Delft University of Technology

Non Simplified Vine Copulas using Generalised Additive Models

Delft University of Technology, Delft, South Holland, 2628CD

Amadeo Villar (5377447)

July 14, 2021

Submission date:	July 14, 2021
Course:	WI4050 Uncertainty and Sensitivity Analysis
Project:	4
Supervisors:	Dr. D. Kurowicka [1]

[1] Assistant Professor, Faculty of Electrical Engineering, Mathematics and Computer Science, Group of Applied Probability, Delft University of Technology.

Contents

1 Introduction

Due to the technological breakthrough of the last decades and the rapid increase in the availability of multidimensional data, massive data analysis[2] has become one of the most important areas of research today. The applications of this field are broad, ranging from weather forecasting or analysis of the human genome to financial systems. Within the field of big data, dependence modeling is having special interest in recent years. Knowing the values of certain variables, we can estimate with some accuracy the values of other dependent variables, thus reducing the dimensions of the problem.

We have studied throughout this course that one of the fundamentals tools for dependency modeling are the so-called copulas[3]. These mathematical constructions allow for separate modelling of marginal distributions and dependence structure [1]. While it is easy to construct bivariate copulas, the construction and modeling of multivariate copulas has been a great challenge with multiple improvements in recent years. Indeed, we already studied the lack of flexibility offered by multidimensional copulas such as the n-dimensional normal copula, or the high complexity of the hierarchical Archimedean nested copulas. We also looked deeply into Markov trees distributions [2], which introduced as a novelty the construction of multivariable models based on 2d-piece models.

However, it was when conditional dependencies between random variables were allowed in Markov trees that a highly flexible models arise. These models, known as Vine Copulas, and introduced by Bedford and Cooke [3], constitute the most powerful models we have to tackle the problem of multivariate dependence modelling. Due to the great flexibility of these models, it might be advisable to work using the simplifying assumptions: pair-copulas of conditional distributions are independent of the values of the variables on which they are conditioned. Therefore under these assumptions the complete joint distribution can be built using unconditional bivariate copulas.

There is currently some controversy regarding these assumptions. On the one hand it cannot be denied that simplified vine copulas are still highly flexible models [4]. On the other hand, some mathematicians have criticised this assumption for being too optimistic [5], [6]. The main objective of this project is to implement non-simplified vine copulas and study their applicability and their differences with respect to the simplified ones [7]. Moreover, to carry out the implementation and model the dependency with the conditioning set (non-simplified assumption), we use generalised additive models [8], [9].

The structure of this project is as follows. We start from a theoretical basis and end up showing practical results later on. In Section 2 we introduce the concept of non-simplified assumptions. Then, in Section 3 we discuss how we can extend vine copulas using generalised additive models, to model the non-simplified assumption. Later Section 4 and Section 5 are of a more practical nature. In the former, we present the construction of a three-dimensional non-simplified vine copula. In the latter, we carry out the implementation of the previous model [10], [11], [12], [13], to check differences between the 2 assumptions. Finally in Section 6 we summarize the most important results and draw a conclusion from them.

[2] Also known as Big Data, https://en.wikipedia.org/wiki/Big_data

[3] https://en.wikipedia.org/wiki/Copula_(probability_theory)

2 Vine Copulas

In this section we first introduce the concept of copula. Then, we study from a mathematical point of view vine copula construction models. Finally we comment on the simplifying assumptions in Subsection 2.1 and on the non-simplifying assumptions in Subsection 2.2.

A copula is a multivariate cumulative distribution function on the unit hypercube $[0,1]^d$, which has uniform margins. The importance of this mathematical object resides in Sklar's Theorem [1].

Theorem 2.1. *Let $X = (X_1, \ldots, X_d)$ be random vector with joint distribution function F and with margins $F_1, \ldots, F_d$. Then there exists a copula $C : [0,1]^d \to [0,1]$ such that, $\forall\ x_1, \ldots, x_d \in \mathbb{R}^d$,*

$$F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)). \tag{1}$$

If $X_1, \ldots, X_d$ are continuous then C is unique.

Differentiating the previous expression if possible, we get:

$$f(x_1, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)) \prod_{i=1}^{f} f_i(x_i), \tag{2}$$

where c is the copula density function and f_i are the marginal densities.

In this context, any copula density c can be decomposed into a product of $d \times (d-1)/2$ bivariate copula densities [3]. These decompositions can be organized using graphical models that we call Regular vines, namely a sequence of trees $T_m = (V_m, E_m)$. Below we introduce a series of concepts[4] that will help to better understand the concept of vine copula and the simplifying assumptions.

Definition 2.1. *V is a regular vine on d elements if:*

1. $V = \{T_1, ..., T_{d-1}\}$
2. *T_1 is a tree with nodes $N_1 = \{1, \ldots, d\}$, and edges E_i; for $i = 2, \ldots, d-1$, T_i is a tree with nodes $N_i = E_{i-1}$.*
3. *(regularity) for $i = 2, \ldots, d-1$, $\{a, b\} \in E_i$ and $a = \{a_1, a_2\}$, $b = \{b_1, b_2\}$ then exactly one of the a_i equals one of the b_i.*

Definition 2.2. *The constraint set, the conditioning set and the conditioned set are defined as:*

1. *The constraint set associated with $e = \{j, k\} \in E_i$ is a subset of $\{1, 2, ..., d\}$ reachable from e by inclusion (membership) relationship denoted by U_e^*.*

[4] All the definitions and theorems showed in this section come from the lectures slides of the TU Delft Uncertainty and Sensitivity course.

2. *For $i = 1, \ldots, d-1, e \in E_i$, $e = \{j, k\}$, the conditioning set associated with e is:*

$$D_e = U_j^* \cap U_k^*,$$

and the conditioned set associated with e is:

$$\{C_{e,j}, C_{e,k}\} = U_j^* \triangle U_k^* = \{U_j^* \backslash D_e, U_k^* \backslash D_e\}$$

The order of node e is $\#D_e$.

Theorem 2.2. *Let $V = (T_1, \ldots, T_{d-1})$ be a regular vine on d elements. For each edge $e(j,k) \in T_i$, $i = 1, \ldots, d-1$ with the conditioned set given by $\{j, k\}$ and the conditioning set given by D_e we write the corresponding copula $C_{j,k|D_e}$ and its density $c_{j.k|D_e}$. Then the unique vine dependent distribution has a copula density given by:*

$$c(u_1, \ldots, u_d) = \prod_{i=1}^{d-1} \prod_{e(j,k) \in E_i} c_{j,k|D_e}(C_{j|D_e}, C_{k|D_e}; u_{D_e}). \tag{3}$$

So each of the bivariate copulas $c_{j,k|D_e}$ of the previous decomposition, describes the dependence between the random variables U_j and U_k conditional on U_{D_e}.

Below we show an example of a 5 dimensional vine copula that we found at [9].

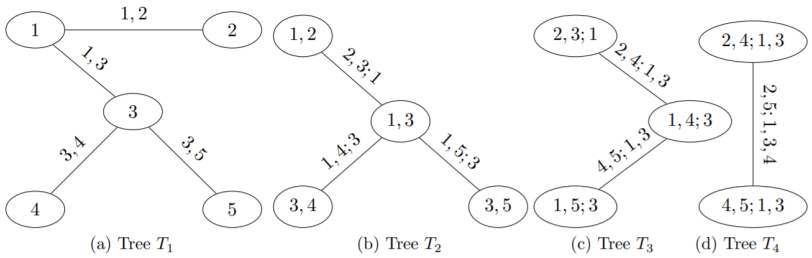


Figure 1: Example of a Regular Vine Tree Sequence.

The copula density associated with the Vine in Figure 1 is given by:

$$\mathrm{c}(\mathrm{u}_1, ..., u_5) = c_{1,2}(u_1, u_2) \times c_{1,3}(u_1, u_3) \times c_{3,4}(u_3, u_4) \times c_{3,5}(u_3, u_5) \times c_{2,3;1}(u_{2|1}, u_{3|1}; u_1) \times c_{1,4;3}(u_{1|3}, u_{4|3}; u_3) \times c_{1,5;3}(u_{1|3}, u_{5|3}; u_3) \times c_{2,4;1,3}(u_{2|1,3}, u_{4|1,3}; u_{1,3}) \times c_{4,5;1,3}(u_{4|1,3}, u_{5|1,3}; u_{1,3}) \times c_{2,5;1,3,4}(u_{2|1,3,4}, u_{5|1,3,4}; u_{1,3,4}).$$

2.1 Simplifying Assumptions

We can see how as we advance in the trees, the number of variables of the conditioning set grows, obtaining expressions that can be very complex. This fact sometimes makes it convenient to ignore the influence that u_{D_e} has on the bivariate copula density $c_{j,k|D_e}$. These are the so-called simplifying assumptions, and mathematically can be understood as:

$$c_{j,k|D_e}(C_{j|D_e}, C_{k|D_e}; u_{D_e}) = c_{j,k|D_e}(C_{j|D_e}, C_{k|D_e})$$

Another reason why it is sometimes convenient to make these assumptions is that the simplifying vine copula models are still highly flexible and capable of modeling many types of dependencies [4]. These models still capture the main features of the data and provide smooth fits [7]. Moreover, they are less computationally expensive, enabling fast and robust fittings. To carry out computations with these simplified vine copula the `VineCopula` package in R is extremely useful.

We can then understand these assumptions as a trade between accuracy and efficiency. Therefore under the simplifying assumptions the copula density can be expressed as:

$$c(u_1, \dots, u_d) = \prod_{i=1}^{d-1} \prod_{e(j,k)\in E_i} c_{j,k|D_e}(C_{j|D_e}, C_{k|D_e}) \tag{4}$$

2.2 Non-Simplifying Assumptions

On the other hand, some authors have showed that there are certain dependent structures that cannot be correctly modeled using the simplified assumption [5], [6]. For instance, 3d non simplified vine copulas can model quite irregular contour shapes, displaying twists, bumps and changing dependence patterns [7]. In addition, it has been seen that these models can adequately fit real data, and account for complex dependencies very well.

Another great advantage of non-simplified copulas is that it allow us to introduce the effect of covariates. This is very useful in real life applications, where exogenous variables such as time or space may have some influence on the dependent structure. For example, when modeling certain economic variables, it is essential to take into account time dependence due to the cyclical nature of market activity [9]. Mathematically, the influence of the covariates can be expressed as:

$$c(u_1, \dots, u_d; \boldsymbol{\omega}) = \prod_{i=1}^{d-1} \prod_{e(j,k)\in E_i} c_{j,k|D_e}(C_{j|D_e,\boldsymbol{\omega}}, C_{k|D_e,\boldsymbol{\omega}}; u_{D_e}, \boldsymbol{\omega}), \tag{5}$$

where $\boldsymbol{\omega}$ is a vector containing the exogenous covariates. The introduction of exogenous covariates is quite useful is real life problem. Despite this fact, in this project we will only deal with models where covariates are given by the conditioning set, as in (3).

Once the concepts of vine copula and non-simplified assumptions have been introduced, we will proceed on how can we introduce the influence of the conditioning sets on the bivariate copulas. The answer to this question can be found in Section 3.

3 Generalised Additive Models for Vine Copula Construction

In this section we study how can we model the influence of the conditioning sets on the bivariate copulas. That is the influence of D_e on $c_{j,k|D_e}(C_{j|D_e}, C_{k|D_e}; u_{D_e})$. The main idea is to allow the parameters of conditional copulas to depend on the conditioning variables via generalized additive models [14]. These pioneering ideas were the brainchild of Vatter and Chavez-Demoulin [8] and a couple of years later, Vatter and Nagler [9] created the `gamCopula` package in R to deal with implementation. Once these ideas have been presented, we will discuss Parameter Estimation in Subsection 3.1, Model Selection in Subsection 3.2, Sequential Estimation in Subsection 3.3 and Structure Selection in Subsection 3.4, of these models.

In this setting, we can model the variation of the dependence parameter with respect to the conditioning set as:

$$\tau_e(\mathbf{u_{D_e}}, \theta) = g\Big\{ \underbrace{\mathbf{z}^t\beta}_{lin.model} + \underbrace{\sum_{k=1}^{K} s_k(\mathbf{t_k})}_{non.lin.model} \Big\}, \tag{6}$$

where:

- τ_e is the Kendall's tau of the conditional copula.
- $\mathbf{u}_{D_e}$ is the conditioning set of variables and $\theta = (\beta, s_k)$ is the vector of stacked parameters.
- $g(x) = (e^x - 1)/(e^x + 1)$, is the Fisher z-transform. Is the link between the gam and the Kendall's τ. Moreover, the image of this function is $[-1, 1]$, which is precisely the allowed values for τ.
- z and t_k are subsets of $\mathbf{u}_{D_e}$ or products thereof (to allow interactions).
- $\beta \in \mathbb{R}^p$ is a vector of parameters which accounts for the linear dependent part.
- $s_k(\cdot)$ are smooth functions which accounts for the non-linear dependent part.

For example, let's consider a Gumbell copula with parameter α to model the dependence between r.v (u_1, u_2), depending on the conditioning set $\mathbf{u}_{D_e} = (u_3, u_4, u_5)$. Let's take $\mathbf{z} = (u_3, u_4)$, $\beta = (2, 5)$, $t_k = u_5$, and $s_1(x) = 3x^2$. The density function of Gumbell is given by:

$$c(u_1, u_2) = 1 + \alpha(1 - 2u_1)(1 - 2u_2),$$

and the relation between the Kendall's τ and the parameter α is:

$$\alpha = \frac{1}{1 - \tau}.$$

Then the explicit expression of the copula[5] is:

$$c(u_1, u_2|u_3, u_4, u_5) = 1 + \frac{1}{1 - \frac{e^{2u_3+5u_4+3u_5^2-1}}{e^{2u_3+5u_4+3u_5^2+1}}}.(1 - 2u_1)(1 - 2u_2)$$

[5] highlighting the complexity of the expressions for non-simplified copulas

We then proceed to study different aspects of its implementation, such as parameter estimation, sequential estimation, model selection and structure selection.

3.1 Parameter Estimation

First of all, we study parameter estimation. Let $c(\mathbf{u}; \mathbf{u}_{D_e})$ be a determined copula, where $\mathbf{u} \in [0,1]^2$, and $\mathbf{u}_{D_e}$ is the conditioning set. Then for $\theta = (\beta, s_1, \ldots, s_K)$, the loglikelihood is given by:

$$l(\mathbf{u}, \mathbf{u}_{D_e}, \theta) = logc(\mathbf{u}; \tau(\mathbf{u}_{D_e}; \theta))$$

Considering a random sample of n observations $\{\mathbf{u}^j, \mathbf{u}^j_{D_e}\}_{j=1}^n$, then the log-likelihood is:

$$l(\theta) = \frac{1}{n} \sum_{j=1}^{n} l(\mathbf{u}^j, \mathbf{u}^j_{D_e}, \theta)$$

Lastly, assuming that all smooth functions s_k are twice continuously differentiable, we add roughness penalties in order to obtain smooth estimates. In this case the penalised log-likelihood is given by:

$$l(\theta, \gamma) = l(\theta) - \frac{1}{2} \sum_{k=1}^{K} \gamma_k \int s_k''(t_k)^2, \tag{7}$$

where $\gamma \in \mathbb{R}^+ \cup \{0\}^K$, is a vector of smoothing parameters. Then the penalized maximum log-likelihood estimator is

$$\hat{\theta} = argmax_{\theta \in \Theta} l(\theta, \gamma) \tag{8}$$

In most cases it is impossible to obtain an explicit expression for $\hat{\theta}$ in (8). For this reason, Fisher's scoring algorithm[6] is typically used to tackle the problem. In the same way, it has been shown [8] that this algorithm is roughly equivalent to performing a generalized ridge regression. The main advantage of this method is that we can define equivalent degrees of freedom (EDF) to represent the complexity of each smooth function. Finally, at each step of the Fisher algorithm we minimize cross validation for the choice of penalty γ. More detailed mathematical explanations of these methods can be found in [9].

3.2 Model Selection

We have seen that the estimators of the smoothing parameters are given by (8). The problem with this expression is that it is not at all informative about the form of the gam. Several questions arise:

1. Which of the covariates should be considered unimportant?
2. What is the appropriate basis size and corresponding smoothing parameter for each of the smooth functions?

[6] https://en.wikipedia.org/wiki/Scoring_algorithm

The answers to these questions are studied in [9].

1. - First, we set $m_k = 10$ the basis size for each component.
   - Second, we use the following algorithm: we start with all the conditioning set and we compute the p-values associated to this conditioning set using a Wald test. We then eliminate those variables for which: p-value $> \alpha$[7] and re-estimate the model. We iterate until all the conditioning set are significantly different from zero.
   - Third, terms for which $EDFs \approx 1$ will be treated as linear component of the algorithm.

   This algorithm is a variant of backward elimination[8]

2. Once the previous procedure has been carried out, the basis size of the smooth components is usually not critical. This value must be large enough to capture the underlying features of the data, but small enough to maintain computational efficiency. Information about how the algorithm that choose the basis size works, can be found in [9].

3.3 Sequential Estimation

Once we have seen how parameter estimation and model selection work, we study how to proceed with estimation of a non-simplified vine copula. We assume that $\mathbf{u}^i = (u_1^i, \dots, u_d^i)$ with $i = 1, \dots, n$, and the vine structure is known. The pair copulas of the first tree can be easily estimated using MLE and then choosing the copula family with the lowest AIC\BIC[9]. Let's note that as we are in the first tree, the conditioning set is null, so there is no need to introduce penalties.

This procedure is not so simple for trees T_m with $m \geq 2$. In the first place the conditioning set is not null any longer. In addition, the data from densities $c_{j,k|D_e}$ are unobserved. To solve the first problem we can use the methods studied in Subsection 3.1 and Subsection 3.2 to obtain estimators for θ_{D_e}. To tackle the second problem, we can sequentially construct pseudo-observations using the fitted copulas of the previous tree. The procedure is:

$$u_{j|D_e} = C_{j|D_e}(u_j|\mathbf{u}_{D_e}, \hat{\theta}_{j,D_e}) \tag{9}$$

As we can infer from (9), the estimators of each pair copula in T_m are required to compute pseudo-observations to fit copulas in T_{m+1}. So the procedure could be summarized using the following algorithm:

[7] pre-specified level
[8] for more information check https://en.wikipedia.org/wiki/Stepwise_regression
[9] trade-off between goodness of fit and model complexity

Algorithm 1 Sequential Estimation

Input: Observations $\mathbf{u}^i = (u_1^i, \ldots, u_d^i)$, Vine structure.
Output: Fitted Non-simplified Vine Copula.
for $m = 1, \ldots, d-1$ **do**
  Consider the tree T_m.
  for $i = 1, \ldots, d-m$ **do**
    Consider the edge i.
    a) Estimate a GAM using penalised-loglikelihood for each copula family
    b) Use the AIC\BIC to choose a copula family
    c) Use the estimates to construct pseudo-observations for the next tree
  end
end

3.4 Structure Selection

To conclude this section, we will comment briefly on the Structure Selection of the Vine Copula. The number of possible Vine structures grows factorially with dimension. Indeed, we can check that:

$$\text{Number of possible structures} = \binom{d}{2} (d-2)!\, 2^{\binom{d-2}{2}} \tag{10}$$

Table 1: Number of possible regular vine structures depending on the vine dimension.

Dimension	N of vine structures
2	1
3	3
4	24
5	480
6	23040

For low dimensions, all possible structures can be checked and those with the best AIC values can be chosen. However, for high dimensions this is computationally infeasible. The process to choose the structure is heuristic. It consists of connecting the most dependent variables in the first tree, and the less dependent ones in the upper trees. Nevertheless, this procedure is not optimal, and it is currently a research field.

4 Building the Model

In this section we are going to build a concrete vine copula model, in order to illustrate in a practical way, what has been studied theoretically in Section 2 and Section 3. To carry out this implementation we have used the mathematical software R. Particularly the libraries `gamCopula` [10], `VineCopula` [11], `pacotest` [12] and `mgcv` [13],

4.1 Setup

The model that we study is a 3-dimensional vine copula with non-simplified assumptions, where we model the dependence between the uniform random variables X_1, X_2 and X_3.

The first tree is specified by the copulas:

- $C_{1,2}$:= Clayton Copula with constant parameter θ, modelling the dependence between X_1 and X_2, where:

$$C_{1,2}(u_1, u_2; \theta) = \max\left\{(u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}, 0\right\}, \tag{11}$$

$$c_{1,2}(u_1, u_2; \theta) = (1+\theta)(u_1 u_2)^{-1-\theta}(u_1^{-\theta} + u_2^{-\theta} - 1)^{-2-\frac{1}{\theta}}. \tag{12}$$

- $C_{2,3}$:= Gumbell Copula with constant parameter α, modelling the dependence between X_2 and X_3, where:

$$C_{2,3}(u_2, u_3; \alpha) = \exp\Big\{-\left((-\log u_2)^\alpha + (-\log u_3)^\alpha\right)^{1/\alpha}\Big\}, \tag{13}$$

$$c_{2,3}(u_2, u_3; \alpha) = 1 + \alpha(1-2u_2)(1-2u_3). \tag{14}$$

The second tree is specified by the following non-simplified copula:

- $C_{1,3|2}$:= Normal copula with parameter ρ, modelling the conditional dependence between X_1 and X_3 given X_2, where:

$$C_{1,3|2}(u_1, u_3; \rho(u_2)) = \mathbf{\Phi}_\rho\left(\Phi^{-1}(u_1), \Phi^{-1}(u_3)\right), \tag{15}$$

$$c_{1,3|2}(u_1, u_3; \rho(u_2)) = \frac{1}{\sqrt{1-\rho^2}} \exp\left\{\frac{-\rho^2(x_1^2 + x_3^2) - 2\rho x_1 x_3}{2(1-\rho^2)}\right\}, \; x_i = \Phi^{-1}(u_i). \tag{16}$$

In this second tree, due to the non-simplified assumptions, the copula parameter will not be constant, but will depend on the variable X_2. The way of introducing that dependence is as we saw in Section 3, through the Kendall's tau parameter.

$$\tau(z) = \frac{e^z - 1}{e^z + 1}, \tag{17}$$

Figure 2: 1^{st} and 2^{nd} trees of our Vine Copula model.

where z is a function of $X_2 \rightarrow z = f(X_2)$.

In this problem we will focus our efforts on the sinusoidal case, given by:

$$z = \sin 2\pi x_2. \tag{18}$$

Lastly, the dependence between the parameter of a copula and the value of Kendall's tau can be calculated using the formula: $\tau = 4 \int CdC - 1$. For the copula families of this model, we have:

$$\text{Clayton Copula} \Rightarrow \theta = \frac{2\tau}{1-\tau} \Longleftrightarrow \tau = \frac{\theta}{2+\theta} \tag{19}$$

$$\text{Gumbell Copula} \Rightarrow \alpha = \frac{1}{1-\tau} \Longleftrightarrow \tau = \frac{\alpha - 1}{\alpha} \tag{20}$$

$$\text{Normal Copula} \Rightarrow \rho = \sin \frac{\pi}{2}\tau \Longleftrightarrow \tau = \frac{2}{\pi} \arcsin \rho \tag{21}$$

Plugging (17) and (18) into the previous expression for the normal copula, we obtain that the dependence of ρ on X_2 is given by:

$$\rho = \sin \left(\frac{\pi}{2} \frac{\exp\{\sin 2\pi x_2\} - 1}{\exp\{\sin 2\pi x_2\} + 1} \right) \tag{22}$$

4.2 Building the Model

Once the setup is done we proceed to build the model. Below we show in detail all the steps performed, in order for the reader to grasp all the ideas and to help him/her to implement models of this type. We also include some articles where the authors deepen in the subject we are dealing with.

1. The first step is to represent the vine copula structure using a matrix. This procedure is illustrated in [15], where an example of 7 dimensional R-Vine copula is carried out. The idea is as follows: for a d-dimensional Vine copula with r.v $X_1, \ldots, X_d$, we start from a lower diagonal matrix $M \in \mathbb{R}^{d\times d}$. The elements of this matrix fulfill $m_{i,j} \in \{1, 2, \ldots, d\}$ $i, j = 1, \ldots, d$. In addition, the diagonal elements are not repeated.

   - To specify the first tree we will have to take into account the diagonal and the last row belonging to the first $d-1$ columns. So these 2 elements of each column form an edge of the tree. In particular the edges of the first tree will be given by the pairs: $\{m_{1,1}, m_{d,1}\}$, $\{m_{2,2}, m_{d,2}\}$, $\ldots$,$\{m_{d-1,d-1}, m_{d,d-1}\}$.
   - For the second tree we take the elements of the diagonal and of the penultimate row of the first d-2 columns. In this case these 2 elements of each column will form an edge of the second tree. The conditioning set for each pair will be given by the element of the last row corresponding to that column. In particular the edges of the second tree will be given by the pairs: $\{m_{1,1}, m_{d-1,1}|m_{d,1}\}$, $\{m_{2,2}, m_{d-1,2}|m_{d,2}\}$, $\ldots$,$\{m_{d-2,d-2}, m_{d-1,d-2}|m_{d,d-2}\}$.
   - Proceeding in the same way, for the k-th tree, the edges will be given by the elements of the diagonal and of the $d-k$ row of the $d-k-1$ first columns. The conditioning set for each edge will be given by the elements of the last $k-1$ rows corresponding to that column.
   - Continuing we would arrive to the fact that the last tree, of a single edge is given by: $\{m_{1,1}, m_{2,1}|m_{3,1}, \ldots, m_{d,1}\}$

   For our 3-dimensional case described in Subsection 4.1 it is easy to observe that the matrix M is given by:

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

2. The second step is to define the copula family to which each of the previously defined edges belongs. For this we will use another matrix $T \in \mathbb{R}^{d\times d}$, strictly lower diagonal. The diagonal of this matrix will be zero. This matrix is related to the previous M matrix in the sense that the element $t_{i,j}$, with $i < j$, will define the copula family of the edge $\{m_{i,i}, m_{j,i}|m_{j+1,i}, \ldots, m_{d,i}\}$. The family will be determined by the value of the matrix element $t_{i,j}$. The value code for each family can be found in [11].

   For the copula families that we are working in this case we have: 1 := Normal Copula, 301 := Clayton Copula and 401 := Gumbel Copula. So the matrix T is given by:

$$T = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 301 & 401 & 0 \end{pmatrix}$$

3. For a vine copula with simplified assumptions, we would have to define another matrix P specifying the family parameter of each of the edges. So the copula would be fully characterized by these 3 matrices. In the case of non-simplified assumptions, the parameters of the copulas belonging to the $2^{nd}, \ldots, d^{th}$ trees depend on the conditioning variables. Therefore we can really only set as constants the parameters of the first tree. For the rest of the copulas, we can establish a relationship between their parameter and their conditioning sets, using generalized additive models [13]. Then, we proceed as follows:

   - First we specify the parameters of the first tree.
   - Secondly, we build our gams. For this purpose we take the set of conditioning variables and create a uniform mesh between 0 and 1. The next step is to take the function f we want to define the dependency, and apply it to the mesh, to get $y = f(x)$. The last step in constructing the gam is to apply the `gam` command in R to this data set and specify the formula to generate it. In Figure 3 we can observe the fitted gam obtained for the case that we will study.

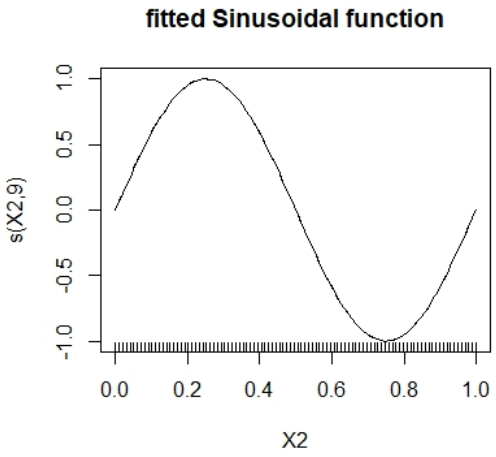


Figure 3: Fitted generalized additive model.

5 Simulations

Once the model is built we can proceed with actual simulations. Before that, we introduce a remark. We know that the higher the correlation in the first tree, the less impact the conditional probabilities of the higher trees have in the whole system[10]. When the first tree is highly correlated, the non-simplifying assumptions will be barely noticeable. Oppositely, if the first tree is highly uncorrelated, the copulas in the first tree will be close to the independent copula. Whereas, the copulas of the second tree will present irregular shapes, which will allow the identification of the non-simplified assumptions. We will mainly focus on a case in between the two extremes in Subsection 5.1 and Subsection 5.2. Finally, in Subsection 5.3 we briefly study the two extreme cases mentioned above.

5.1 Summary Correlated Gam Vine Copula.

The choice of parameters for our model is:

- Clayton Copula $\rightarrow \theta = 2$
- Gumbell Copula $\rightarrow \alpha = 1.5$

Notice that using (19) and (20), we obtain $\tau = 0.5$ and $\tau = 1/3$, respectively. This fact, places us in a case with correlations, but far from the highly correlated and highly uncorrelated cases. Once the model is built following the steps explained in Subsection 4.2, we can use the `Summary` function to obtain an overview of the model:

```
    GAM-Vine matrix:
     [,1] [,2] [,3]                  Where
[1,]    1    0    0                  1  <->  X1
[2,]    3    3    0                  2  <->  X2
[3,]    2    2    2                  3  <->  X3

 Tree 1:
X1,X2: Clayton type 1 (standard and 90 degrees rotated) with par=2 (tau=0.5)
X3,X2: Gumbel type 1 (standard and 90 degrees rotated) with par=1.5 (tau=0.33)

 Tree 2:
X1,X3|X2 : Gaussian copula with tau(z) = (exp(z)-1)/(exp(z)+1) where
Formula:
 z ~ s(X2, k = 10, bs = "cr")

 Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
```

[10] Dr. D. Kurowicka - Personal communication

```
(Intercept)   1.0000     0.1118   8.944 2.37e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Approximate significance of smooth terms:
      edf Ref.df     F  p-value
s(X2)   1      1 26.74 1.46e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
R-sq.(adj) =  0.0158   Deviance explained = 2.57%
GCV = 0.78691  Scale est. = 0.77117   n = 100
```

We also present in Figure 4 a visualising of this three-dimensional gam vine copula.

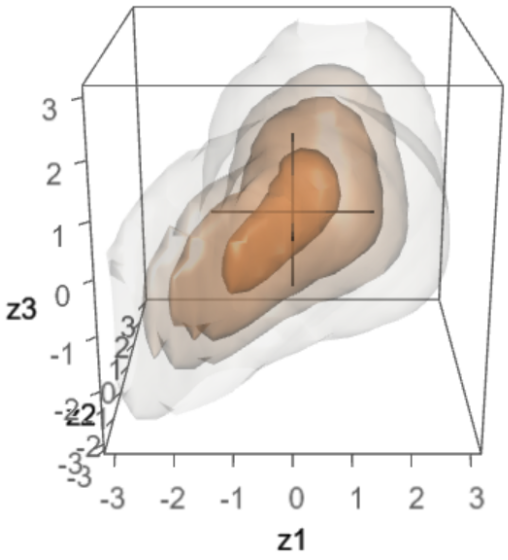


Figure 4: Visualising of our 3d gam Vine Copula, with normal margins.

This figure was obtained by using a web-application[11] implemented in [7]. This application and this paper are very useful in this context. The conclusion we draw from [7] is that non-simplified vine copulas exhibit quite irregular contour shapes, which cannot be captured using the simplified assumptions. In our case, looking at the irregular shape in Figure 4, we check the aforementioned statement.

The next step will be to simulate from this model. To do so, we use the command `gamVineSimulate`. We take $n = 1000$ and we use `set.seed(50)` for reproducibility of our results. Once the data has been simulated, we proceed to perform an analysis of it in Subsection 5.2.

[11] https://vinecopula.shinyapps.io/Vine3DPlot

5.2 Data Analysis

In this subsection we analyse the simulated data. First, in Subsubsection 5.2.1 we will perform some plots to obtain visual results of our data. Second, in Subsubsection 5.2.2 we will fit some models for our data. Finally, in Subsubsection 5.2.3 we will perform some goodness of fit test. The main objective is to see if simplified vine copulas can capture the underlying features of these data, or if non- simplified vine copulas are required.

5.2.1 Visual Analysis

We first perform a visual analysis of the data. To do so we use the `pairs.copuladata` command. In Figure 5 we can observe bivariate contour plots on the lower panel, scatter plots and correlations on the upper panel and histograms on the diagonal panel. Moreover, in Figure 6 we show the scatter plots with more clarity.

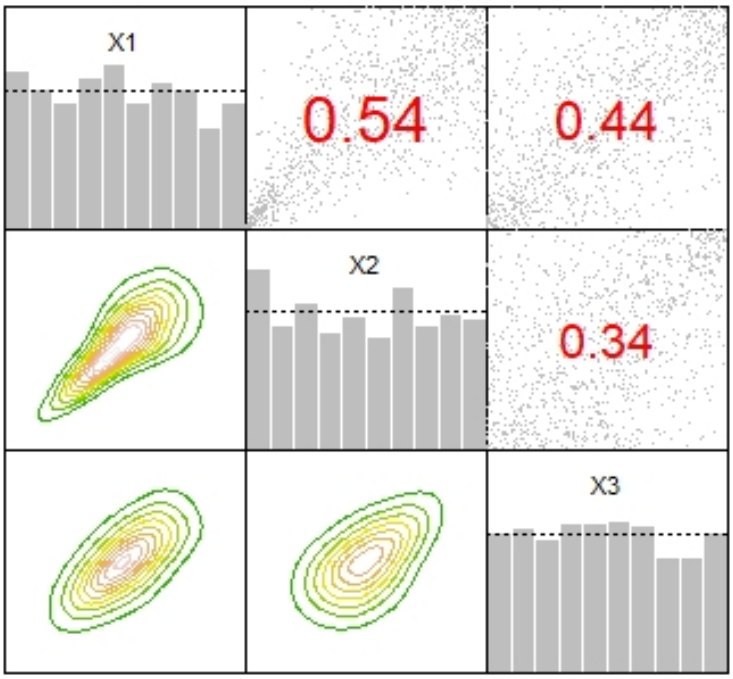
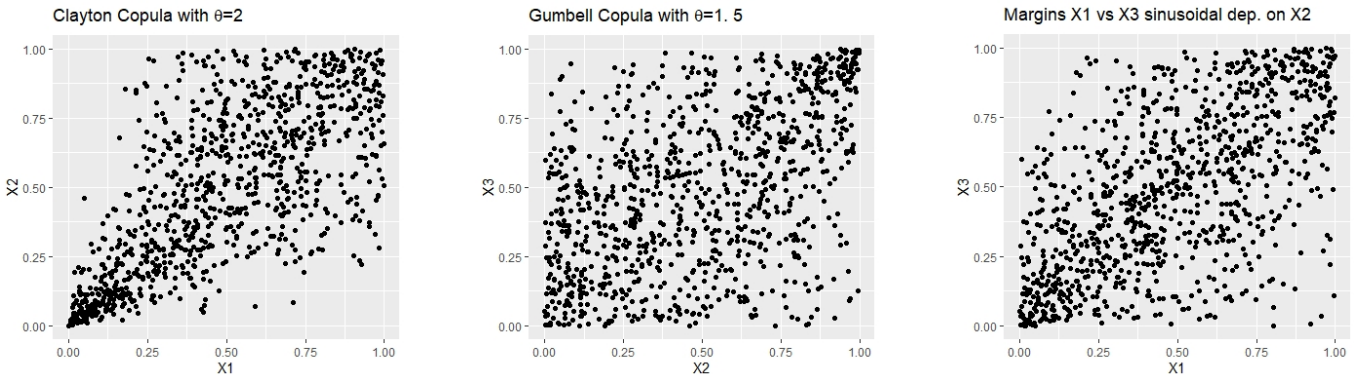


Figure 5: Bivariate contour plots, scatter plots, correlations and histograms of our simulated data.

Analyzing Figure 5, we can see how the histograms of X_1, X_2 and X_3 are almost uniform, as expected. Kendall's τ empirical values for the copulas in the 1^{st} tree are close to the true values $\tau = 0.5$ and $\tau = 1/3$. Regarding the contours, we do not see any irregular or unusual shapes. In addition, looking at Figure 6, there does not seem to be any indication of a non-simplified copula either.



(a) X_1 vs X_2 (b) X_2 vs X_3 (c) X_1 vs X_3

Figure 6: Scatter plot for our non-simplified copula: sinusoidal case.

These facts seem to contradict the statement made about Figure 4. However, in the 2 previous figures we showed the relation between X_1 and X_3 without taking into account the values of X_2. In order to study the copula $C_{1,3|2}$, we really have to study the relation between X_1 and X_3 conditioning on the value of X_2. The way of tackle this problem is to study the relation between the pseudo-observations $X_{1|2}$ and $X_{3|2}$. To construct them we have to use conditional copulas as described in (9). Applying the `BiCopHfunc1` command to our data and to the copulas $C_{1,2}$ and $C_{2,3}$, described in Subsection 4.1, we obtain the desired pseudo-observations. In Figure 7 we show the contour plots, scatter plots, histograms and correlations for our computed pseudo-observations.

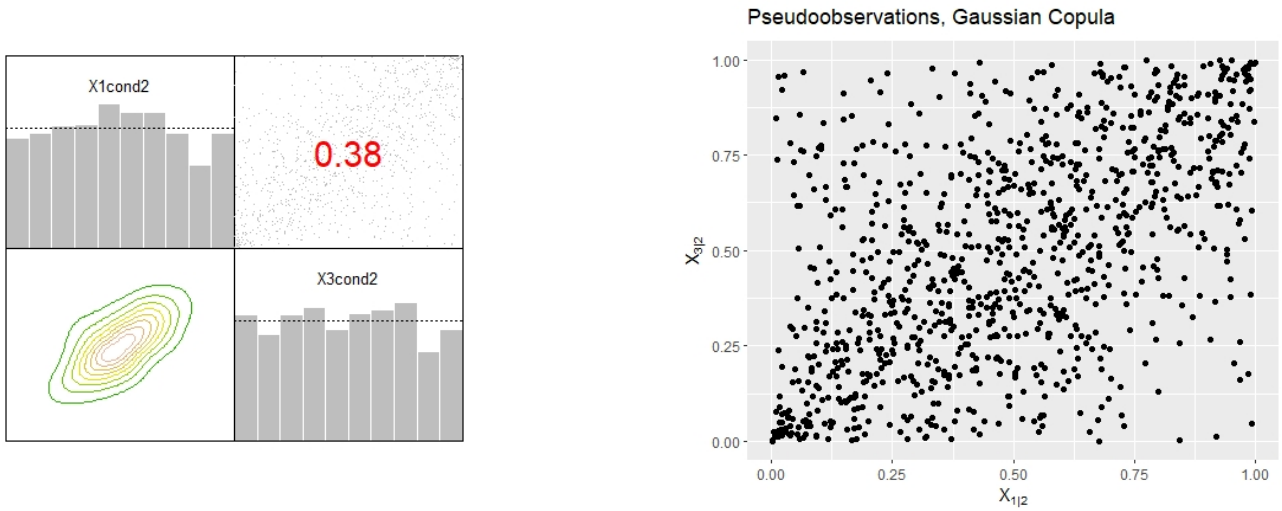


Figure 7: Contour plots, Scatter plots, Histograms and Correlations for the Pseudo-Observations.

In this case we can see how the contour has a slightly irregular shape. This fact gives us clues that working with non-simplified assumptions might be more adequate. On the other hand $\tau = 0.38$ and as we are dealing with Gaussian copula, using (21) we get $\rho = 0.562$. For this value, it is unlikely to get as many observations in the upper left quadrant $[0, 0.25] \times [0.75, 1]$ as we see in the scatter plot. Therefore, the need for non-simplified copulas is gradually becoming more significant.

5.2.2 Fitting Models

Next, we are going to fit several models with these data. These models will be presented from best to worst theoretical performance. They are mainly characterized by the amount of information that we have about the model that generated our data set. For the different models we will use the logLikelihood as a measure of performance.

To compute the logLikelihoods we use the command `gamVinePDF`, which allow us to evaluate some data on the probability density function of a given model. Moreover we use the `system.time` command to compute the computational times of the different fittings.

- **Known Model Generator:**

  For the model that we use in Subsection 5.1 to generate our data we obtain:

  $$\boxed{\text{logLik} = 1030.868}$$

- **Known Vine Structure + Family Copula:**

  If we knew the structure of the vine model and the families to which each of the edges belong, we could use the command `gamVineSeqFit` to estimate the parameters of the 1^{st} tree and the gam for the 2^{nd} tree. Applying this procedure, the fitted model we obtain is characterized by the following features:

  $$\boxed{\theta_{\text{Clayton}} = 2.16,\ \alpha_{\text{Gumbell}} = 1.51}$$

  $$\boxed{\text{Time elapsed} = 0.63}$$

  $$\boxed{\text{logLik} = 1031.948}$$

  It is not surprising that the loglikelihood for this model is higher, since we are calculating the model that best fits our data. We can also verify that the estimators that R uses for the copula parameters are consistent. Indeed, taking $n = 100000$, we can check after a long computation time that the parameter estimators coincide exactly with the generating parameters, that is: $\theta_{\text{Clayton}} = 2.00$, and $\theta_{\text{Gumbell}} = 1.50$.

- **Known Vine Structure:**

  If we only knew the structure of the vine model, we could use the command `gamVineCopSelect` to obtain families and their parameters that best fit our data. Applying this procedure, the fitted model we obtain is characterized by the following features:

```
     Tree 1:
X1,X2: Clayton type 1 (standard and 90 degr. rotated) with par=2.16 (tau=0.52)
X3,X2: Gumbel type 1 (standard and 90 degr. rotated) with par=1.51 (tau=0.34)
     Tree 2:
X1,X3|X2 : Gaussian copula with tau(z) = (exp(z)-1)/(exp(z)+1) where
Formula: z ~ s(X2, k = 10, bs = "cr")
```

$$\boxed{\theta_{\text{Clayton}} = 2.16,\ \alpha_{\text{Gumbell}} = 1.51}$$

$$\boxed{\text{Time elapsed} = 6.85}$$

$$\boxed{\text{logLik} = 1031.948}$$

The results obtained are the same as in the previous case, which shows the power of the estimation methods described in Subsection 3.3 On the other hand, the computation time is longer, as expected, since the programme has to check the fitting of other families of copulas such as the t-copula.

- **No Previous Information:**

  If we knew nothing about the model: neither the type of structure nor the families involved, we could use the command `gamVineStructureSelect` to obtain the fitted model. Applying this procedure, the fitted model we obtain is characterized by the following features:

```
    Tree 1:
X2,X1: Clayton type 1 (standard and 90 degr. rotated) with par=2.16 (tau=0.52)
X1,X3: t with par=0.62 and par2=10.3 (tau=0.42)
    Tree 2:
X2,X3|X1 : Clayton type 4 (survival and 270 degr. rotated) copula
with tau(z) = (exp(z)-1)/(exp(z)+1) where
Formula: z ~ s(X1, k = 20, bs = "cr")
```

$$\boxed{\text{Time elapsed} = 10.70}$$

$$\boxed{\text{logLik} = 864.3057}$$

In this case we can see how the performance of this model worsens notably with respect to the 2 previous cases. These results were to be expected, since, as we commented in Subsection 3.4, the process to choose the Vine structure is heuristic and not optimal. Moreover, we can plot the gam obtained for this fitting.

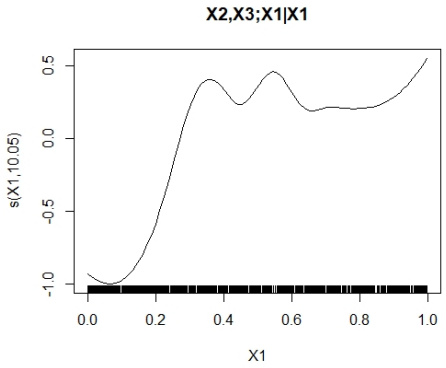


Figure 8: Gam for the fitted Gam Vine with no previous information.

We realize after fitting this model how important it is to have some kind of information[12] about the data you we trying to fit.

[12]Even knowing only the structure, the results obtained are very good.

Finally, we will fit 2 more models, although in these cases we will relax the non-simplified assumptions of copulas.

- **Known Vine Structure:**

  If we only knew the structure of the simplified vine copula model, we could use the command `RVineCopSelect` to obtain families and their parameters that best fit our data. Applying this procedure and restricting to the copula families studied during the course, the fitted model we obtain is characterized by the following features:

```
tree   edge | family  cop   par  par2 |  tau   utd   ltd
-----------------------------------------------------
   1    2,1 |      3    C  2.16  0.00 | 0.52     -  0.73
        2,3 |      4    G  1.51  0.00 | 0.34  0.42     -
   2  3,1;2 |      2    t  0.57  4.16 | 0.39  0.29  0.29
---
type: C-vine    logLik: 884.7    AIC: -1761.39    BIC: -1741.76
---
1 <-> X1,   2 <-> X2,   3 <-> X3
```

  Time elapsed = 3.14

  In this case we can check the power of the `VineCopula` package and the flexibility of the simplified vine copulas. Since we obtain slightly worse performance than the previous model, but less computational time.

- **No previous information**

  Lastly, if we do not even know the structure of the Vine we can use the `RVineStructureSelect` command. The results that we obtain are:

```
tree   edge | family  cop    par   par2 |   tau   utd   ltd
-----------------------------------------------------------
   1    1,2 |      3    C   2.16   0.00 |  0.52     -  0.73
        3,1 |      2    t   0.62  10.30 |  0.42  0.13  0.13
   2  3,2;1 |      2    t  -0.04   4.61 | -0.02  0.05  0.05
---
type: C-vine    logLik: 768.6    AIC: -1527.2    BIC: -1502.66
```

  Time elapsed = 1.89

As in the previous case, we can see how the simple fact of knowing the structure, means a great improvement of the fitting. This last model is the one with the least logLikelihood and is the worst of all the discussed models. The only good thing about this model is that is computationally fast. But what good is a model that is fast if it is very inaccurate?

In the last 2 simplified fittings, the AIC of the models is shown. On the other hand, we do not have access to the AIC for the gam Vine models because we are working with non parametric models. This statistic would be useful to choose the most appropriate model. Since, although the non simplified models have greater logLikelihoods, they are more computationally expensive.

In our 3-dimensional case, the computation time required by the gam Vine is affordable. Looking at Table 1, there are only 3 different Vine Structures. Checking these 3 different models would not be very expensive. In return, we get a model that captures the data generating model almost perfectly.

The big difference between logLikelihoods:

$$\boxed{\text{Non-simplified Assumpt.} \rightarrow logLik = 1031.948 \text{ vs Simplified Assumpt.} \rightarrow logLik = 884.7}$$

together with the results shown in Subsubsection 5.2.1, presage the need for non-simplified copulas in this case. To check if we really need non-simplified copula, we will perform goodness of fit test in Subsubsection 5.2.3.

5.2.3 Goodness of fit test

To finish the analysis of the simulated data, we will perform some goodness of fit test. The main objective is to check if the non simplified assumptions are needed. Moreover we will also study the differences between the simplified and non-simplified fitted models, when only the vine structure is known. To tackle the main objective we will use the Constant Conditional Correlation test, developed by M. Kurz and F. Spanhel [16] and implemented in the library `pacostest` [12]. To study the differences, we will use visual tests using different plots. There are other types of tests for non-simplified assumptions based on distance measurement, although we will not discuss them in this project.

- **Constant Conditional Correlation Test**

This test is based on the application of regression trees to the pseudo-observations of the best simplified vine model. The idea is to partition the conditioning space onto disjoint regions and compute the correlation of the data on these regions. For carrying out the partitions we use the recursive binary splitting[13]. Moreover, these partitions are optimal in the sense that they maximize the difference between the correlations of both partitioned spaces.

In this test the null hypothesis is given by:

$$H_0 : (\mathbf{u}_{j|D_e}, \mathbf{u}_{k|D_e}) \perp \mathbf{u}_{D_e}.$$

The test statistic T is based on the difference between the correlations in the different nodes of the tree. The critical region is of the form $C = \{T > \beta\}$. So when there are large differences between correlations, we reject the null hypothesis, and therefore the non-simplified assumptions are needed.

[13] splittings of the form $\{x|x_j < s\}$ and $\{x|x_j \geq s\}$.

To carry out this test we need the `pacotestRvineSeq` command. It is also useful to use the `pacotestset` command to show some characteristics of our data, such as scatter plots of the nodes, or showing the structure of the regression tree. Applying the aforementioned commands to our data we obtain:

```
    $pacotestResultLists          $pValues
     [,1]   [,2] [,3]              [,1]  [,2] [,3]
[1,] NULL   NULL NULL        [1,]   NA   NA   NA
[2,] List,3 NULL NULL        [2,]    0   NA   NA
[3,] NULL   NULL NULL        [3,]   NA   NA   NA

$testResultSummary
  Tree NumbOfRejections IndividualTestLevel
1    2                1                0.05

Interpretation
1 1 rejections at a individual test level of  0.05 -->
Sequential test procedure can be stopped due to a rejection
```

As mentioned before, we also include in Figure 9 and in Figure 10, the Regression Tree structure and the Scatter plots for the nodes of this tree. Both pictures are quite informative and help us to grasp what is happening.

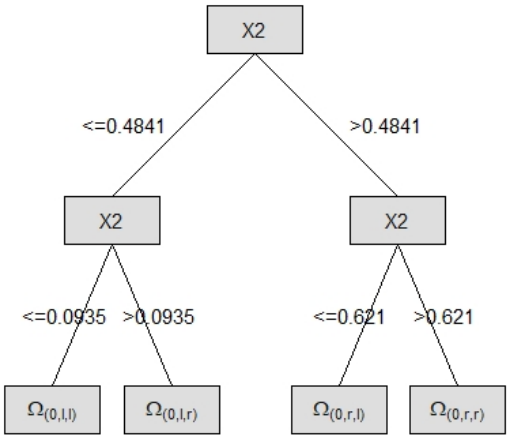
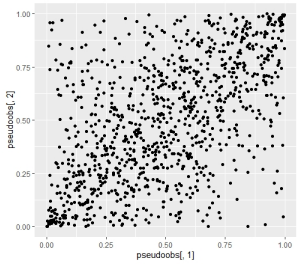
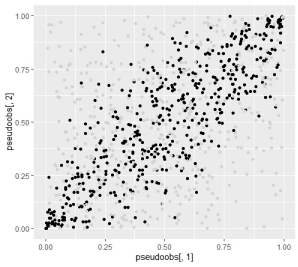


Figure 9: Structure of the regression tree for the CCC test for non-simplifying assumptions.

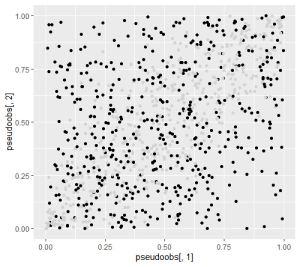
Figure 10: Scatter plots for the nodes of the Regression Tree for the CCC test for non-simplifying assumptions.



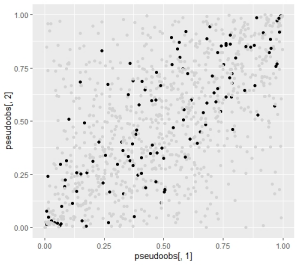
(a) $\rho = 0.536$, $\tau = 0.38$



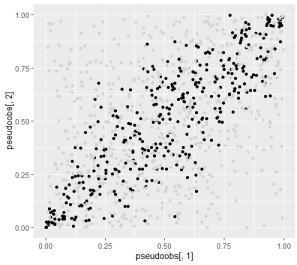
(b) $X_2 \leq 0.4821$
$\rho = 0.828$, $\tau = 0.621$.



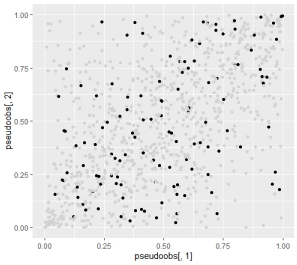
(c) $X_2 > 0.4821$
$\rho = 0.239$, $\tau = 0.154$



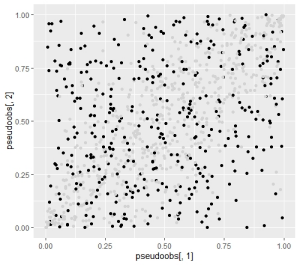
(d) $X_2 \leq 0.0935$
$\rho = 0.764$, $\tau = 0.553$



(e) $X_2 > 0.0935$
$\rho = 0.851$, $\tau = 0.648$



(f) $X_2 \leq 0.621$
$\rho = 0.437$, $\tau = 0.288$



(g) $X_2 > 0.621$
$\rho = 0.179$ $\tau = 0.114$

From Figure 10 we can see a clear difference between both the scatter plots and the correlation values for the nodes e) and g). This fact makes us reject the null hypothesis and confirms the hints that we had in Subsubsection 5.2.1 and in Subsubsection 5.2.2. We therefore conclude that we need the non-simplified assumptions to capture the existing dependencies in our simulated data.

We could also have performed this test on our own. Indeed, in Figure 11 we show the estimated value of Kendall's tau estimator depending on the values of X_2. From this plot, we can check in a practical way the sinusoidal dependence that we introduced in Subsection 4.1

It is worth to mention the power of the Constant Conditional Correlation test, and the great applications that it has. It is also efficient computationally, and it has been proven to work on real data cases of dimension up to n=49 [16].

During our implementation we also used the Probability Integral Transformation + Kolmogorov Smirnoff test, applied on our data and on the best fitted simplified model. The results we obtained were that the vast majority of the time the null hypothesis of simplified assumptions is not rejected.

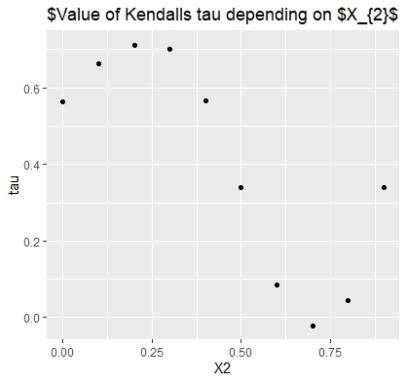


Figure 11: Estimated value of Kendall's tau estimator depending on the values of X_2.

Therefore, this test is not entirely suitable for attempting to test the non-simplifying assumptions.

- **Visual Plots**

To conclude this section we will simulate data from the Known Structure fitted models, both for non-simplified and simplified assumptions. With these data, we perform a brief visual analysis to check the differences between these models. As we saw in Subsubsection 5.2.1, the most informative figures are the contour and scatter plots of the pseudo-observations. Hence, these will be the ones that interest us most.

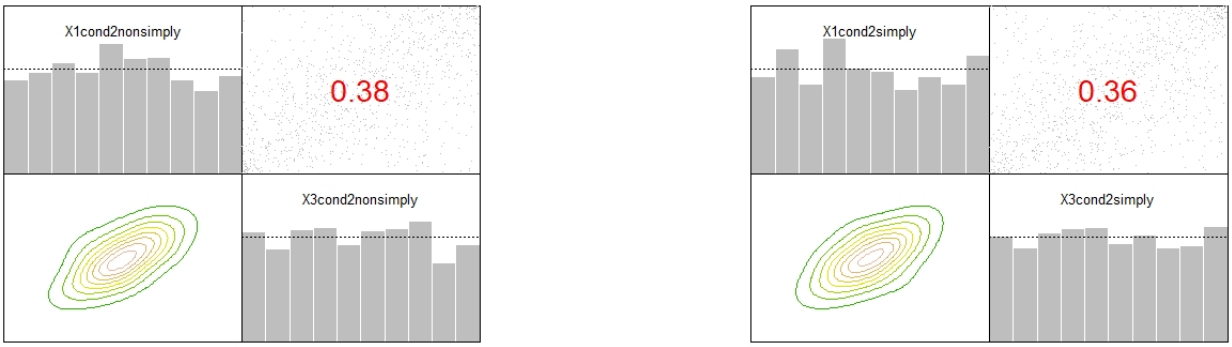


Figure 12: Contour plots, Scatter plots, Histograms and Correlations for the Pseudo-Observations of: the non-simplified fitted model on the left, the simplified fitted model on the right.

We can see how the contour on the left is a little more irregular than the one on the right. Moreover, comparing Figure 7 and Figure 12 we observe that the contour for the non-simplified case is more similar to the contour of the data simulated by our initial model. This fact once again highlights the need for non-simplified assumptions.

5.3 Extreme Cases

To conclude this project, we will study the highly uncorrelated and the highly correlated cases very briefly. For this purpose, we study 2 models with the same structure as the one described in Section 4, but we choose different values of the parameters. The main objective is to simulate from both cases and fit models for the simulated data. In this section we will only show 2 fitted models per each case. For both models we will assume that the vine structure is known. In the first one, assuming non-simplified assumptions, we will use the `gamVineCopSelect` command. For the second one, we will assume that the simplified assumptions hold, so we will use the `RVineCopSelect` command. The purpose is mainly to compare the relative error for the obtained logLikelihoods.

- **Highly Uncorrelated:** The choice of parameters for our model is:
  - Clayton Copula $\rightarrow \theta = 0.25 \Rightarrow \tau = 0.111$
  - Gumbell Copula $\rightarrow \alpha = 1.1 \Rightarrow \tau = 0.0909$

  The logLikelihood for the simulated data is $\Rightarrow \boxed{\text{logLik} = 409.7593}$

  1. **Non-simplifying assumptions**

```
 Tree 1:
X1,X2: Clayton type 1 (standard and 90 degrees rotated) par=0.26 (tau=0.11)
X3,X2: Gumbel type 1 (standard and 90 degrees rotated) par=1.08 (tau=0.07)
 Tree 2:
X1,X3|X2 : Gaussian copula with tau(z) = (exp(z)-1)/(exp(z)+1) where
Formula:
 z ~ s(X2, k = 10, bs = "cr")
```

$$\boxed{\text{logLik} = 415.9389}$$

$$\boxed{\epsilon_r(\%) = \frac{|415.9389 - 409.7593|}{409.7593} \times 100 = 1.508\%}$$

  2. **Simplifying assumptions**

```
tree   edge | family  cop   par  par2 |  tau   utd   ltd
-------------------------------------------------------
   1    2,1 |      3    C  0.26  0.00 | 0.11     -  0.07
        2,3 |      4    G  1.08  0.00 | 0.07  0.10     -
   2  3,1;2 |      2    t  0.58  3.76 | 0.39  0.31  0.31
---
type: C-vine    logLik: 257.21    AIC: -506.42    BIC: -486.79
```

$$\boxed{\epsilon_r(\%) = \frac{|257.21 - 409.7593|}{409.7593} \times 100 = 37.229\%}$$

- **Highly Correlated:** The choice of the parameters for our model is:
  - Clayton Copula $\rightarrow \theta = 8 \Rightarrow \tau = 0.8$
  - Gumbell Copula $\rightarrow \alpha = 10 \Rightarrow \tau = 0.9$

  The logLikelihood for the simulated data is $\Rightarrow \boxed{\text{logLik} = 3595.546}$

  1. **Non-simplifying assumptions**

```
  Tree 1:
X1,X2: Clayton type 1 (standard and 90 degrees rotated) with par=7.91 (tau=0.8)
X3,X2: Gumbel type 1 (standard and 90 degrees rotated) with par=9.82 (tau=0.9)

  Tree 2:
X1,X3|X2 : Gaussian copula with tau(z) = (exp(z)-1)/(exp(z)+1) where
Formula:
 z ~ s(X2, k = 10, bs = "cr")
```

$$\boxed{\text{logLik} = 3599.152}$$

$$\boxed{\epsilon_r(\%) = \frac{|3595.546 - 3599.152|}{3595.546} \times 100 = 0.1002\%}$$

  2. **Simplifying assumptions**

```
tree   edge | family  cop   par  par2 |  tau   utd   ltd
-------------------------------------------------------
   1    2,1 |      3    C  7.91  0.00 | 0.80     -  0.92
        2,3 |      4    G  9.82  0.00 | 0.90  0.93     -
   2  3,1;2 |      2    t  0.61  3.39 | 0.42  0.36  0.36
---
type: C-vine    logLik: 3434.3    AIC: -6860.6    BIC: -6840.97
```

$$\boxed{\epsilon_r(\%) = \frac{|3595.546 - 3434.3|}{3595.546} \times 100 = 4.4846\%}$$

In the highly uncorrelated case we can see how important it is to take into account the non-simplifying assumptions. Since when we work with simplified assumptions the relative errors committed are higher than 30%. Simplified models are not a good choice in this case. On the contrary, in the highly correlated case, we see that the relative error for the simplified model is less than 5%. So working with the simplified models would be a good option, taking into account that the computation times are lower.

These facts highlight what was mentioned at the beginning of Section 5. The higher the correlation in the first tree, the less impact the conditional probabilities of the higher trees have in the whole system, and therefore the less important is to consider the non-simplifying assumptions.

6 Conclusion

In this project we have studied non-simplified copulas using generalized additive models. The main objective was to develop this type of copulas, to explore their applicability and their differences with respect to the simplified ones. For this purpose, we implemented a 3-dimensional non-simplified vine.

First, we simulated data from this model in order to study it visually. The most striking results were that the 3d-contours of our model had very irregular shapes, far from those studied in class with the simplified assumptions. In the same line, it was interesting to compute the pseudo observations using the conditional copulas of the first tree. Plotting these pseudo-observations we also observed 2d-contour irregular shapes. Regarding the scatter plot we noticed that although there was some correlation between the variables, we obtained many outliers.

Secondly, we fitted different models for our simulated data. We studied 5 different models, depending on the amount of information available. These models are summarized in the following table:

Table 2: Fitted Models intermediate case.

Information	Simplified	logLik	$C_{1,2}$	θ	$C_{2,3}$	α	$C_{2,3\|1}$	Time
Study model	Not	1030.8	Clayton	2	Gumbell	1.5	Gaussian	·
Vine Struct+Copulas	Not	1031.9	Clayton	2.16	Gumbell	1.51	Gaussian	0.63
Vine Structure	Not	1031.9	Clayton	2.16	Gumbell	1.51	Gaussian	6.85
No previous info.	Not	864.3	Clayton	2.16	-	-	-	10.70
Vine Structure	Yes	884.7	Clayton	2.16	Gumbell	1.51	Student t	3.14
No previous info	Yes	768.6	Clayton	2.16	-	-	-	1.89

We found that the loglikelihoods of the non-simplified models were higher than those of the simplified models. In reality, we do not usually have information about the structure or the copula families. For dimension 3, there were only 3 different structures. Considering the computational times showed in the previous table, fitting these 3 models would not be computationally expensive, and in return we got a model that captures the data generating model almost perfectly.

Thirdly, we applied the Constant Conditional Correlation Test to our data. As expected, this test was able to detect the need for not-simplified assumptions. We went deeper and we checked how Kendall's τ was varying for different subdomains. We also checked the visual differences between models with simplified and non-simplified assumptions.

The main conclusion that we draw from this project is that for 3-dimensional models, non-simplified vine copulas exhibit some special characteristics, which sometimes cannot be captured using the simplified assumptions. We saw that when dealing with highly uncorrelated copulas in the first tree, an uncritical use of the simplifying hypothesis can be misleading. Therefore, it is highly recommended to perform an analysis of these assumptions, such as the one performed in this work, before working directly with simplified assumptions. All these statements correspond to the case of dimension 3. In the case of higher dimensions, the computation times become so long, that simplified assumptions make more sense. For these high dimensional cases, simplified vine copulas capture in general terms the main features of the data and provide a more smooth fit.

References

[1] Abe Sklar. "Fonctions de répartition à n dimensions et leurs marges". In: *Publications de l'Institut de Statistique de l'Université de Paris* 8 (1959), pp. 229–231.

[2] D. Kurowicka and R.M. Cooke. *Uncertainty Analysis with High Dimensional Dependence Modelling*. Wiley Series in Probability and Statistics. Wiley, 2006. ISBN: 9780470863060. URL: https://books.google.nl/books?id=G4tQAAAAMAAJ.

[3] Tim Bedford and Roger M. Cooke. "Vines–a new graphical model for dependent random variables". In: *The Annals of Statistics* 30.4 (2002), pp. 1031–1068. DOI: 10.1214/aos/1031689016. URL: https://doi.org/10.1214/aos/1031689016.

[4] Ingrid Hobæk Haff, Kjersti Aas, and Arnoldo Frigessi. "On the simplified pair-copula construction — Simply useful or too simplistic?" In: *Journal of Multivariate Analysis* 101.5 (2010), pp. 1296–1310. ISSN: 0047-259X. DOI: https://doi.org/10.1016/j.jmva.2009.12.001. URL: https://www.sciencedirect.com/science/article/pii/S0047259X09002206.

[5] Fabian Spanhel and Malte S. Kurz. "Simplified vine copula models: Approximations based on the simplifying assumption". In: *Electronic Journal of Statistics* 13.1 (2019), pp. 1254–1291. DOI: 10.1214/19-EJS1547. URL: https://doi.org/10.1214/19-EJS1547.

[6] Elif F Acar, Christian Genest, and Johanna Nešlehová. "Beyond simplified pair-copula constructions". eng. In: *Journal of multivariate analysis* 110 (2012), pp. 74–90. ISSN: 0047-259X.

[7] Matthias Killiches, Daniel Kraus, and Claudia Czado. *Examination and visualisation of the simplifying assumption for vine copulas in three dimensions*. 2016. arXiv: 1602.05795 [stat.AP].

[8] Thibault Vatter and Valérie Chavez-Demoulin. "Generalized additive models for conditional dependence structures". In: *Journal of Multivariate Analysis* 141 (2015), pp. 147–167. ISSN: 0047-259X. DOI: https://doi.org/10.1016/j.jmva.2015.07.003. URL: https://www.sciencedirect.com/science/article/pii/S0047259X15001633.

[9] Thibault Vatter and Thomas Nagler. "Generalized Additive Models for Pair-Copula Constructions". In: *Journal of Computational and Graphical Statistics* 27.4 (2018), pp. 715–727. DOI: 10.1080/10618600.2018.1451338. eprint: https://doi.org/10.1080/10618600.2018.1451338. URL: https://doi.org/10.1080/10618600.2018.1451338.

[10] Thomas Nagler and Thibault Vatter. *gamCopula: Generalized Additive Models for Bivariate Conditional Dependence Structures and Vine Copulas*. R package version 0.0-7. 2020. URL: https://CRAN.R-project.org/package=gamCopula.

[11] Thomas Nagler et al. *VineCopula: Statistical Inference of Vine Copulas*. R package version 2.4.1. 2020. URL: https://CRAN.R-project.org/package=VineCopula.

[12] Malte S. Kurz. *pacotest: Testing for Partial Copulas and the Simplifying Assumption in Vine Copulas*. R package version 0.4.0. 2020.

[13] S.N Wood. *Generalized Additive Models: An Introduction with R*. 2nd ed. Chapman and Hall/CRC, 2017.

[14] Trevor Hastie and Robert Tibshirani. "Generalized Additive Models". In: *Statist. Sci.* 1.3 (Aug. 1986), pp. 297–310. DOI: 10.1214/ss/1177013604. URL: https://doi.org/10.1214/ss/1177013604.

[15] J. Dißmann et al. "Selecting and estimating regular vine copulae and application to financial returns". In: *Computational Statistics Data Analysis* 59 (2013), pp. 52–69. ISSN: 0167-9473. DOI: https://doi.org/10.1016/j.csda.2012.08.010. URL: https://www.sciencedirect.com/science/article/pii/S0167947312003131.

[16] Malte S. Kurz and Fabian Spanhel. *Testing the simplifying assumption in high-dimensional vine copulas*. 2021. arXiv: 1706.02338 [stat.ME].