	0.95
$D-A1$	0.1744	0.1205	0.0506	0.1287	0.1826
$D-A2$	**0.0476**	**0.0326**	**0.0132**	**0.0353**	**0.0519**
$D-Bmax$	0.1273	0.0872	0.0340	0.0929	0.1335
$D-Bavg$	0.2063	0.1414	0.0544	0.1427	0.2048
$QLin$	0.0636	0.0638	0.0646	0.0655	0.0658

Table 8.6: In sample mean square errors.

The OMSE and IMSE are evaluated at the values $\alpha = 0.05, 0.1, 0.5, 0.9, 0.95$. Assume predicted $q_{0.5}$ as a prediction of y_{true}. For $\alpha = 0.5$ the true values of the response in the test data set were used, while for the other values of α the quantiles were estimated from the test set and the known true underlying distribution. The smaller the value of either the OMSE or the IMSE indicates a better performance on the respective data set.

The values of the OMSE and IMSE from Tables 8.5 and 8.6 indicate that the two step ahead model performs the best in both, predicting the response of the test set, as well as fitting the training set. Linear regression had the second best score, while the next best scores were from the one step ahead algorithm and the batch maximum algorithm. For the D-vine copula based approaches there is again a relationship between the measures of the obtained models. Namely, the summary statistics from Table B.1 shows that for the two step ahead algorithms, the estimated models at iterations 1, 10, 14 and 19 showed the best OMSE and IMSE scores among the models from the 20 iterations. Further, they were also among the best performing cll and cll_{AIC} models, while also estimating the true order up to at least five predictors correctly. On the other hand, the worst OMSE and IMSE scores were obtained by models 2 and 15, which were also the two lowest scoring models according to cll, cll_{AIC} and order. Tables B.2, B.3 and B.4 show the same behavior for the models independently of the applied algorithm.

Finally, in order to better compare the OMSE and IMSE score of the competitor methods relative to two step ahead forward selection we also consider the relative out of and in sample mean square error of algorithm A, defined as

$$\text{RoMSE}_A = \frac{OMSE_A}{OMSE} \quad \text{RiMSE}_A = \frac{IMSE_A}{IMSE},$$

where $OMSE$ and $IMSE$ represent the OMSE and IMSE score of the two step ahead algorithm while, $OMSE_A$ and $IMSE_A$ represent the OMSE and IMSE score of algorithm A. Values greater than one imply a worse relative performance compared to the two step ahead algorithm

α	0.05	0.1	y_{true}	0.9	0.95
RoMSE_{D-A1}	3.489	3.477	3.925	4.224	3.995
RoMSE_{QLin}	1.373	1.961	4.793	2.064	1.423

α	0.05	0.1	y_{true}	0.9	0.95
RiMSE_{D-A1}	3.664	3.696	3.833	3.646	3.518
RiMSE_{QLin}	1.336	1.957	4.902	1.855	1.268

Table 8.7: RoMSE and RiMSE values for the one step ahead algorithm and linear quantile regression.

Table 8.7 shows that the two step ahead has performed on average twice better than quantile linear regression and four times better than the one step ahead algorithm, both on the test and train data sets.

We conclude the review of the results with a short note on the computational time of each algorithm. Averaged over the 20 iteration, the following computational times were recorder

	One step ahead	Two step ahead	Batch algorithms
$x.x$ minutes	5.90	32.19	19.33

Table 8.8: Average computational times given in minutes.

As already discussed in Section 6.1, the much larger computational intensity of the two step ahead algorithm is expected. The two step ahead algorithm pays a price for the better fit, being six times slower than the one step ahead approach. The batch algorithms are considerably faster than the two step ahead algorithm, but still fail to reach the time of the one step ahead algorithm by far. For the purpose of obtaining these values the algorithms were run on a commercial machine and can be drastically reduced if larger computational power is available at hand.

8.2 Case Study II: C-vine regression models

For the second case study, we sample a data set D of $n = 400$ data points from a random vector $(V, U_1, \ldots, U_{20})$ following a 21-dimensional C-vine copula distribution $\mathcal{C}_{21}$ with order

$$\mathcal{O}\left(V, U_1, U_2, U_3, \ldots U_{20}, U_{21}\right). \tag{8.2}$$

To make sure there is significant dependence between the response variable and the predictors we define the following pair copulas of the C-vine $\mathcal{C}_{21}$

Pair copula	Family	Kendall's τ
C_{V,U_1}	Clayton	0.8
$C_{V,U_2;U_1}$	Gumbel	0.8
$C_{V,U_3;U_2,U_1}$	Joe	0.6
$C_{V,U_4;U_3,U_2,U_1}$	Clayton	0.6
$C_{V,U_5;U_4,U_3,U_2,U_1}$	Frank	0.8
$C_{V,U_6;U_5,U_4,U_3,U_2,U_1}$	Joe	0.6
$C_{V,U_7;U_6,U_5,U_4,U_3,U_2,U_1}$	Frank	0.6

Table 8.9: Pair copulas defining the dependence between the response and the first seven predictors.

For the dependence in-between the first ten predictors of the order, the pair copulas were sampled from a set of families containing the Clayton, Gumbel, Joe and Frank copula with parameters corresponding to Kendall's τ between 0.4 and 0.7, except the pair copulas already defined in Table 8.9 . All other pair copulas were sampled from a set containing Gaussian copulas with $\tau < 0.3$ and the independence copula.

After obtaining the data set D we split it into a training set, D_{train} and a testing set, D_{test}, of sizes 300 and 100, respectively. In this case study, we employ the four algorithms developed in Chapter 7 and linear quantile regression from Section 3.3. All algorithms, from Chapter 7, were set to choose a model from the C-vine quantile regression class $\mathfrak{C}^{21}$ and include all possible predictors. The number of candidates for the two step ahead forward selection is set to $k = 5$ and the batch size for the maximum and average batch method is $b = 5$. Again, no *cll* penalization was used, however the bivarite copula selection criteria is set to BIC for the two step ahead models, and log likelihood for the one step ahead model. All together $r = 20$ replication of case study one were run with seeds from 1 to 20 and all the results are averaged over the 20 iterations.

We give a pairs plot of the train data set $D^{(9)}_{train}$ in Figure 8.2.

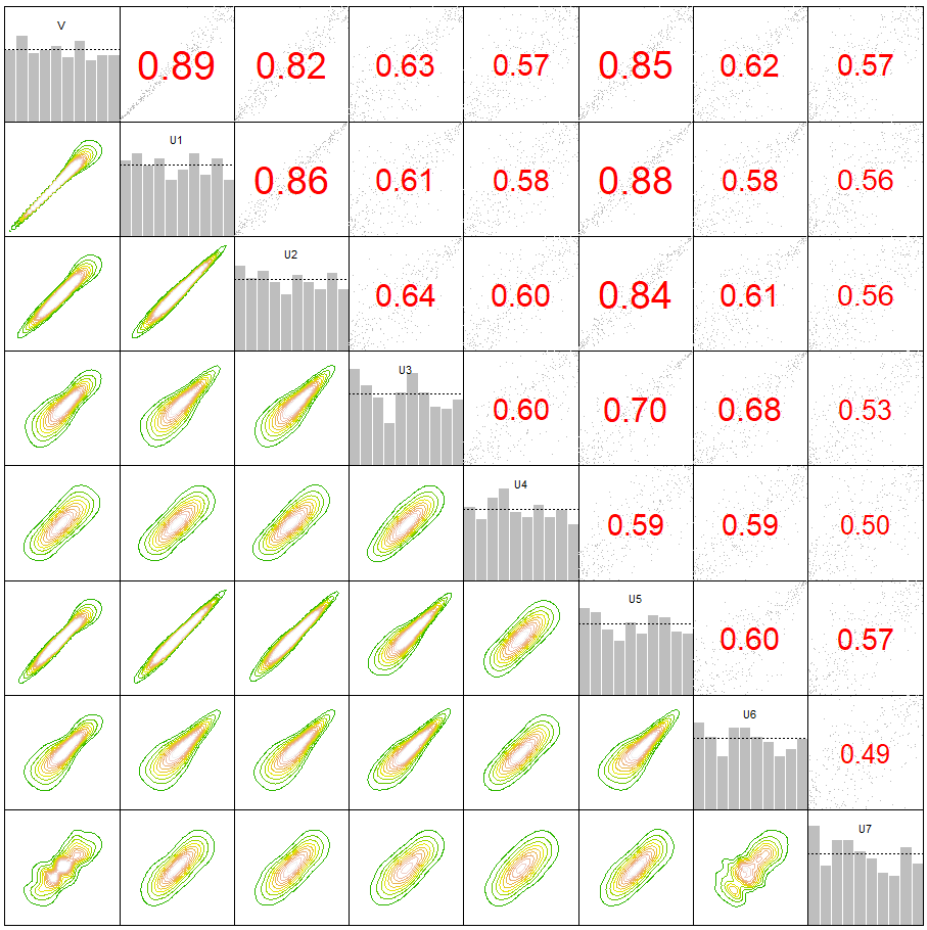

Figure 8.2: Pairs copula plot between the response and the predictors $U_1, \ldots, U_7$ of data set $D^{(9)}_{train}$, with contour plots of the pair copulas, Kendall's tau values and u-plots.

As in case study one, we first look at the *cll* and AIC penalized *cll* values of the estimated models. Tabels 8.10 and 8.11 give a summary of the *cll* and cll_{AIC} values averaged over the 20 iterations for each of the algorithms.

Algorithm	$C-A1$	$C-A2$	$C-Bmax$	$C-Bavg$
Max *cll*	**1731.62**	1073.44	1085.65	1095.93
Min *cll*	**1211.98**	959.02	890.89	990.18
Mean *cll*	**1338.40**	1020.93	1003.81	1034.66
Standard deviation of *cll*	118.96	26.32	48.66	28.11

Table 8.10: *cll* statistics of Algorithms 1-4 from Case study II.

Algorithm	$C-A1$	$C-A2$	$C-Bmax$	$C-Bavg$
Max cll_{AIC}	**-1795.95**	-1743.44	-1580.02	-1809.76
Min cll_{AIC}	**-2821.24**	-1959.87	-1991.05	-1998.84
Mean cll_{AIC}	**-2026.30**	-1861.22	-1822.28	-1893.09
Standard deviation of cll_{AIC}	237.31	49.92	100.82	51.70

Table 8.11: cll_{AIC} statistics of Algorithms 1-4 Case study II.

For the *cll* higher values indicate a better fit, while for the cll_{AIC} lower values indicate a better fit. Both goodness of fit statistics indicate that the best model comes from the one step ahead algorithm. From Tables 8.10 and 8.11 we see that by far the best values were achieved by the one step ahead algorithm, while the other three algorithms had approximately the same values in both statistics.

Next, as in case study one, we look at the order of the estimated models. We check how consistent the orders of the models are and how close the estimated orders came to the true order given in equation (8.2). Again, we look only at the order of the first seven included predictors, since we expect this seven predictors to be again the most influential. From the definitions of the pair copulas between the response and the first seven predictors of the true order, we see that the predictors U_1, U_2, U_5 have the strongest dependence with the response. The next predictors in line, according to dependence, are U_3, U_4, U_6, U_7, while the other predictors have comparably small dependence with the response.

When it comes to the true order, in this case study, all four algorithms could not estimate the true order after the second or third predictor. All algorithms constantly include U_1 as the first predictor of the order. The two step ahead algorithms included U_5 as the second predictor most of the time, while the one step ahead algorithm has managed to fit the correct order up to the third predictor in about 50% of the iterations.

Algorithm	$C-A1$	$C-A2$	$C-Bmax$	$C-Bavg$
Different orders up to 3	2	5	12	2
Different orders up to 4	2	8	17	4
Different orders up to 5	6	13	18	10

Table 8.12: Order statistics of the fits obtained from Algorithms 1-4 from Case study II.

Table 8.12 shows the number of different orders up to 3, 4 and 5 predictors for the 20 models obtained through the 20 iterations of one algorithm. It is apparent that the most consistent orders are obtained by the one step ahead algorithm. Up to 5 predictors, only 6 different orders were estimated in the 20 iterations. The next best two models were the two step ahead algorithm and the batch average, while the batch maximum, as expected due to the randomness, did by far the worst.

We conclude the discussion about the order by looking if we can find the same relationship between the correct order and the *cll*, as showed in case study one. Since the two step ahead based algorithms showed almost no standard deviation in the *cll* we look only at the one step ahead algorithm. From Table B.7 we see that the best *cll* performing models, 13, 14, 16 and 18, had also an order very close to the true order. However, due to the inconsistent order estimates and the small variation of the *cll* we can neither confirm nor deny any correlation.

After the order, we now take a look at the out of sample mean square error (OMSE) and the in sample mean square error (IMSE). Again, as in case study one, the OMSE and IMSE are evaluated at the values $\alpha = 0.05, 0.1, 0.5, 0.9, 0.95$. For $\alpha = 0.5$ the true values of the response in the test data set were used, while for the other values of α the quantiles were estimated from the test set and the known true underlying distribution. The smaller the value of either the OMSE or the IMSE indicates a better performance on the respective data set. Tables 8.13 and 8.14 give the OMSE and IMSE for the competitor models.

α	0.05	0.1	y_{true}	0.9	0.95
$C-A1$	0.0496	0.0483	0.0464	0.0468	0.0471
$C-A2$	**0.0326**	**0.0307**	**0.0280**	**0.0341**	**0.0369**
$C-Bmax$	0.0474	0.0444	0.0358	0.0309	0.0307
$C-Bavg$	**0.0302**	**0.0287**	**0.0262**	**0.0284**	**0.0294**
$QLin$	0.0371	0.0371	0.0371	0.0371	0.0371

Table 8.13: Out of sample mean square errors.

α	0.05	0.1	y_{true}	0.9	0.95
$C-A1$	0.1490	0.1450	0.1374	0.1387	0.1395
$C-A2$	**0.0982**	**0.0930**	**0.0867**	**0.1024**	**0.1101**
$C-Bmax$	0.1514	0.1403	0.1123	0.0959	0.0957
$C-Bavg$	**0.0953**	**0.0904**	**0.0817**	**0.0867**	**0.0899**
$QLin$	0.1166	0.1166	0.1166	0.1166	0.1166

Table 8.14: In sample mean square errors.

The values of the OMSE and IMSE from Tables 8.13 and 8.14 indicate that the two step ahead algorithm and the batch average algorithm performed the best in predicting the response of the test set, as well as fitting the training set. The second best score was obtained by the linear quantile regression algorithm, while the batch max and one step ahead algorithms were the worst according to both OMSE and IMSE scores.

The OMSE and IMSE are contradicting the *cll* statistics on which model performed the best. According to both OMSE and IMSE, the two step ahead and batch average algorithms have performed much better than the one step ahead algorithm, despite having considerably smaller conditional log likelihoods. This indicates that the one step ahead algorithm may have over-fitted the models despite the cll_{AIC} from Table 8.11 do not indicate over-fitting. To further investigate whether the one step ahead algorithm over-fitted the data we take a look at the BIC penalized conditional log likelihood, cll_{BIC}, of the models.

Algorithm	$C-A1$	$C-A2$	$C-Bmax$	$C-Bavg$
Max cll_{BIC}	-568.29	-1420.12	-1206.39	-1493.85
Min cll_{BIC}	-1632.32	-1613.54	-1657.25	-1659.39
Mean cll_{BIC}	-821.64	-1526.71	-1479.07	-1566.72
Standard deviation of cll_{BIC}	237.86	48.98	110.15	48.77

Table 8.15: cll_{BIC} statistics of Algorithms 1-5 for Case study II.

With the cll_{BIC} the same rule applies as with the cll_{AIC}, smaller values of the cll_{BIC} indicate a better fit. From the cll_{BIC} of Table 8.15 we see now a completely different trend than with the cll_{AIC}. According to the BIC penalized log likelihood the two step ahead algorithms performed significantly better than the one step ahead algorithm. The best two algorithms according to the cll_{BIC} were, as also indicated by the OMSE and IMSE, the batch average algorithm and the two step ahead algorithm. On the other hand, the cll_{BIC} indicates that the one step ahead algorithm has over-fitted the models significantly. The one step ahead algorithm had an approximately two times smaller cll_{BIC} than the other algorithms, while it had by far the best *cll*. The hypothesis that the one step ahead algorithm is over-fitting the data in this case study is supported by the relatively bad OMSE and IMSE scores.

The over-fitting indicated for the one step ahead algorithm in comparison to the two step ahead algorithms is partially coming from the different pair copula selection criteria. The one step ahead was set to fit the pair copulas based on the log likelihood, while the two step ahead algorithms were fitting the pair copulas using the BIC criteria. Note that here only the pair copula selection was penalized by the BIC and not the *cll*.

Again we conclude the review of the results with a short note on the computational time of each algorithm. Averaged over the 20 iteration, the following computational times were recorded

	One step ahead	Two step ahead	Batch algorithms
$x.x$ minutes	5.55	30.64	18.46

Table 8.16: Average computational times given in minutes.

We see almost the same computational times as in Table 8.16 for case study one, which was also expected since there is not much difference in estimating C- and D-vines.

8.3 Summary of results

The main focus of our simulation study is examining the performance of the newly introduced two step ahead algorithm compared to other methods. In the simulation study from Kraus and Czado (2017), they have shown that the one step ahead forward selection of D-vine quantile regression is usually more accurate than other benchmark methods. Therefore, we use the one step ahead algorithms for C- and D-vine models as the primary benchmark. As an additional reference we consider the linear quantile regression defined in Section 3.3.

In case study one, the two step ahead algorithm shows a consistently better performance than both, the one step ahead algorithm and linear quantile regression, with regard to all goodness of fit measures. Even more, the one step ahead algorithm performed worse than linear quantile regression based on the OMSE and IMSE measures. Kraus and Czado (2017) note that on specific data sets it can happen that the one step ahead algorithm does not over perform linear quantile regression due to the greedy selection procedure. In case study one, we encountered a similar situation where the one step ahead algorithm failed to match the true order of predictors, specifically because of the greedy estimation approach. The predictors U_2 and U_3 were chosen over U_1 as the first predictor based on the one step ahead extension, despite the fact that models having U_1 as the first predictor have a significantly better overall fit. The two step ahead algorithm avoids this problem by taking into account the two step ahead extension and therefore, it recognizes the better overall fits.

In case study two, the one step ahead algorithm showed a better *cll* than the two step ahead algorithm. However, it failed to reproduce the same result when taking into account the cll_{BIC} and the OMSE. This is due to the fact that the one step ahead algorithm over-fitted the data and therefore, produced a better *cll* score, while it failed to estimate the response on the testing set as good as the other methods. The over-fitting did not occur within the two step ahead algorithm due to using the BIC penalization for pair copula estimations. Taking into account the over-fitting of the one step ahead algorithm, again the two step ahead algorithm performs significantly better than both benchmark algorithms. This is supported by the vastly better cll_{BIC}, IMSE and OMSE scores.

Considering the orders of the estimated models, both the one step ahead algorithm and the two step ahead algorithm are very consistent in including the "best" predictors very early in the order. Additionally, the small number of different orders, when considering only the first few covariates which have a significant influence of the fit, shows that the algorithms tend to give approximately the same output models for data following the same distribution.

The batch algorithms showed significant inconsistencies between the output models. In both case studies they managed to find the best fits on several occasions, but also some of the worst fits were given by the batch algorithms. However, such results are expected due to the small batch size and relatively small number of predictors.

Finally, we conclude that the two step algorithm performs significantly better then both benchmark algorithms, for both C- and D-vines.

Part IV

Conclusion

In this master thesis, we are concerned with the development of new models for quantile regression. Given the shortfalls and the strict assumptions of linear quantile regression, we are introducing new methods for vine copulas based quantile regression.

Building upon the ideas in Kraus and Czado (2017), we propose an algorithm which allows for more flexibility and which in particular is less greedy, given the intention to obtain a globally optimal D-vine fit. The already existing algorithm builds the D-vine step by step, starting with an empty vine consisting of only the response variable on the u-scale, V, and in each step is adding one of the predictors to the tree so that the model fit of the one-step-ahead vine is improved the most. In our newly proposed algorithm we also sequentially build the vine, but based on the model fit in the next two trees, which is the main idea of the less greedy approach. To make the vine based quantile regression even more flexible in estimating dependence structures, we extend both approaches to include the class of C-vine copulas. To unify the approaches for both C- and D-vines, new theoretical concepts of C- and D-vine quantile regression classes are introduced. Further, to make the derivation of conditional distribution functions in the set up of quantile regression tractable, we defined the concept of orders on C- and D-vine quantile regression classes. Additionally, detailed illustrations of both the one step ahead and the two step ahead algorithms for C- and D-vine copulas are provided.

The two step ahead algorithm being less greedy comes with the price of a higher computational complexity. Section 6.1 provides an extensive discussion on this topic and comparisons of computational times for the two vine copula based approaches are given. Furthermore, to adapt the two step ahead algorithm for large data sets two new stochastic versions of the algorithm are introduced.

Finally, all new methods are put to a test in a simulation study. The new methods are compared to the already existing one step ahead D-vine algorithm and linear quantile regression. The simulation study has shown a significant improvement when using the two step ahead algorithm compared to the one step ahead algorithm. Kraus and Czado (2017) note in their simulation study that the one step ahead algorithm is usually more accurate then other benchmark methods. However, they also indicate the possibility of the one step ahead algorithm to fail in certain conditions due to the greedy selection procedure. Case study one shows that the two step ahead algorithm can improve the fit significantly also in such conditions.

In addition to developing the theoretical background to C- and D-vine based quantile regression, we implemented all proposed methods in the statistical software `R`.

Bibliography

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics 44*(2), 182–198.

Bedford, T. and R. Cooke (2001). Monte carlo simulation of vine dependent random variables for applications in uncertainty analysis. *ESREL 2003*.

Bernard, C. and C. Czado (2015). Conditional quantiles and tail dependence. *Journal of Multivariate Analysis 138*, 104–126.

Cheng, R. (2017). *Non-standard parametric statistical inference*. Oxford University Press.

Czado, C. (2019). *Analyzing Dependent Data with Vine Copulas*. Springer International Publishing.

Embrechts, P., F. Lindskog, and A. McNeil (2003). Modelling dependence with copulas and applications to risk management. handbook of heavy tailed distributions in finance, edited by st rachev.

Hürlimann, W. (2003). Hutchinson-lai's conjecture for bivariate extreme value copulas. *Statistics & probability letters 61*(2), 191–198.

Joe, H. (1996). Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters. *Lecture Notes-Monograph Series*, 120–141.

Koenker, R. (2005). *Quantile Regression*. Econometric Society Monographs. Cambridge University Press.

Koenker, R. and G. Bassett Jr (1978). Regression quantiles. *Econometrica: journal of the Econometric Society*, 33–50.

Kraus, D. and C. Czado (2017). D-vine copula based quantile regression. *Computational Statistics and Data Analysis 110*, 1–18.

Mosteller, F. and J. W. Tukey (1977). Data analysis and regression: a second course in statistics. *Addison-Wesley Series in Behavioral Science: Quantitative Methods*.

Nelsen, R. B. (2007). *An introduction to copulas*. Springer Science & Business Media.

Noh, H., A. E. Ghouch, and T. Bouezmarni (2013). Copula-based regression estimation and inference. *Journal of the American Statistical Association 108*(502), 676–688.

Shao, J. (2003). Mathematical statistics.

Sheather, S. and M. Jones (1991, 01). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B. Methodological 53*, 683–690.

Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de L'Université de Paris 8*, 229–23.

Stoeber, J. and C. Czado (2017). Pair copula constructions. in simulating copulas: Stochastic models, sampling algorithms, and applications.

Vanderbei, R. J. et al. (2015). *Linear programming*. Springer.

Yule, G. U. and M. G. Kendall (1950). An introduction to the theory of statistics. *London: Charles Griffin & Co., Ltd 113*, 578–580.

Part V

Appendix

Appendix A

Illustrations on Gaussian data

A.1 Illustration of Algorithm 2 on Gaussian data

In this section, we give a step by step illustration of the algorithm introduced in Section 5.1 on a simulated 5-dimensional data set. Further, detailed estimates of pair copulas and their corresponding log likelihoods are provided, as well as the intermediate optimal models at each step.

Input: We consider a five dimensional data set $\left(y^{(i)}, x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}\right)^T$, $i = 1, \ldots, 500$, sampled from $(Y, X_1, X_2, X_3, X_4) \sim \mathcal{N}_5(0, \Sigma)$ with

$$\Sigma = \begin{pmatrix} 1 & 0.75 & 0.8 & 0.2 & 0 \\ 0.75 & 1 & 0.4 & 0.3 & 0 \\ 0.8 & 0.4 & 1 & 0.4 & 0 \\ 0.2 & 0.3 & 0.4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{A.1}$$

Data preprocessing:

We compute the margins $\hat{F}_Y$ and $\hat{F}_{X_j}$ $j = 1, \ldots, 4$ using the kernel smoothing estimator from Section 4.2. For this we use the function `kde1d` implemented in the package `kde1d`.

With the margins estimated the data is transformed to pseudo copula data

$$\left(\hat{v}^{(i)}, \hat{u}_1^{(i)}, \hat{u}_2^{(i)}, \hat{u}_3^{(i)}, \hat{u}_4^{(i)}\right)^T.$$

Initialization: To model the given data we decide to use a vine copula model from the C-vine class $\mathfrak{C}^p$ with $p \leq 4$.

Step 1:

First, we estimate the pair copulas C_{VU_j} for $j = 1, \ldots, 4$. For this we use the function `bicop` from the package `rvinecopulib`, using the default selection criteria for bivariate copula estimation BIC.

Pair copulas	$\hat{C}_{VU_1}$	$\hat{C}_{VU_2}$	$\hat{C}_{VU_3}$	$\hat{C}_{VU_4}$
Family	Gauss	Gauss	Gauss	Indep.
Parameter	0.74	0.8	0.22	0

Table A.1: Estimated bivariate copulas with their parameters.

Then, we can compute the conditional log likelihood for the estimated pair copulas, which in the first step is just the log likelihood of the corresponding pair copula.

Pair copulas	$\hat{C}_{VU_1}$	$\hat{C}_{VU_2}$	$\hat{C}_{VU_3}$	$\hat{C}_{VU_4}$
Conditional log likelihood	201.1717	**259.7633**	13.17153	0

Table A.2: Conditional log likelihood of the estimated pair copulas.

Since the copula $\hat{C}_{VU_2}$ has the greatest conditional log likelihood we choose U_2 as the first predictor to be included in the order. Thus, our estimated C-vine copula model after the first step is $\mathcal{C}^* \in \mathfrak{C}^1$ with order

$$\mathcal{O}\left(\mathcal{C}^*\right) = (V, U_2)\,.$$

Before we continue with the second step, we estimate the pseudo copula data needed for fitting pair copulas in the second tree.

$$\hat{u}^{(i)}_{V|U_2} = h_{V|U_2}\left(\hat{v}^{(i)}|\hat{u}^{(i)}_2\right).$$

This is done using the function `hbicop` from the package `rvinecopulib`.

Step 2:

Now, for $j = 1, 3, 4$, in order to estimate the conditional log likelihood of the copulas $\mathcal{C}_j \in \mathfrak{C}^2$ with order $\mathcal{O}\left(\mathcal{C}_j\right) = (V, U_2, U_j)$ we first have to fit the pair copulas $\hat{C}_{U_2U_j}$ and $\hat{C}_{VU_j;U_2}$.

The pair copulas $\hat{C}_{U_2U_j}$ have to be estimated in order to obtain the pseudo copula data

$$\hat{u}^{(i)}_{U_j|U_2} = h_{U_j|U_2}\left(\hat{u}^{(i)}_j|\hat{u}^{(i)}_2\right).$$

Then, we can estimate $\hat{C}_{VU_j;U_2}$ using the pseudo copula data $\hat{u}^{(i)}_{V|U_2}$ and $\hat{u}^{(i)}_{U_j|U_2}$.

Now, we can calculate the conditional log likelihood of the C-vine models $\mathcal{C}_j$ which are given as

$$cll\left(\mathcal{C}_j, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_j)\right) = cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2)\right) + l\left(\hat{C}_{VU_j;U_2}\right)$$

Pair copulas	$\hat{C}_{U_2U_1}$	$\hat{C}_{VU_1;U_2}$	$\hat{C}_{U_2U_3}$	$\hat{C}_{VU_3;U_2}$	$\hat{C}_{U_2U_4}$	$\hat{C}_{VU_4;U_2}$
Family	Gauss	Gauss	Gauss	Clayton(90)	Indep.	Indep.
Parameter	0.39	0.78	0.4	0.21	0	0
Log likelihood	-	235.7242		8.651463	-	0

Table A.3: Estimated pair copulas in Step 2.

Candidate models	$\mathcal{C}_1$	$\mathcal{C}_3$	$\mathcal{C}_4$
Conditional log likelihood	**495.4875**	268.4148	235.7242

Table A.4: Conditional log likelihood of candidate C-vine models.

Further, we apply the conditional likelihood ratio test 4.12 to the C-vine model with the greatest conditional log likelihood. That is, we check

$$cll\left(\mathcal{C}_j, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_j)\right) - cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2)\right) > \chi^2_{0.95,1}.$$

Since the model $\mathcal{C}_1$ has the greatest conditional log likelihood, it follows that

$$cll\left(\mathcal{C}_1, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_1)\right) - cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2)\right) = 235.7242 > 3.841459 = \chi^2_{0.95,1}.$$

Thus, it is justified to add U_1 to the model.

We update our optimal fit $\mathcal{C}^*$ to the model $\mathcal{C}_1$. This means that we add U_1 as the next predictor, thus our estimated C-vine copula model after the second step is $\mathcal{C}^* \in \mathfrak{C}^2$ with order

$$\mathcal{O}\left(\mathcal{C}^*\right) = (V, U_2, U_1).$$

Again, we calculate the pseudo copula data needed for the next step,

$$\hat{u}^{(i)}_{V|U_2,U_1} = h_{V|U_1;U_2}\left(\hat{u}^{(i)}_{V|U_2}|\hat{u}^{(i)}_{U_1|U_2}\right).$$

Step 3:

For $j = 3, 4$, in order to estimate the condition log likelihood of the C-vine model $\mathcal{C}_j \in \mathfrak{C}^3$ with order $\mathcal{O}\left(\mathcal{C}_j\right) = (V, U_2, U_1, U_j)$ we first have to fit the pair copulas $\hat{C}_{U_jU_1;U_2}$ and $\hat{C}_{VU_j;U_2,U_1}$.

Similarly as in Step 2, the copulas $\hat{C}_{U_jU_1;U_2}$ are fitted only to obtain the pseudo copula data

$$\hat{u}^{(i)}_{U_j|U_2,U_1} = h_{U_j|U_1;U_2}\left(\hat{u}^{(i)}_{U_j|U_2}|\hat{u}^{(i)}_{U_1|U_2}\right),$$

so that we can estimate $\hat{C}_{VU_j;U_2,U_1}$ with the pseudo data $\hat{u}^{(i)}_{V|U_2,U_1}$ and $\hat{u}^{(i)}_{U_j|U_2,U_1}$.

We calculate the conditional log likelihood for the C-vine models $\mathcal{C}_j$

Pair copulas	$\hat{C}_{U_3U_1;U_2}$	$\hat{C}_{VU_3;U_2,U_1}$	$\hat{C}_{U_4U_1;U_2}$	$\hat{C}_{VU_4;U_2,U_1}$
Family	Gumbel	Gauss	Indep.	Indep.
Parameter	1.15	0.53	0	0
Log likelihood	-	82.98469	-	0

Table A.5: Estimated pair copulas in Step 3.

Candidate models	$\mathcal{C}_3$	$\mathcal{C}_4$
Conditional log likelihood	**578.4722**	495.4875

Table A.6: Conditional log likelihood of candidate C-vine models.

Obviously, $\mathcal{C}_3$ has the greatest conditional log likelihood. Again, we have to apply the conditional likelihood ratio test which yields

$$cll\left(\mathcal{C}_3, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_1, \hat{\mathbf{u}}_3)\right) - cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_2, \hat{\mathbf{u}}_1)\right) = 82.98469 > 3.841459 = \chi^2_{0.95,1}.$$

Thus, it is justified to add U_3 to the model.

We update our optimal fit $\mathcal{C}^*$ to the C-vine model $\mathcal{C}_3$. Thus, our estimated C-vine copula model after the third step is $\mathcal{C}^* \in \mathfrak{C}^3$ with order

$$\mathcal{O}\left(\mathcal{C}^*\right) = (V, U_2, U_1, U_3)\,.$$

Calculate the pseudo copula data needed for the next step,

$$\hat{u}^{(i)}_{V|U_2,U_1,U_3} = h_{V|U_3;U_2,U_1}\left(\hat{u}^{(i)}_{V|U_2,U_1} | \hat{u}^{(i)}_{U_3|U_2,U_1}\right).$$

Step 4:

At this point only one predictor is remaining. We have to decide whether to add U_4 to the model or not.

The copula $\hat{C}_{VU_4;U_2,U_1,U_3}$ is estimated to be the Independence copula.

Thus, adding U_4 to the model will not change the conditional log likelihood and therefore, we do not include U_4 in the model.

Output: The optimal C-vine copula model $\mathcal{C}^* \in \mathfrak{C}^3$, which was last updated at Step 3, with order

$$\mathcal{O}\left(\mathcal{C}^*\right) = (V, U_2, U_1, U_3)\,.$$

The conditional log likelihood of $\mathcal{C}^*$ is 578.4722.

A.2 Illustration of Algorithm 2 on Gaussian data

In this section, we give a step by step illustration of the algorithm introduced in Section 5.2 on a simulated 5-dimensional data set. Further, detailed estimates of pair copulas and their corresponding log likelihoods are provided, as well as the intermediate optimal models at each step.

Input: We consider the same five dimensional data set $\left(y^{(i)}, x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}\right)^T$, $i = 1, \ldots, 500$, sampled from $(Y, X_1, X_2, X_3, X_4) \sim \mathcal{N}_5(0, \Sigma)$ with Σ defined as in equation (A.1).

Data preprocessing: We compute the margins $\hat{F}_Y$ and $\hat{F}_{X_j}$ $j = 1, \ldots, 4$ using the kernel smoothing estimator from Section 4.2. For this we use the function `kde1d` implemented in the package `kde1d`. Then the data is transformed to pseudo copula data

$$\left(\hat{v}^{(i)}, \hat{u}_1^{(i)}, \hat{u}_2^{(i)}, \hat{u}_3^{(i)}, \hat{u}_4^{(i)}\right)^T.$$

Initialization:

To model the given data we decide to use a vine copula model from the C-vine class.

The number of candidates k also has to be predefined. We choose $k = 2$ for this particular example.

Step 1:

To obtain the candidates in the first step we estimate the Kendall's tau values between the response and all possible predictors, denoted by $\hat{\tau}_{VU_j}$ for $j = 1, \ldots 4$ based on

$$\left\{v^{(i)}, u_j^{(i)} | i = 1, \ldots, 500\right\}.$$

For this we use the function `cor` from the package `stats`.

$\hat{\tau}_{VU_1}$	$\hat{\tau}_{VU_2}$	$\hat{\tau}_{VU_3}$	$\hat{\tau}_{VU_4}$
0.5308377	**0.5919038**	0.1421403	0.02154709

Table A.7: Estimated Kendall's tau values.

$k = 2$ predictors with the greatest estimated absolute Kendall's tau values are chosen as candidate predictors. In this case, that are U_2 and U_1, as can be seen from Table A.7, and the set of candidate indices is then $K = \{2, 1\}$.

For $c \in K$ the conditional log likelihoods of interest are those of the two step ahead copulas $\mathcal{C}_{c,j} \in \mathfrak{C}^2$ for $j \in \{1, \ldots, 4\} \backslash \{c\}$ with order $\mathcal{O}(\mathcal{C}_{c,j}) = (V, U_c, U_j)$. Those *cll*'s are obtained as

$$cll\left(\mathcal{C}_{c,j}, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j)\right) = l\left(\hat{C}_{VU_c}, (\hat{\mathbf{v}}, \hat{\mathbf{u}}_j)\right) + l\left(\hat{C}_{VU_j;U_c}, \left(\hat{\mathbf{u}}_{V|U_c}, \hat{\mathbf{u}}_{U_j|U_c}\right)\right), \quad \text{(A.2)}$$

where $\hat{\mathbf{u}}_{V|U_c}$ and $\hat{\mathbf{u}}_{U_j|U_c}$ are the pseudo copula data defined as

$$\hat{\mathbf{u}}_{V|U_c} := h_{V|U_c}\left(\hat{\mathbf{v}}|\hat{\mathbf{u}}_c\right) \quad \text{and} \quad \hat{\mathbf{u}}_{U_j|U_c} := h_{U_j|U_c}\left(\hat{\mathbf{u}}_j|\hat{\mathbf{u}}_c\right).$$

In order to obtain the desired conditional log likelihoods, equation (A.2) implies that we have to estimate the pair copulas C_{VU_c} and $C_{VU_j;U_c}$, because their log likehoods contribute to the *cll*. However, we also have to estimate the pair copulas $C_{U_cU_j}$, since they are needed to obtain the pseudo copula data $\hat{\mathbf{u}}_{U_j|U_c}$ which is used in the estimation of $C_{VU_j;U_c}$.

Pair copulas	$\hat{C}_{VU_2}$	$\hat{C}_{U_2U_1}$	$\hat{C}_{VU_1;U_2}$	$\hat{C}_{U_2U_3}$	$\hat{C}_{VU_3;U_2}$	$\hat{C}_{U_2U_4}$	$\hat{C}_{VU_4;U_2}$
Family	Gauss	Gauss	Gauss	Gauss	Clayton(90)	Indep.	Indep.
Parameter	0.8	0.39	0.78	0.4	0.21	0	0
Loglik	259.7633	-	235.7242	-	8.651463	-	0

Table A.8: Estimated pair copulas for $c = 2$.

Pair copulas	$\hat{C}_{VU_1}$	$\hat{C}_{U_1U_2}$	$\hat{C}_{VU_2;U_1}$	$\hat{C}_{U_1U_3}$	$\hat{C}_{VU_3;U_1}$	$\hat{C}_{U_1U_4}$	$\hat{C}_{VU_4;U_1}$
Family	Gauss	Gauss	Gauss	Gumbel	Indep.	Indep.	Indep.
Parameter	0.74	0.39	0.83	1.24	0	0	0
Loglik	201.1717	-	294.317	-	0	-	0

Table A.9: Estimated pair copulas for $c = 1$.

The log likehoods of the copulas $C_{U_cU_j}$ were omitted since they are not contributing to the *cll* of the models.

Following equation (A.2) we calculate the conditional log likelihoods of the two step ahead vine models

Candidate predictor	U_2			U_1		
Candidate models	$\mathcal{C}_{2,1}$	$\mathcal{C}_{2,3}$	$\mathcal{C}_{2,4}$	$\mathcal{C}_{1,2}$	$\mathcal{C}_{1,3}$	$\mathcal{C}_{1,4}$
cll	495.4875	268.4148	259.7633	**495.4887**	201.1717	201.1717

Table A.10: Conditional log likelihood of the candidate models.

Since the greatest *cll* is associated with the candidate model $\mathcal{C}_{1,2}$, the corresponding candidate predictor, U_1, is added first to the C-vine model. Consequently, the current optimal fit $\mathcal{C}^* \in \mathfrak{C}^1$ is the C-vine model with order

$$\mathcal{O}\left(\mathcal{C}^*\right) = (V, U_1).$$

The *cll* of $\mathcal{C}^*$ is 201.1717.

Step 2:

In the second step, the candidates are obtained using the empirical partial correlations. Namely, we choose the $k = 2$ candidates based on $|\hat{\rho}_{V,U_j;U_1}|$ for $j = 2, 3, 4$.

$\hat{\rho}_{V,U_2;U_1}$	$\hat{\rho}_{V,U_3;U_1}$	$\hat{\rho}_{V,U_4;U_1}$
0.8118951	**-0.02416175**	-0.01078996

Table A.11: Estimated partial correlations .

The partial correlations are calculated using the function `pcor` from the package `ppcor`.

Since U_2 and U_3 have the greatest absolute partial correlations with V, they are the candidate predictors at step two and the set of candidate indices is consequently $K = \{2, 3\}$.

The conditional log likelihoods of interest now are those of the two step ahead models $\mathcal{C}_{c,j}$ for $c \in K$ and $j \in \{2, 3, 4\} \setminus \{c\}$, with order $\mathcal{O}(\mathcal{C}_{c,j}) = (V, U_1, U_c, U_j)$. The *cll*'s of interest are then calculated as

$$\begin{aligned} cll\left(\mathcal{C}_{c,j}, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j)\right) = cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1)\right) + l\left(\hat{C}_{VU_c;U_1}, \left(\hat{\mathbf{u}}_{V|U_1}, \hat{\mathbf{u}}_{U_c|U_1}\right)\right) \\ + l\left(\hat{C}_{VU_j;U_1,U_c}, \left(\hat{\mathbf{u}}_{V|U_c;U_1}, \hat{\mathbf{u}}_{U_j|U_c;U_1}\right)\right), \end{aligned} \tag{A.3}$$

where the pseudo copula data is obtained as

$$\hat{\mathbf{u}}_{V|U_1} := h_{V|U_1}\left(\hat{\mathbf{v}}|\hat{\mathbf{u}}_1\right), \quad \hat{\mathbf{u}}_{U_c|U_1} := h_{U_c|U_1}\left(\hat{\mathbf{u}}_c|\hat{\mathbf{u}}_1\right), \quad \hat{\mathbf{u}}_{U_j|U_1} := h_{U_j|U_1}\left(\hat{\mathbf{u}}_j|\hat{\mathbf{u}}_1\right), \tag{A.4}$$

and

$$\hat{\mathbf{u}}_{V|U_c;U_1} := h_{V|U_c;U_1}\left(\hat{\mathbf{u}}_{V|U_1}|\hat{\mathbf{u}}_{U_c|U_1}\right), \quad \hat{\mathbf{u}}_{U_j|U_c;U_1} := h_{U_j|U_c;U_1}\left(\hat{\mathbf{u}}_{U_j|U_1}|\hat{\mathbf{u}}_{U_c|U_1}\right). \tag{A.5}$$

In order to be able to estimate the pseudo copula data from equation (A.4), (A.5) and consequently the *cll*, the pair copulas associated with the $h-$ functions need to be estimated. Namely, the associated pair copulas with the h-functions of equation (A.4) are C_{VU_1}, $C_{U_cU_1}$ and $C_{U_jU_1}$, respectively. The associated pair copulas with the h-functions of equation (A.5) are $C_{VU_c;U_1}$ and $C_{U_cU_j;U_1}$, respectively.

Additionally, also the pair copulas whose log likelihood contributes to the *cll* need to be estimated. Those pair copulas are $C_{VU_j;U_1,U_c}$ and $C_{VU_c;U_1}$, as can be seen from equation (A.3).

Pair copulas	$\hat{C}_{VU_2;U_1}$	$\hat{C}_{U_2U_3;U_1}$	$\hat{C}_{VU_3;U_1,U_2}$	$\hat{C}_{U_2U_4;U_1}$	$\hat{C}_{VU_4;U_1,U_2}$
Family	Gauss	Gauss	Gauss	Indep.	Indep.
Parameter	0.83	0.3	-0.53	0	0
Loglik		-	82.83771	-	0

Table A.12: Estimated pair copulas for $c = 2$.

Pair copulas	$\hat{C}_{VU_3;U_1}$	$\hat{C}_{U_3U_2;U_1}$	$\hat{C}_{VU_2;U_1,U_3}$	$\hat{C}_{U_3U_4;U_1}$	$\hat{C}_{VU_4;U_1,U_3}$
Family	Indep.	Gauss	Gauss	Indep.	Indep.
Parameter	0	0.3	0.88	0	0
Loglik	0	-	376.1752	-	0

Table A.13: Estimated pair copulas for $c = 3$.

The pair copulas C_{VU_1}, $C_{U_cU_1}$ and $C_{U_jU_1}$ are already estimated in step 1 and can be found in the Table A.9. The pair copulas which need to be additionally estimated are given in the following tables.

The conditional log likelihood of the two step ahead models are calculated and given as:

Candidate predictor	U_2		U_3	
Candidate models	$\mathcal{C}_{2,3}$	$\mathcal{C}_{2,4}$	$\mathcal{C}_{3,2}$	$\mathcal{C}_{3,4}$
cll	**578.3264**	495.4887	577.3469	201.1717

Table A.14: Conditional log likelihood of the candidate models.

The greatest *cll* is associated with the candidate model $\mathcal{C}_{2,3}$. Thus, we check whether updating the current optimal fit $\mathcal{C}^* \in \mathfrak{C}^1$ with order $\mathcal{O}(\mathcal{C}^*) = (V, U_1)$ to $\mathcal{C}_2 \in \mathfrak{C}^2$ which will have order $\mathcal{O}(\mathcal{C}_2) = (V, U_1, U_2)$ will increase the *cll* significantly. Namely, we employ the conditional log likelihood ratio test, and since

$$cll\left(\mathcal{C}_2, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2)\right) - cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1)\right) = 294.317 > 3.841459 = \chi^2_{0.95,1},$$

we reject H_0 at level 0.05. Thus, updating the optimal fit to $\mathcal{C}_2$ will improve the model fit significantly.

The current optimal fit $\mathcal{C}^*$ is updated to $\mathcal{C}^* \in \mathfrak{C}^2$ with order

$$\mathcal{O}(\mathcal{C}_2) = (V, U_1, U_2).$$

The *cll* of $\mathcal{C}^*$ is 495.4887.

Step 3:

Now, only two more possible predictors are left, and since the number of possible predictors is less or equal to k, we do not calculate the partial correlations. Instead, we take all remaining possible predictors as candidates. Thus, now $K = \{3, 4\}$.

The conditional log likelihoods of interest now are those of the two step ahead models $\mathcal{C}_{c,j}$ for $c \in K$ and $j \in \{3, 4\} \setminus \{c\}$, with order $\mathcal{O}(\mathcal{C}_{c,j}) = (V, U_1, U_2, U_c, U_j)$. The *cll*'s of interest are then calculated as

$$\begin{aligned} cll\left(\mathcal{C}_{c,j}, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_c, \hat{\mathbf{u}}_j)\right) = cll\left(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2)\right) \\ + l\left(\hat{C}_{VU_c;U_1,U_2}, \left(\hat{\mathbf{u}}_{V|U_2;U_1}, \hat{\mathbf{u}}_{U_c|U_2;U_1}\right)\right) \\ + l\left(\hat{C}_{VU_j;U_1,U_2,U_c}, \left(\hat{\mathbf{u}}_{V|U_c;U_1,U_2}, \hat{\mathbf{u}}_{U_j|U_c;U_1,U_2}\right)\right), \end{aligned}$$

where

$$\hat{\mathbf{u}}_{V|U_2;U_1} := h_{V|U_2;U_1}\left(\hat{\mathbf{u}}_{V|U_1}|\hat{\mathbf{u}}_{U_2|U_1}\right), \quad \hat{\mathbf{u}}_{U_c|U_2;U_1} := h_{U_c|U_2;U_1}\left(\hat{\mathbf{u}}_{U_c|U_1}|\hat{\mathbf{u}}_{U_2|U_1}\right)$$

and

$$\begin{aligned} \hat{\mathbf{u}}_{V|U_c;U_1,U_2} &:= h_{V|U_c;U_1,U_2}\left(\hat{\mathbf{u}}_{V|U_2;U_1}|\hat{\mathbf{u}}_{U_c|U_2;U_1}\right), \\ \hat{\mathbf{u}}_{U_j|U_c;U_1,U_2} &:= h_{U_j|U_c;U_1,U_2}\left(\hat{\mathbf{u}}_{U_j|U_2;U_1}|\hat{\mathbf{u}}_{U_c|U_2;U_1}\right). \end{aligned}$$

The pair copulas that are not estimated in the first two steps, and the pair copulas that will be contributing to the *cll*'s are given in the table below.

Candidate predictor		U_3		U_4	
Candidate models		$\mathcal{C}_{3,4}$		$\mathcal{C}_{4,3}$	
Pair copulas	$\hat{C}_{U_4U_3;U_1,U_2}$	$\hat{C}_{VU_3;U_1,U_2}$	$\hat{C}_{VU_4;U_1,U_2,U_3}$	$\hat{C}_{VU_4;U_1,U_2}$	$\hat{C}_{VU_3;U_1,U_2,U_4}$
Family	Indep	Gauss	Indep.	Indep.	Gauss
Parameter	0	-0.53	0	0	-0.53
Loglik.	-	**82.83771**	0	0	82.83771
Conditional loglik.		82.83771		82.83771	

Table A.15: Estimated pair copulas and conditional log likelihoods.

From Table A.15 we see that both candidate predictors have the same maximal *cll* among their respective candidate models. Thus, we choose the candidate with the best *cll* at the one step ahead model. Comparing $C_{V,U_3;U_1,U_2}$ and $C_{V,U_4;U_1,U_2}$, we see that the copula $C_{V,U_4;U_1,U_2}$ is estimated to be the Independence copula, and thus we choose $C_{V,U_3;U_1,U_2}$. In this case, the chosen candidate then becomes U_3.

Before we add U_3 to the model, we employ the conditional log likelihood ratio test. Namely, we check whether updating the current optimal fit $\mathcal{C}^* \in \mathfrak{C}^2$ with order $\mathcal{O}(\mathcal{C}^*) = (V, U_1, U_2)$ to $\mathcal{C}_3 \in \mathfrak{C}^3$ which will have order $\mathcal{O}(\mathcal{C}_3) = (V, U_1, U_2, U_3)$ will increase the *cll* significantly. Since

$$cll(\mathcal{C}_3, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \hat{\mathbf{u}}_3)) - cll(\mathcal{C}^*, \hat{\mathbf{v}}, (\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2)) = 82.83771 > 3.841459 = \chi^2_{0.95,1},$$

we reject H_0 at level 0.05. Thus, updating the optimal fit to $\mathcal{C}_3$ will improve the model fit significantly.

The current optimal fit $\mathcal{C}^*$ is updated to $\mathcal{C}^* \in \mathfrak{C}^3$ with order

$$\mathcal{O}(\mathcal{C}_3) = (V, U_1, U_2, U_3).$$

The *cll* of $\mathcal{C}^*$ is 578.3264.

Step 4:

At this point only one predictor is remaining. Thus, we can not calculate any two step ahead *cll*'s and the only thing left is to decide whether to add U_4 to the model or not.

The copula $\hat{C}_{VU_4;U_1,U_2,U_3}$ is estimated to be the Independence copula.

Thus, adding U_4 to the model will not change the conditional log likelihood and therefore, we do not include U_4 in the model.

Output: The optimal C-vine copula model $\mathcal{C}^* \in \mathfrak{C}^3$ which was last updated at Step 3 and has order

$$\mathcal{O}(\mathcal{C}^*) = (V, U_1, U_2, U_3).$$

The conditional log likelihood of $\mathcal{C}^*$ is 578.3264.

Comment:
The two illustrations on Gaussian data reveal that if the underlying model is close to a mutlivariate gaussian copula there is no big difference in conditional log likelihoods of the models obtained from Algorithm 1 ("Forward selection of C- and D-vine quantile regression models" Section 5.1) and Algorithm 2 ("Two step ahead forward selection of C- and D-vine quantile regression models" Section 5.2). Algorithm 1 fitted a C-vine model

$$\mathcal{C}_{AG1} \in \mathfrak{C}^3, \text{ with order } \mathcal{O}(\mathcal{C}_{AG1}) = (V, U_2, U_1, U_3) \text{ and } cll(\mathcal{C}_{AG1}) = 578.4722,$$

while Algorithm 2 fitted a model with

$$\mathcal{C}_{AG2} \in \mathfrak{C}^3, \text{ with order } \mathcal{O}(\mathcal{C}_{AG2}) = (V, U_1, U_2, U_3) \text{ and } cll(\mathcal{C}_{AG2}) = 578.3264.$$

There is almost no difference between the conditional log likelihoods of the two fits. Despite the fact that their order is different both models are almost identical in performance, since on Gaussian data the order of predictors in the model is less important and with the two step ahead approach of Algorithm 2 there is no room for much improvement. The difference in conditional log likelihoods is due to an estimation error.

Appendix B

Results from the simulation study

B.1 Case study I results

The following tables provide the order of the estimated models up to 7 predictors, the cll, cll_{AIC}, the IMSE for $\alpha = 0.5$ and the OMSE for $\alpha = 0.1, 0.5, 0.9$ for each iteration.

id	order	cll	cll_{AIC}	$imse_{0.5t}$	$omse_{0.1}$	$omse_{0.5t}$	$omse_{0.9}$
1	$U_1, U_2, U_3, U_4, U_5, U_9, U_{20}$	718.77	-772.67	0.0079	0.0060	0.0019	0.0095
2	$U_3, U_1, U_2, U_5, U_4, U_8, U_{16}$	524.88	-398.39	0.0518	0.0327	0.017	0.0367
3	$U_1, U_2, U_3, U_4, U_5, U_9, U_6$	737.96	-795.54	0.0083	0.0074	0.0032	0.0083
4	$U_1, U_2, U_3, U_{10}, U_5, U_7, U_{12}$	706.81	-751.68	0.0096	0.0103	0.0031	0.0081
5	$U_1, U_2, U_3, U_4, U_5, U_{10}, U_{20}$	696.06	-745.11	0.017	0.0117	0.0035	0.0093
6	$U_1, U_2, U_3, U_4, U_8, U_9, U_5$	700.2	-730.48	0.0083	0.0147	0.0053	0.0122
7	$U_1, U_2, U_3, U_4, U_5, U_6, U_8$	727.16	-777.92	0.0077	0.0102	0.0051	0.0114
8	$U_1, U_2, U_3, U_5, U_9, U_{15}, U_{17}$	690.13	-686.65	0.0124	0.0182	0.0059	0.0117
9	$U_1, U_2, U_3, U_4, U_7, U_9, U_5$	641.97	-617.58	0.0126	0.0195	0.0061	0.0117
10	$U_1, U_2, U_3, U_4, U_5, U_{10}, U_6$	720.19	-776.99	0.0085	0.0088	0.0023	0.0079
11	$U_1, U_2, U_3, U_4, U_9, U_{11}, U_5$	713.35	-745.3	0.0097	0.0065	0.0034	0.0125
12	$U_1, U_2, U_3, U_4, U_5, U_{11}, U_{12}$	730.47	-800.51	0.0077	0.0094	0.0035	0.0073
13	$U_1, U_2, U_3, U_4, U_5, U_9, U_6$	778.45	-894.07	0.0066	0.0068	0.0020	0.0052
14	$U_1, U_2, U_3, U_4, U_5, U_7, U_9$	752.12	-817.25	0.0084	0.0052	0.0027	0.0099
15	$U_3, U_1, U_2, U_4, U_7, U_5, U_{12}$	558.21	-432.46	0.034	0.0269	0.0143	0.0275
16	$U_1, U_2, U_3, U_4, U_5, U_{14}, U_9$	690.7	-692.68	0.0162	0.01063	0.0066	0.0164
17	$U_1, U_2, U_3, U_4, U_5, U_{16}, U_6$	721.19	-774.98	0.0081	0.0124	0.0050	0.0103
18	$U_1, U_2, U_3, U_4, U_7, U_8, U_5$	669.69	-677.14	0.0120	0.0145	0.0044	0.0122
19	$U_1, U_2, U_3, U_4, U_5, U_6, U_9$	723.83	-795.53	0.0087	0.0081	0.0037	0.0116
20	$U_1, U_2, U_3, U_4, U_5, U_{11}, U_9$	711.46	-739.73	0.0082	0.0158	0.0062	0.0097

Table B.1: Two step ahead forward selection algorithm statistics.

id	order	cll	cll_{AIC}	$imse_{0.5t}$	$omse_{0.1}$	$omse_{0.5t}$	$omse_{0.9}$
1	$U_3, U_1, U_2, U_4, U_{10}, U_5, U_9$	500.82	-331.63	0.0534	0.0406	0.0132	0.0361
2	$U_3, U_1, U_4, U_2, U_5, U_9, U_8$	504.39	-348.77	0.0418	0.0355	0.0189	0.0438
3	$U_3, U_1, U_2, U_4, U_5, U_9, U_6$	521.87	-395.75	0.0476	0.0451	0.0163	0.0638
4	$U_3, U_1, U_2, U_4, U_5, U_6, U_9$	492.55	-335.09	0.0513	0.0366	0.0172	0.0434
5	$U_3, U_1, U_2, U_4, U_5, U_6, U_{10}$	511.4	-368.8	0.0485	0.0399	0.0172	0.0542
6	$U_3, U_1, U_2, U_4, U_5, U_6, U_{19}$	537.98	-437.97	0.0422	0.0456	0.0328	0.0577
7	$U_3, U_1, U_4, U_2, U_9, U_6, U_5$	510.45	-358.9	0.0578	0.0435	0.0254	0.0745
8	$U_3, U_1, U_2, U_4, U_5, U_6, U_{11}$	521.94	-371.89	0.0337	0.0313	0.0204	0.0466
9	$U_3, U_1, U_2, U_4, U_5, U_6, U_8$	530.51	-397.01	0.0416	0.0427	0.0247	0.0402
10	$U_3, U_1, U_4, U_2, U_7, U_9, U_6$	469.77	-263.53	0.0586	0.0518	0.0267	0.0619
11	$U_3, U_1, U_4, U_2, U_{17}, U_{14}, U_{11}$	457.43	-244.86	0.0702	0.0466	0.0251	0.0794
12	$U_2, U_3, U_1, U_4, U_{11}, U_5, U_6$	416.68	-181.35	0.0736	0.0567	0.0147	0.0730
13	$U_2, U_3, U_4, U_1, U_5, U_6, U_8$	445.66	-235.31	0.0486	0.0647	0.0198	0.0554
14	$U_3, U_1, U_2, U_4, U_7, U_5, U_6$	500.88	-343.76	0.0525	0.0319	0.0136	0.0377
15	$U_3, U_1, U_2, U_4, U_5, U_9, U_{13}$	536.68	-441.35	0.0351	0.0252	0.0130	0.0304
16	$U_2, U_3, U_1, U_4, U_5, U_{12}, U_6$	463.43	-274.86	0.0399	0.0480	0.0168	0.0540
17	$U_3, U_1, U_2, U_4, U_9, U_5, U_6$	525.72	-409.43	0.0419	0.0349	0.0156	0.0262
18	$U_3, U_1, U_4, U_2, U_9, U_8, U_6$	461.75	-267.51	0.0728	0.0413	0.0234	0.0507
19	$U_3, U_1, U_2, U_4, U_5, U_9, U_{14}$	505.82	-355.65	0.0484	0.0493	0.0330	0.0651
20	$U_3, U_1, U_2, U_4, U_6, U_9, U_8$	501.26	-360.53	0.0519	0.0780	0.0293	0.0624

Table B.2: One step ahead forward selection algorithm statistics.

id	order	cll	cll_{AIC}	$imse_{0.5t}$	$omse_{0.1}$	$omse_{0.5t}$	$omse_{0.9}$
1	$U_1, U_2, U_3, U_4, U_5, U_9, U_{20}$	718.6	-780.32	0.0078	0.0070	0.0020	0.0099
2	$U_1, U_2, U_3, U_4, U_5, U_{19}, U_6$	669.19	-693	0.0110	0.0122	0.0038	0.0104
3	$U_1, U_2, U_3, U_4, U_5, U_9, U_6$	738.11	-809.84	0.0082	0.0072	0.0032	0.0082
4	$U_5, U_3, U_4, U_1, U_2, U_9, U_6$	426.76	-181.57	0.0818	0.0638	0.0357	0.0939
5	$U_1, U_2, U_7, U_3, U_4, U_6, U_5$	654.75	-662.48	0.0215	0.0230	0.0065	0.0134
6	$U_3, U_7, U_1, U_2, U_4, U_5, U_{10}$	523.72	-377.51	0.0452	0.0470	0.0292	0.0581
7	$U_1, U_2, U_3, U_4, U_5, U_6, U_8$	727.29	-772.18	0.0076	0.0101	0.0050	0.0113
8	$U_2, U_3, U_1, U_5, U_8, U_4, U_9$	429.43	-177.25	0.0522	0.0588	0.0205	0.0700
9	$U_3, U_1, U_2, U_4, U_7, U_8, U_5$	470.59	-280.82	0.0548	0.0705	0.0328	0.0760
10	$U_1, U_2, U_3, U_4, U_8, U_9, U_{10}$	719.55	-783.73	0.0082	0.0093	0.0025	0.0081
11	$U_1, U_2, U_3, U_4, U_5, U_{11}, U_{12}$	712.42	-769.45	0.0094	0.0063	0.0027	0.0113
12	$U_3, U_1, U_2, U_4, U_{11}, U_6, U_8$	459.11	-239.79	0.0702	0.0508	0.0137	0.0479
13	$U_5, U_3, U_1, U_2, U_4, U_{14}, U_9$	471.31	-279.79	0.0572	0.0461	0.0171	0.0424
14	$U_3, U_1, U_2, U_4, U_5, U_7, U_9$	502.32	-327.64	0.0610	0.0400	0.0190	0.0494
15	$U_3, U_1, U_2, U_4, U_{13}, U_7, U_5$	557.85	-419.76	0.0334	0.0262	0.0145	0.0283
16	$U_1, U_2, U_3, U_4, U_7, U_8, U_5$	696.8	-716.87	0.0150	0.0103	0.0071	0.0172
17	$U_1, U_2, U_3, U_4, U_5, U_{14}, U_{16}$	719.5	-789.6	0.0091	0.0119	0.0055	0.0110
18	$U_1, U_2, U_3, U_4, U_7, U_8, U_5$	671.87	-667.52	0.0109	0.0132	0.0035	0.0110
19	$U_3, U_7, U_4, U_1, U_2, U_5, U_6$	452.87	-223.61	0.0704	0.0304	0.0388	0.0799
20	$U_1, U_3, U_2, U_4, U_5, U_{11}, U_7$	476.02	-280.85	0.0455	0.0496	0.0165	0.0415

Table B.3: Batch maximum algorithm statistics.

id	order	cll	cll_{AIC}	$imse_{0.5t}$	$omse_{0.1}$	$omse_{0.5t}$	$omse_{0.9}$
1	$U_3, U_1, U_2, U_4, U_5, U_7, U_9$	492.62	-306.35	0.0584	0.0317	0.0129	0.0348
2	$U_3, U_1, U_2, U_5, U_4, U_{10}, U_{16}$	521.45	-371.52	0.0412	0.0323	0.0176	0.0364
3	$U_3, U_1, U_2, U_4, U_5, U_9, U_6$	508.69	-365.00	0.0445	0.0423	0.0149	0.0291
4	$U_3, U_1, U_2, U_5, U_8, U_4, U_{12}$	452.08	-224.22	0.0783	0.0725	0.0290	0.0777
5	$U_3, U_1, U_2, U_4, U_5, U_7, U_{10}$	499.21	-343.41	0.0631	0.0408	0.0209	0.0396
6	$U_3, U_1, U_2, U_4, U_5, U_{16}, U_{11}$	546.78	-437.64	0.0404	0.0445	0.0277	0.0430
7	$U_3, U_1, U_2, U_5, U_7, U_8, U_6$	518.39	-342.38	0.0525	0.0445	0.0247	0.0498
8	$U_3, U_1, U_2, U_5, U_9, U_8, U_{11}$	513.30	-341.00	0.0383	0.0309	0.0192	0.0463
9	$U_3, U_1, U_2, U_4, U_7, U_8, U_5$	468.46	-264.55	0.0545	0.0675	0.0325	0.0727
10	$U_3, U_1, U_2, U_4, U_{10}, U_5, U_8$	517.70	-364.02	0.0487	0.0436	0.0209	0.0498
11	$U_3, U_1, U_2, U_4, U_{17}, U_{11}, U_{19}$	488.72	-298.06	0.0637	0.0422	0.0218	0.0789
12	$U_2, U_3, U_1, U_4, U_{11}, U_5, U_9$	399.89	-133.65	0.0685	0.0695	0.0137	0.0746
13	$U_2, U_3, U_1, U_4, U_7, U_8, U_5$	437.13	-209.43	0.0511	0.0619	0.0199	0.0611
14	$U_3, U_1, U_2, U_4, U_7, U_5, U_{14}$	510.02	-351.04	0.0511	0.0392	0.0194	0.0423
15	$U_3, U_1, U_2, U_4, U_5, U_9, U_{13}$	549.11	-420.27	0.0358	0.0248	0.0142	0.0291
16	$U_3, U_1, U_4, U_2, U_5, U_{19}, U_{14}$	453.99	-227.24	0.0663	0.0594	0.0322	0.0774
17	$U_3, U_1, U_2, U_5, U_6, U_7, U_{15}$	509.68	-355.96	0.0451	0.0367	0.0150	0.0298
18	$U_3, U_1, U_2, U_4, U_7, U_5, U_9$	454.33	-238.43	0.0735	0.0545	0.0289	0.0668
19	$U_2, U_1, U_3, U_4, U_{19}, U_8, U_6$	419.58	-153.04	0.0690	0.0814	0.0374	0.0935
20	$U_2, U_3, U_1, U_4, U_5, U_{12}, U_{11}$	453.86	-226.53	0.0440	0.0823	0.0215	0.0543

Table B.4: Batch average algorithm statistics.

id	order	$imse_{0.5t}$	$omse_{0.1}$	$omse_{0.5t}$	$omse_{0.9}$
1	$U_3, U_1, U_2, U_5, U_4, U_6, U_7$	0.0815	0.0218	0.0218	0.0220
2	$U_3, U_1, U_2, U_6, U_4, U_9, U_7$	0.0604	0.0203	0.0206	0.0208
3	$U_3, U_1, U_2, U_4, U_9, U_5, U_6$	0.0692	0.0342	0.0346	0.0348
4	$U_3, U_1, U_2, U_8, U_6, U_5, U_9$	0.0682	0.0319	0.0322	0.0324
5	$U_3, U_1, U_2, U_4, U_7, U_8, U_{11}$	0.0674	0.0215	0.0219	0.0223
6	$U_3, U_1, U_2, U_4, U_5, U_9, U_{13}$	0.0578	0.0274	0.0278	0.0284
7	$U_3, U_1, U_2, U_4, U_5, U_6, U_9$	0.0561	0.0320	0.0330	0.0336
8	$U_3, U_1, U_2, U_8, U_6, U_5, U_9$	0.0579	0.0330	0.0324	0.0322
9	$U_3, U_1, U_2, U_4, U_6, U_8, U_5$	0.0620	0.0283	0.0291	0.0293
10	$U_3, U_1, U_2, U_6, U_4, U_8, U_7$	0.0481	0.0227	0.0227	0.0226
11	$U_3, U_1, U_2, U_8, U_5, U_6, U_4$	0.0664	0.0203	0.0205	0.0203
12	$U_3, U_1, U_2, U_5, U_4, U_6, U_9$	0.0916	0.0195	0.0201	0.0204
13	$U_3, U_1, U_2, U_6, U_4, U_8, U_9$	0.0636	0.0256	0.0254	0.0253
14	$U_3, U_1, U_2, U_4, U_6, U_7, U_9$	0.0673	0.0243	0.0243	0.0251
15	$U_3, U_1, U_2, U_4, U_5, U_6, U_9$	0.0543	0.0203	0.0202	0.0202
16	$U_3, U_1, U_2, U_4, U_6, U_8, U_9$	0.0631	0.0250	0.0252	0.0256
17	$U_3, U_1, U_2, U_4, U_6, U_{11}, U_7$	0.0565	0.0246	0.0251	0.0262
18	$U_3, U_1, U_2, U_6, U_4, U_8, U_9$	0.0790	0.0224	0.0223	0.0227
19	$U_3, U_1, U_2, U_6, U_8, U_4, U_7$	0.0645	0.0291	0.0298	0.0310
20	$U_3, U_1, U_2, U_6, U_9, U_7, U_4$	0.0591	0.0182	0.0194	0.0201

Table B.5: Linear quantile regression statistics.

B.2 Case study II results

The following tables provide the order of the estimated models up to 7 predictors, the cll, cll_{AIC}, the IMSE for $\alpha = 0.5$ and the OMSE for $\alpha = 0.1, 0.5, 0.9$ for each iteration.

id	order	cll	cll_{AIC}	$imse_{0.5t}$	$omse_{0.1}$	$omse_{0.5t}$	$omse_{0.9}$
1	$U_1, U_4, U_6, U_2, U_7, U_{16}, U_3$	1006.83	-1823.27	0.0899	0.0326	0.0259	0.0274
2	$U_1, U_4, U_3, U_{11}, U_6, U_5, U_2$	1039.38	-1878.07	0.0814	0.0227	0.0240	0.0274
3	$U_1, U_4, U_5, U_2, U_7, U_{11}, U_8$	1019.85	-1861.56	0.0557	0.0321	0.0284	0.0253
4	$U_1, U_4, U_5, U_2, U_8, U_7, U_3$	1023.86	-1861.97	0.1051	0.0194	0.0181	0.0183
5	$U_1, U_3, U_2, U_6, U_{11}, U_7, U_4$	1000.98	-1834.58	0.1289	0.0226	0.0201	0.0247
6	$U_1, U_3, U_2, U_6, U_7, U_{13}, U_{11}$	998.83	-1815.74	0.0945	0.0334	0.0348	0.0386
7	$U_1, U_3, U_2, U_4, U_{11}, U_7, U_{13}$	983.31	-1787.1	0.1271	0.0538	0.0300	0.0367
8	$U_1, U_3, U_2, U_6, U_{11}, U_4, U_7$	1043.19	-1904.2	0.2059	0.0627	0.0684	0.0803
9	$U_1, U_3, U_2, U_4, U_{13}, U_6, U_8$	1004.96	-1835.45	0.0800	0.0243	0.0229	0.0250
10	$U_1, U_3, U_2, U_7, U_{13}, U_6, U_8$	1073.44	-1959.87	0.1214	0.0327	0.0317	0.0436
11	$U_1, U_4, U_5, U_2, U_7, U_3, U_8$	1048.22	-1911.14	0.0633	0.0208	0.0199	0.0201
12	$U_1, U_3, U_2, U_4, U_8, U_6, U_7$	1029.82	-1881.98	0.0727	0.0280	0.0271	0.0290
13	$U_1, U_3, U_2, U_4, U_6, U_{11}, U_7$	1024.19	-1880.27,	0.1796	0.0360	0.0384	0.0503
14	$U_1, U_3, U_2, U_4, U_8, U_7, U_{11}$	1041.74	-1902.99	0.0893	0.0201	0.0208	0.0229
15	$U_1, U_4, U_5, U_2, U_8, U_7, U_{17}$	1019.72	-1845.91	0.0625	0.0196	0.0177	0.0179
16	$U_1, U_4, U_5, U_2, U_8, U_3, U_6$	1034.95	-1885.57	0.0704	0.0175	0.0195	0.0238
17	$U_1, U_3, U_2, U_6, U_7, U_8, U_{11}$	1053.58	-1926.91	0.0800	0.0183	0.0194	0.0228
18	$U_1, U_4, U_5, U_2, U_8, U_3, U_{11}$	1003.67	-1829.07	0.0869	0.0164	0.0174	0.0207
19	$U_1, U_3, U_6, U_2, U_4, U_7, U_{13}$	959.02	-1743.45	0.1944	0.0807	0.0538	0.1021
20	$U_1, U_3, U_2, U_8, U_6, U_7, U_{11}$	1009.03	-1855.41	0.1019	0.0203	0.0212	0.0244

Table B.6: Two step ahead forward selection algorithm statistics.

id	order	cll	cll_{AIC}	$imse_{0.5t}$	$omse_{0.1}$	$omse_{0.5t}$	$omse_{0.9}$
1	$U_1, U_2, U_3, U_6, U_5, U_7, U_4$	1303.28	-1960.57	0.0811	0.0197	0.0204	0.0224
2	$U_1, U_2, U_4, U_7, U_8, U_5, U_3$	1286.7	-1905.39	0.0886	0.0287	0.0299	0.0317
3	$U_1, U_2, U_3, U_6, U_7, U_8, U_4$	1332.4	-2020.8	0.0683	0.0279	0.0271	0.0269
4	$U_1, U_2, U_4, U_7, U_6, U_3, U_8$	1371.54	-2097.07	0.0858	0.0291	0.0211	0.0181
5	$U_1, U_2, U_3, U_6, U_8, U_7, U_4$	1271.84	-1877.68	0.2159	0.0485	0.0546	0.0606
6	$U_1, U_2, U_3, U_6, U_5, U_8, U_7$	1226.23	-1812.46	0.0809	0.0397	0.0400	0.0472
7	$U_1, U_2, U_3, U_6, U_5, U_8, U_4$	1211.98	-1795.96	0.9849	0.2141	0.2087	0.2021
8	$U_1, U_2, U_3, U_6, U_5, U_8, U_{11}$	1262.86	-1873.72	0.1302	0.0511	0.0535	0.0552
9	$U_1, U_2, U_3, U_6, U_7, U_8, U_4$	1352.05	-2054.09	0.0770	0.0309	0.0260	0.0256
10	$U_1, U_2, U_3, U_6, U_5, U_9, U_8$	1297.48	-1956.95	0.0549	0.0360	0.0300	0.0272
11	$U_1, U_2, U_4, U_7, U_8, U_5, U_6$	1259.59	-1871.18	0.6066	0.1780	0.1769	0.1721
12	$U_1, U_2, U_3, U_6, U_5, U_8, U_7$	1286.66	-1929.31	0.0530	0.0369	0.0300	0.0316
13	$U_1, U_2, U_3, U_6, U_7, U_8, U_4$	1455.87	-2257.73	0.1889	0.0393	0.0405	0.0417
14	$U_1, U_2, U_4, U_7, U_8, U_3, U_5$	1468.29	-2260.57	0.0695	0.0257	0.0231	0.0226
15	$U_1, U_2, U_3, U_6, U_7, U_8, U_{12}$	1330.99	-2013.98	0.0906	0.0261	0.0235	0.0248
16	$U_1, U_2, U_4, U_7, U_8, U_3, U_6$	1466.52	-2281.03	0.0676	0.0222	0.0218	0.0206
17	$U_1, U_2, U_4, U_7, U_3, U_6, U_5$	1330.91	-2007.83	0.0862	0.0242	0.0225	0.0242
18	$U_1, U_2, U_3, U_6, U_5, U_7, U_4$	1731.62	-2821.23	0.0897	0.0243	0.0247	0.0251
19	$U_1, U_2, U_3, U_6, U_5, U_8, U_7$	1245.23	-1816.47	0.0975	0.0211	0.0197	0.0200
20	$U_1, U_2, U_3, U_6, U_7, U_8, U_4$	1276	-1911.99	0.1166	0.0426	0.0339	0.0361

Table B.7: One step ahead forward selection algorithm statistics.

id	order	cll	cll_{AIC}	$imse_{0.5t}$	$omse_{0.1}$	$omse_{0.5t}$	$omse_{0.9}$
1	$U_1, U_5, U_2, U_4, U_7, U_8, U_{12}$	1024.03	-1877.65	0.0656	0.0210	0.0190	0.0198
2	$U_1, U_3, U_4, U_2, U_5, U_8, U_{14}$	1032.98	-1865.28	0.0544	0.0377	0.0362	0.0377
3	$U_1, U_5, U_3, U_2, U_{11}, U_8, U_{14}$	1010	-1825.86	0.1081	0.0446	0.0437	0.0447
4	$U_7, U_1, U_3, U_4, U_5, U_2, U_8$	890.89	-1580.02	0.1993	0.0261	0.0255	0.0284
5	$U_1, U_7, U_5, U_2, U_8, U_4, U_3$	1006.72	-1830.06	0.0615	0.0189	0.0195	0.0217
6	$U_1, U_4, U_5, U_2, U_8, U_7, U_3$	974.83	-1767.74	0.1021	0.0349	0.0352	0.0386
7	$U_1, U_5, U_4, U_2, U_8, U_6, U_7$	991.26	-1803	0.1282	0.0263	0.0248	0.0253
8	$U_1, U_3, U_2, U_6, U_4, U_{11}, U_5$	1039.79	-1899.39	0.0820	0.0232	0.0232	0.0253
9	$U_1, U_5, U_2, U_8, U_7, U_4, U_{11}$	1037.95	-1897.41	0.0724	0.0244	0.0246	0.0268
10	$U_1, U_{11}, U_3, U_6, U_2, U_{16}, U_{13}$	1037.11	-1889.21	0.0677	0.0416	0.0354	0.0401
11	$U_1, U_5, U_3, U_8, U_3, U_4, U_6$	1070.5	-1945.7	0.1633	0.1396	0.1047	0.0491
12	$U_1, U_3, U_2, U_6, U_7, U_4, U_{11}$	1028.42	-1877.19	0.0977	0.0295	0.0282	0.0361
13	$U_7, U_1, U_3, U_2, U_4, U_6, U_{11}$	977.91	-1755.7	0.1009	0.0348	0.0336	0.0369
14	$U_7, U_{11}, U_4, U_5, U_2, U_1, U_8$	911.13	-1621.78	0.0736	0.0285	0.0285	0.0323
15	$U_1, U_3, U_2, U_7, U_6, U_8, U_4$	979.21	-1774.88	0.1274	0.0715	0.0483	0.0212
16	$U_1, U_4, U_9, U_6, U_{11}, U_3, U_2$	1026.87	-1855.41	0.0633	0.0185	0.0169	0.0190
17	$U_1, U_5, U_2, U_8, U_7, U_{17}, U_9$	1085.65	-1991.05	0.0822	0.0199	0.0167	0.0158
18	$U_1, U_7, U_5, U_2, U_8, U_3, U_6$	1011.3	-1846.33	0.0494	0.0200	0.0186	0.0204
19	$U_1, U_4, U_{11}, U_6, U_5, U_2, U_3$	939.37	-1702.15	0.3151	0.1920	0.1008	0.0454
20	$U_1, U_4, U_{11}, U_3, U_6, U_2, U_{16}$	1000.25	= -1839.85	0.0840	0.0345	0.0330	0.0343

Table B.8: Batch maximum algorithm statistics.

id	order	cll	cll_{AIC}	$imse_{0.5t}$	$omse_{0.1}$	$omse_{0.5t}$	$omse_{0.9}$
1	$U_1, U_5, U_2, U_4, U_6, U_7, U_3$	1021.31	-1872.21	0.0602	0.0161	0.0163	0.0185
2	$U_1, U_5, U_2, U_3, U_8, U_4, U_6$	1053.26	-1915.83	0.0667	0.0195	0.0208	0.0236
3	$U_1, U_5, U_3, U_2, U_4, U_{11}, U_7$	1011.76	-1839.39	0.0517	0.0350	0.0313	0.0296
4	$U_1, U_5, U_2, U_4, U_7, U_8, U_{11}$	1026.78	-1885.81	0.0930	0.0201	0.0180	0.0182
5	$U_1, U_5, U_2, U_3, U_4, U_7, U_6$	1013.72	-1858.07	0.0863	0.0178	0.0187	0.0216
6	$U_1, U_5, U_2, U_3, U_4, U_7, U_9$	1010.38	-1846.84	0.0796	0.0342	0.0347	0.0369
7	$U_1, U_5, U_2, U_4, U_8, U_6, U_7$	1000.1	-1830.68	0.1052	0.0265	0.0252	0.0258
8	$U_1, U_5, U_2, U_4, U_7, U_6, U_8$	1076.95	-1973.71	0.0624	0.0241	0.0248	0.0268
9	$U_1, U_5, U_2, U_3, U_8, U_6, U_{11}$	1038.71	-1880.95	0.0734	0.0243	0.0249	0.0292
10	$U_1, U_5, U_2, U_3, U_6, U_4, U_8$	1095.93	-1998.84	0.0559	0.0275	0.0268	0.0273
11	$U_1, U_5, U_2, U_4, U_3, U_6, U_8$	1061.72	-1940.14	0.0612	0.0294	0.0223	0.0218
12	$U_1, U_5, U_2, U_3, U_7, U_8, U_6$	1040.13	-1894.61	0.0578	0.0263	0.0266	0.0285
13	$U_1, U_5, U_2, U_3, U_4, U_7, U_8$	1040.89	-1909.67	0.0837	0.0310	0.0270	0.0559
14	$U_1, U_5, U_2, U_4, U_8, U_6, U_7$	1050.57	-1928.66	0.0784	0.0194	0.0195	0.0208
15	$U_1, U_5, U_2, U_3, U_7, U_8, U_4$	1014.32	-1849.12	0.0448	0.0275	0.0199	0.0215
16	$U_1, U_5, U_2, U_4, U_8, U_{11}, U_7$	1034.43	-1890.53	0.0442	0.0201	0.0189	0.0191
17	$U_1, U_5, U_2, U_4, U_8, U_7, U_6$	1078.17	-1982.09	0.0702	0.0143	0.0158	0.0212
18	$U_1, U_5, U_2, U_4, U_3, U_6, U_8$	1008.2	-1858.12	0.1762	0.1130	0.0916	0.0807
19	$U_1, U_5, U_2, U_8, U_4, U_3, U_6$	990.18	-1809.76	0.0848	0.0282	0.0202	0.0181
20	$U_1, U_5, U_2, U_3, U_8, U_6, U_4$	1025.67	-1896.7	0.0763	0.0194	0.0203	0.0224

Table B.9: Batch average algorithm statistics.

id	order	$imse_{0.5t}$	$omse_{0.1}$	$omse_{0.5t}$	$omse_{0.9}$
1	$U_1, U_2, U_7, U_5, U_6, U_8, U_{17}$	0.0146	0.0337	0.0337	0.0337
2	$U_1, U_2, U_4, U_7, U_5, U_{17}, U_3$	0.0132	0.0268	0.0268	0.0268
3	$U_1, U_2, U_5, U_8, U_3, U_6, U_7$	0.0114	0.0410	0.0410	0.0410
4	$U_1, U_2, U_4, U_7, U_5, U_8, U_{17}$	0.0153	0.0431	0.0431	0.0431
5	$U_1, U_2, U_5, U_7, U_4, U_8, U_6$	0.0143	0.0376	0.0376	0.0376
6	$U_1, U_2, U_5, U_4, U_8, U_6, U_7$	0.0163	0.0439	0.0439	0.0439
7	$U_1, U_2, U_5, U_7, U_9, U_3, U_8$	0.0156	0.0461	0.0461	0.0461
8	$U_1, U_2, U_4, U_5, U_8, U_7, U_9$	0.0108	0.0436	0.0436	0.0436
9	$U_1, U_2, U_4, U_5, U_{11}, U_7, U_3$	0.0166	0.0386	0.0386	0.0386
10	$U_1, U_2, U_5, U_7, U_3, U_{11}, U_8$	0.0110	0.0453	0.0453	0.0453
11	$U_1, U_2, U_7, U_3, U_5, U_8, U_{11}$	0.0127	0.0380	0.0380	0.0380
12	$U_1, U_2, U_5, U_4, U_{11}, U_3, U_6$	0.0123	0.0444	0.0444	0.0444
13	$U_1, U_2, U_5, U_7, U_8, U_{17}, U_3$	0.0131	0.0339	0.0339	0.0339
14	$U_1, U_2, U_5, U_4, U_7, U_8, U_3$	0.0127	0.0291	0.0291	0.0291
15	$U_1, U_2, U_5, U_4, U_6, U_{11}, U_3$	0.0132	0.0330	0.0330	0.0330
16	$U_1, U_2, U_4, U_5, U_3, U_8, U_7$	0.0114	0.0353	0.0353	0.0353
17	$U_1, U_2, U_7, U_5, U_4, U_{11}, U_{17}$	0.0155	0.0337	0.0337	0.0337
18	$U_1, U_2, U_3, U_7, U_8, U_{11}, U_{17}$	0.0146	0.0319	0.0319	0.0319
19	$U_1, U_2, U_5, U_7, U_6, U_{17}, U_8$	0.0143	0.0330	0.0330	0.0330
20	$U_1, U_2, U_4, U_5, U_7, U_3, U_6$	0.0171	0.0307	0.0307	0.0307

Table B.10: Linear quantile regression statistics.