



Technische Universität München  
Zentrum Mathematik  
Lehrstuhl für Mathematische Statistik

# Model distances, block maxima and repeated measurements in the context of vine copulas

Matthias Markus Killiches

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende: Prof. Dr. Christina Kuttler  
Prüfer der Dissertation: 1. Prof. Claudia Czado, Ph.D.  
2. Prof. Dr. Paul Janssen  
(Universität Hasselt, Belgien)

Die Dissertation wurde am 29.05.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 14.08.2017 angenommen.



## Zusammenfassung

Vine Copulas sind Abhängigkeitsmodelle, die multivariate Copuladichten als Produkt über bivariate Bausteine, so genannte Paar-Copulas, darstellen. Ihre größten Vorteile sind ihre Flexibilität und Anwendbarkeit, selbst in hohen Dimensionen. Diese Arbeit erweitert die vorhandene Literatur zu Vine Copulas um drei neuartige Aspekte. Klassische Modell-distanzen wie die Kullback-Leibler-Distanz können nur in niedrigen Dimensionen verwendet werden, da sie die Berechnung mehrdimensionaler Integrale voraussetzen. Um die Berechnung von Abständen zwischen hochdimensionalen Vine Copulas zu ermöglichen, entwickeln wir Modelldistanzen für Vine Copulas, die auf der Kullback-Leibler-Distanz basieren. Die Tauglichkeit unseres Ansatzes wird in numerischen Beispielen und Simulationsstudien belegt und die vorgeschlagenen Methoden werden zur Modellwahl verwendet. Block Maxima werden üblicherweise benutzt, um in der Extremwerttheorie Resultate herzuleiten, wenn die Blockgröße gegen unendlich geht. In der Praxis sind Blockgrößen allerdings stets endlich. Deshalb liefern wir eine explizite Formel für die Copuladichte der komponentenweisen endlichen Block Maxima von multivariaten Verteilungen. Das Ergebnis wird auf Vine Copulas in numerischen Beispielen und einer Datenanalyse angewendet. Schließlich wird ein flexibles D-Vine-Copula-basiertes Modell für unbalancierte Longitudinaldaten entwickelt. Wir präsentieren eine sequenzielle Schätzmethode und passen das Bayessche Informationskriterium für unsere Situation an. Außerdem zeigen wir, dass unser Ansatz als Erweiterung einer großen Klasse von linearen gemischten Modellen interpretiert werden kann. In einer Datenanalyse werden die beiden Modellklassen insbesondere bezüglich ihrer Fähigkeit, bedingte Quantile für zukünftige Messungen zu schätzen, verglichen.



## Abstract

Vine copulas are dependence models that represent multivariate copula densities as products over bivariate building blocks, so-called pair-copulas. Their main advantages are flexibility and applicability even in high dimensions. This thesis extends the existing literature on vine copulas by three novel aspects. Classical model distances such as the Kullback–Leibler distance are limited to low dimensions since they require multivariate integration. In order to facilitate the determination of distances between high-dimensional vine copulas we develop model distances for vine copulas based on the Kullback–Leibler distance. The validity of our approach is verified in numerical examples and simulation studies and the proposed methods are applied for model selection. Block maxima are usually used to derive results in extreme-value theory as the block size goes to infinity. In practice, however, block sizes are always finite. Therefore, we provide an explicit formula for the copula density of the componentwise finite block maxima for multivariate distributions. The result is applied to vine copulas in numerical examples and a data application. Finally, a flexible D-vine copula based model is developed for unbalanced longitudinal data. We present a sequential estimation method for the model and adjust the Bayesian information criterion to our situation. Further, we show that our approach can be interpreted to be an extension of a wide class of linear mixed models. In a data application the two model classes are compared, in particular regarding their ability to predict conditional quantiles for future measurements.



## **Acknowledgements**

Firstly, I want to thank my advisor Prof. Claudia Czado, Ph.D., who gave me the opportunity to write this thesis and has been guiding me in my research for the last three years. Her supervision and support were extremely helpful and her advice was always constructive. Especially, I appreciate that she has constantly been available for fruitful discussions and encouraged me to attend several international conferences.

I would also like to thank Prof. Dr. Paul Janssen for acting as a referee of this thesis.

I am happy to thank my colleagues from Technische Universität München for many stimulating discussions and the enjoyable time we spent together. It is a particular pleasure for me to express my gratitude to my colleague and friend Daniel Kraus for excellent collaboration in three joint research projects and particularly for twenty-two years of invaluable friendship. Without him the way from elementary school to this thesis would definitely have been much harder and far less joyful. Further, I would like to thank my dear friends from swimming for making life fun—inside and outside the pool.

Financial support through a research stipend from Technische Universität München, financed by Allianz Deutschland AG, is gratefully acknowledged.

Last, and most importantly, I would like to sincerely thank my parents, Sabine and Manfred Killiches, for their love, support and encouragement.



---

# *Contents*

---

<b>Zusammenfassung</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Vine copula models</b>	<b>7</b>
2.1. Copulas . . . . .	7
2.2. Vine copulas . . . . .	8
2.2.1. Tree representation . . . . .	8
2.2.2. Matrix representation . . . . .	10
2.2.3. Simplifying assumption . . . . .	12
2.2.4. Notation of simplified parametric vine copulas . . . . .	12
2.2.5. Simulation and fitting of vine copulas . . . . .	13
<b>3. Model distances for vine copulas</b>	<b>15</b>
3.1. Introduction . . . . .	15
3.2. Model distances for vines . . . . .	17
3.2.1. Kullback–Leibler distance . . . . .	17
3.2.2. Approximate Kullback–Leibler distance . . . . .	23
3.2.3. Diagonal Kullback–Leibler distance . . . . .	27
3.2.4. Single diagonal Kullback–Leibler distance . . . . .	33
3.3. Comparison of all introduced model distances . . . . .	35
3.3.1. Comparison of all introduced KL approximations . . . . .	35
3.3.2. Comparison of the resulting JD approximations . . . . .	38
3.3.3. Calibration . . . . .	38
3.4. Hypothesis test for model selection . . . . .	40

3.5.	Model selection . . . . .	42
3.5.1.	KL based model selection . . . . .	42
3.5.2.	Five-dimensional mixed vine . . . . .	43
3.5.3.	20-dimensional t vine . . . . .	44
3.6.	Determination of the optimal truncation level . . . . .	46
3.6.1.	Algorithms for the determination of optimal truncation levels . . . . .	46
3.6.2.	Simulation study . . . . .	48
3.6.3.	Real data examples . . . . .	51
3.7.	Conclusion . . . . .	53
<b>4.</b>	<b>Block maxima for vine copulas</b>	<b>55</b>
4.1.	Introduction . . . . .	55
4.2.	Copula density of the distribution of block maxima . . . . .	56
4.3.	Application to three-dimensional vine copulas . . . . .	59
4.4.	Copula density of scaled block maxima . . . . .	67
4.5.	Application to scaled three-dimensional vine copulas . . . . .	70
4.6.	Conclusion . . . . .	74
<b>5.</b>	<b>Modeling repeated measurements using D-vine copulas</b>	<b>77</b>
5.1.	Introduction . . . . .	77
5.2.	D-vine based repeated measurement model . . . . .	79
5.2.1.	Setting . . . . .	79
5.2.2.	D-vine based dependence model . . . . .	80
5.3.	Connection between the D-vine based model and linear mixed models . . . . .	86
5.3.1.	Linear mixed models for repeated measurements . . . . .	86
5.3.2.	Aligning linear mixed models and the D-vine based approach . . . . .	87
5.4.	Estimation methods for the D-vine based model . . . . .	89
5.4.1.	Marginal modeling . . . . .	89
5.4.2.	Dependence modeling . . . . .	90
5.4.3.	Model selection . . . . .	92
5.5.	Simulation study . . . . .	93
5.6.	Application . . . . .	95
5.7.	Conclusion and outlook . . . . .	102
<b>A.</b>	<b>Appendix to Chapter 3</b>	<b>103</b>
A.1.	Proof of Proposition 3.6 . . . . .	103
A.2.	Regarding Remark 3.11 . . . . .	104
A.2.1.	Limit of the dKL . . . . .	104
A.2.2.	Tail transformation . . . . .	105

A.3. Finding the diagonal with the highest weight . . . . .	106
A.3.1. Procedure 1: Finding a starting value . . . . .	106
A.3.2. Procedure 2: Local search for better candidates . . . . .	107
<b>B. Appendix to Chapter 4</b>	<b>109</b>
B.1. Proof of Theorem 4.1 . . . . .	109
B.2. Proof of Proposition 4.4 . . . . .	112
<b>C. Appendix to Chapter 5</b>	<b>115</b>
C.1. Proof of Proposition 5.1 . . . . .	115
<b>Bibliography</b>	<b>119</b>



# Chapter 1

---

## *Introduction*

---

*Essentially, all models are wrong, but some are useful.*<sup>1</sup>

George E. P. Box (1919–2013)

The financial crisis of 2007–2009 showed that some of the models that were commonly used by financial institutions were clearly wrong—and no longer useful. In his famous article Felix Salmon blamed the Gaussian copula to have been “the formula that killed Wall Street” (Salmon, 2009). Although this formulation may have been a bit bold and a wrong model was certainly not the single trigger for the financial crisis, the basic criticism in fact had its justification: the Gaussian copula was broadly used in banking to describe the interdependencies between times-to-default of different financial entities, while major weaknesses of this relatively simple model were widely ignored. This fairly imprudent approach eventually proved to be fatal.

The foundations for copula modeling were already provided in the seminal work of Sklar (1959) at the end of the 1950s. Sklar had shown that multivariate distributions could be decomposed into marginal distributions and the dependence structure. The function describing the dependence structure is the copula. Yet, there was rather little interest in copulas at the beginning since computational capacities did not allow for practical applications. Around the turn of the millennium, however, the rise of copulas began (among others Joe, 1997; Embrechts et al., 1999, 2002). The previously mentioned Gaussian copula is the dependence structure underlying a multivariate normal distribution. Although

---

<sup>1</sup>Page 424 of Box, G. E. P., and Draper, N. R., (1987), *Empirical Model Building and Response Surfaces*, John Wiley & Sons, New York, NY.

it inherits many convenient properties from the normal distribution, it also has some drawbacks such as the lack of tail dependence when it comes to describing real world data. Fortunately, for modeling the dependence between two random variables a vast number of alternative flexible bivariate copula families has been developed (see for example Joe, 1997; Nelsen, 2006). However, classical multivariate dependence models such as Archimedean copulas usually lack flexibility and become numerically challenging in high dimensions (Hofert et al., 2012), which makes them unattractive for practical applications.

An alternative approach are vine copulas, which are also known as pair-copula constructions. The underlying idea of these models is to construct the density of a multivariate copula by bivariate building blocks. This way the complex problem of specifying a  $d$ -dimensional model is transformed to specifying  $d(d - 1)/2$  bivariate models, so-called pair-copulas. Since each pair-copula can be chosen arbitrarily this construction allows for great flexibility. The first vine copula model was proposed by Joe (1996) (at that time, however, not under the name ‘vine copula’). Bedford and Cooke (2002) introduced vines as a graph theoretical model. Vines describe the structure underlying a vine copula that determines which bivariate pair-copulas are to be specified. The fact that for a  $d$ -dimensional copula there are  $\frac{d!}{2} \cdot 2^{\binom{d-2}{2}}$  valid vine structures available (Morales-Nápoles, 2011) adds even more flexibility. However, it was not before the seminal work of Aas et al. (2009) that the popularity of vine copulas gained momentum. They developed statistical inference for vine copulas making this model class attractive to users from all fields of applications. Since then literature on vines has—figuratively speaking—exploded. There has been extensive research on various aspects of this model class. Dißmann et al. (2013) developed a sequential fitting algorithm for vine copulas. Panagiotelis et al. (2012) and Panagiotelis et al. (2017) investigated vine copulas with both continuous and discrete margins. Bayesian methods for vine copulas were for example developed in Min and Czado (2010), Czado and Min (2011) and Gruber and Czado (2015). Non-parametric estimation of vine copulas was performed by Nagler and Czado (2016). Brechmann et al. (2012) developed a model selection tool called truncation in order to reduce model complexity. Moreover, there has been a large number of applications in various fields such as finance (e.g. Maya et al., 2015; Kraus and Czado, 2017a), insurance (Shi and Yang, 2016), spatial statistics (e.g. Gräler, 2014; Erhardt et al., 2015), veterinary medicine (Barthel et al., 2016) or sociology (Cooke et al., 2015).

In this thesis we extend the existing literature on vine copulas by three important aspects. First, we consider model distances. In order to determine how much two vine copula models differ, Stöber et al. (2013), Spanhel and Kurz (2015) and Schepsmeier (2015) use the Kullback-Leibler (KL) distance, also known as KL divergence, developed in Kullback and Leibler (1951). Since the Kullback-Leibler distance requires multivariate integration

and corresponding analytical expressions are not known in general, its applicability is limited to a relatively low number of dimensions, even on high-performance computers. Maybe the greatest advantage of vine copulas is, however, that they can be used in very high dimensions; the methods presented in Müller and Czado (2016) for example allow for fitting models in hundreds of dimensions. Therefore, we provide modifications of the Kullback-Leibler distance in order to facilitate determining distances between high-dimensional models. For this purpose, we provide a representation of the KL distance as the sum over expectations of KL distances between univariate conditional densities. We reduce computational costs drastically by approximating these expectations via structured Monte Carlo integration on sparsely determined grids. We also consider symmetrized versions of our proposed KL-modifications, which can be interpreted as substitutes for the Jeffreys distance, a symmetrized version of the Kullback-Leibler distance. We verify the validity of these modifications and consider numerical examples and simulation studies to see that they outperform competing methods such as Monte Carlo integration. Further, we apply the proposed KL-modifications for model selection. In particular, we develop two algorithms for determining the optimal truncation level outperforming the methods proposed by Brechmann et al. (2012). For this purpose, we introduce a parametric bootstrap based hypothesis test deciding between copulas from nested model classes.

Secondly, we examine block maxima that are usually used to develop results in extreme-value theory as the block size goes to infinity (e.g. Genest and Nešlehová, 2012). We, however, investigate the behavior of finite componentwise block maxima in a multivariate framework, where the focus is on the dependence structure. The copula function of componentwise block maxima is known. We provide an explicit formula for the corresponding copula density. We argue why three-dimensional vine copulas are particularly useful in this setting and show numerical examples as well as an application to hydrological data. Hüsler and Reiss (1989) developed a scaling approach assuring that the block maxima of normal distributions converge to a non-trivial limit, i.e. the Hüsler-Reiss distribution with associated Hüsler-Reiss copula. We mimic this approach for vine copulas and investigate how the copula density of the scaled block maxima behaves for different block sizes in numerical examples.

Thirdly, we consider repeated measurement data that are obtained in longitudinal studies and can be found in several areas, especially in medical research. The most common approach for modeling such data is to use linear mixed models (see e.g. Diggle, 2002). However, vine copulas with a certain fixed sequential structure (so-called D-vine copulas) have recently also been applied by Smith et al. (2010) and Shi and Yang (2016) to model serial dependence in such longitudinal data. An extension to multivariate data can be found in Smith (2015) and Nai Ruscone and Osmetti (2017). All the above approaches work in a balanced setting, i.e. all individuals in the data set have the same number of

measurements. Shi et al. (2016) model the dependence structure in an unbalanced setting using Gaussian copulas. We develop a D-vine based model for unbalanced data. Various properties of this model are investigated: We provide a sequential estimation method that facilitates applications even for a large number of measurements and can handle missing values. For model selection we derive an adjusted version of one of the most frequently used selection criteria, the Bayesian information criterion (BIC). Further, we show that our proposed model can be seen as an extension of a rich class of linear mixed models. In an application to heart surgery data both model classes are compared. In particular, we illustrate the strengths of our approach by predicting conditional quantiles for future measurements.

## Outline of this thesis

The contents of this thesis are based on the following four research papers.

- Killiches, M., Kraus, D., and Czado, C. (2017b). Model distances for vine copulas in high dimensions. *Statistics and Computing*, doi:10.1007/s11222-017-9733-y.
- Killiches, M., Kraus, D., and Czado, C. (2017c). Using model distances to investigate the simplifying assumption, model selection and truncation levels for vine copulas. *arXiv preprint arXiv:1610.08795v3*. Under review at *Canadian Journal of Statistics*.
- Killiches, M. and Czado, C. (2015). Block-Maxima of Vines. In Dey, D. and Yan, J., editors, *Extreme Value Modelling and Risk Analysis: Methods and Applications*, pages 109–130. Boca Raton, FL: Chapman & Hall/CRC Press.
- Killiches, M. and Czado, C. (2017). A D-vine copula based model for repeated measurements extending linear mixed models with homogeneous correlation structure. *arXiv preprint arXiv:1705.06261*. Under review at *Biometrics*.

Chapter 2, which is based on Killiches et al. (2017b) and Killiches et al. (2017c), provides the necessary background that is needed throughout the rest of the thesis. We define copulas briefly in order to introduce vine copula afterwards. Two different representations for vine copulas are provided and the so-called simplifying assumption is discussed. Further, we clarify notation for vine copulas. Finally, simulation and fitting algorithms are presented.

In Chapter 3, which is based on contents of Killiches et al. (2017b) and Killiches et al. (2017c), we present model distances for vine copulas that are suited for applications in high dimensions. We first develop an alternative representation of the Kullback-Leibler distance as a sum of expectations over KL distances between univariate conditional densities. Based

on this result we develop three distance measures with decreasing computational costs. For this purpose we approximate the expectations in the alternative KL representation by structured Monte Carlo integration. The number of evaluation points is reduced by focusing only on diagonals. In plausibility checks and simulation studies we verify the validity of our proposed distances as substitutes for the KL. Similarly we substitute the Jeffreys distance, a symmetrized version of the KL, by symmetrizing our distances. In both cases we outperform competing methods such as Monte Carlo integration due to considerably lower computational effort.

Further, we apply the KL-substitutes for model selection. In particular, we present two algorithms that can be used for determining the optimal truncation level for a vine copula; our algorithms yield more precise results than the methods proposed in Brechmann et al. (2012). For our algorithms we develop a parametric bootstrap based test deciding between copulas from nested model classes.

Chapter 4, which presents material of Killiches and Czado (2015), considers the dependence structure of finite componentwise block maxima of multivariate distributions. The central result of this chapter is that we derive an explicit formula for the copula density of the vector of componentwise block maxima. We argue that vine copulas are particularly useful in this context and investigate how the copula density behaves for different block sizes in numerical examples and an application to hydrological data. In order to investigate if non-trivial limits are obtained (for block sizes going to infinity) when using proper scaling, we adapt the scaling for Gaussian distributions developed by Hüsler and Reiss (1989) to vine copulas and examine the results in numerical examples.

In Chapter 5, which is based on Killiches and Czado (2017), we develop a D-vine based model to describe repeated measurement data in an unbalanced setting. For comparison we introduce linear mixed models. A central aspect of the chapter is that the proposed model can be interpreted as an extension to the large class of linear mixed models for which the implied correlation structure is homogeneous over the considered individuals. Further, we provide a sequential estimation algorithm which can handle missing values and check its performance in a simulation study. In an application to a heart surgery data set the performance of both linear mixed models and our model is considered. In particular, results of the prediction of conditional quantiles for future measurements are compared.



# Chapter 2

---

## *Vine copula models*

---

Since vine copulas are the common theme of all parts of this thesis, we provide the necessary mathematical background in this chapter, which is in a large part based on Killiches et al. (2017b) and Killiches et al. (2017c).

After a brief definition of copulas in Section 2.1 we present vine copulas in Section 2.2. We introduce different ways of representation for vine copulas (Section 2.2.1 and Section 2.2.2) and approach the so-called simplifying assumption for vine copulas (Section 2.2.3). Section 2.2.4 clarifies the notation of vine copulas and Section 2.2.5 provides algorithms for the simulation from and the fitting of vine copulas.

### 2.1. Copulas

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

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)). \quad (2.1)$$

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

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

exists, one has

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

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

## 2.2. Vine copulas

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

### 2.2.1. Tree representation

In the following we consider a  $d$ -dimensional random vector  $\mathbf{U} = (U_1, \dots, U_d)^\top$  with uniform marginals  $U_j$ ,  $j = 1, \dots, d$ , following a copula  $C$  with corresponding copula density  $c$ . For  $j \in \{1, \dots, d\}$  and  $D \subseteq \{1, \dots, d\} \setminus \{j\}$  we denote by  $C_{j|D}$  the conditional distribution function of  $U_j$  given  $\mathbf{U}_D = (U_i)_{i \in D}$ . For  $j, k \in \{1, \dots, d\}$  and  $D \subseteq \{1, \dots, d\} \setminus \{j, k\}$  the copula density of the distribution associated with the *conditioned variables*  $U_j$  and  $U_k$  given the *conditioning variables*  $\mathbf{U}_D$  is denoted by  $c_{j,k;D}$ .

The structure of a  $d$ -dimensional vine copula is organized by a sequence of trees  $\mathcal{V} = (T_1, \dots, T_{d-1})$  satisfying

1.  $T_1 = (V_1, E_1)$  is a tree with nodes  $V_1 = \{1, \dots, d\}$  and edges  $E_1$ ;
2. For  $m = 2, \dots, d-1$ , the tree  $T_m$  consists of nodes  $V_m = E_{m-1}$  and edges  $E_m$ ;
3. Whenever two nodes of  $T_m$  are connected by an edge, the corresponding edges of  $T_{m-1}$  share a node ( $m = 2, \dots, d-1$ ).

The third property is often referred to as *proximity condition*.

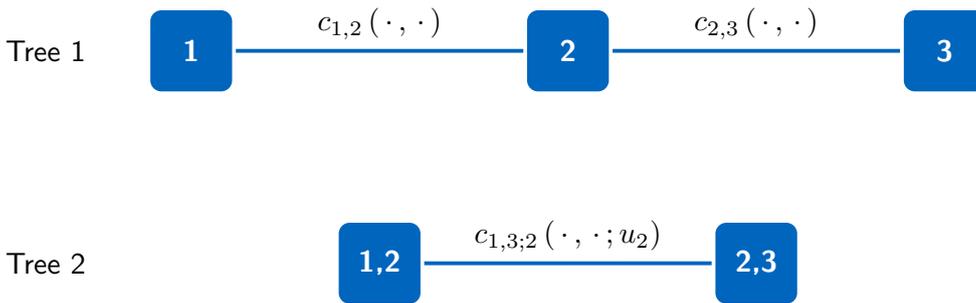
In a vine copula model each edge of the  $d - 1$  trees corresponds to a bivariate pair-copula. Let  $\bigcup_{m=1}^{d-1} \{c_{j_e, k_e; D_e} \mid e \in E_m\}$  be the set of pair-copulas associated with the edges in  $\mathcal{V}$ , where—following the notation of Czado (2010)— $j_e$  and  $k_e$  denote the indices of the conditioned variables  $U_{j_e}$  and  $U_{k_e}$  and  $D_e$  represents the conditioning set corresponding to edge  $e$ . The vine density can be written as

$$c(u_1, \dots, u_d) = \prod_{m=1}^{d-1} \prod_{e \in E_m} c_{j_e, k_e; D_e} (C_{j_e | D_e}(u_{j_e} | \mathbf{u}_{D_e}), C_{k_e | D_e}(u_{k_e} | \mathbf{u}_{D_e}); \mathbf{u}_{D_e}). \quad (2.2)$$

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

$$\begin{aligned} c(u_1, u_2, u_3) &= c_{1,3|2}(u_1, u_3 | u_2) c_2(u_2) \\ &\stackrel{\text{Sklar}}{=} c_{1,3;2} (C_{1|2}(u_1 | u_2), C_{3|2}(u_3 | u_2); u_2) c_{1|2}(u_1 | u_2) c_{2|3}(u_2 | u_3) \\ &= c_{1,3;2} (C_{1|2}(u_1 | u_2), C_{3|2}(u_3 | u_2); u_2) c_{1,2}(u_1, u_2) c_{2,3}(u_2, u_3), \end{aligned} \quad (2.3)$$

where  $c_{1,3|2}(\cdot, \cdot | u_2)$  denotes the density of the conditional distribution of  $(U_1, U_3) | U_2 = u_2$ , while  $c_{1,3;2}(\cdot, \cdot; u_2)$  is the associated copula density. The distribution function of the conditional distribution of  $U_j$  given  $U_2 = u_2$  is denoted by  $C_{j|2}(\cdot | u_2)$ ,  $j = 1, 3$ . Hence, we have expressed the three-dimensional copula density as the product over three bivariate pair-copulas. The corresponding tree representation can be found in Figure 2.1, where above each edge the associated pair-copula is denoted.



**Figure 2.1.:** Tree representation of a three-dimensional vine structure, where 2 is the central node. The associated pair-copulas are denoted above the edges.

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

$$c(u_1, u_2, u_3) = c_{2,3;1}(C_{2|1}(u_2 | u_1), C_{3|1}(u_3 | u_1); u_1) c_{1,2}(u_1, u_2) c_{1,3}(u_1, u_3) \text{ or}$$

$$c(u_1, u_2, u_3) = c_{1,2;3}(C_{1|3}(u_1|u_3), C_{2|3}(u_2|u_3); u_3) c_{1,3}(u_1, u_3) c_{2,3}(u_2, u_3).$$

This way of decomposing copula densities into bivariate building blocks can be extended to arbitrary dimensions yielding Equation 2.2. Morales-Nápoles (2011) show that in  $d$  dimensions there are  $\frac{d!}{2} \cdot 2^{\binom{d-2}{2}}$  possible vine decompositions. This flexibility and variety of choice can be of great advantage when it comes to modeling.

Vine copulas with general tree structure are often referred to as *regular vines* or in short *R-vines*. Special cases of vine copula structures are so-called *C-vines* and *D-vines*. In a C-vine for each tree  $T_m$  there exists a root node with degree  $d - m$ , i.e. it is a neighbor of all other nodes. Each tree then has a star-like structure. For a D-vine each node in tree  $T_1$  has a degree of at most 2 such that the trees are simply connected paths.

### 2.2.2. Matrix representation

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

**Definition 2.1** (Vine structure matrix). A lower-triangular matrix  $M = (m_{i,j})_{i,j=1}^d$  with non-zero entries  $m_{i,j} \in \{1, \dots, d\}$ ,  $i > j$ , is called a *vine structure matrix* if it has the following three properties:

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

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

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

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

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

$$c(u_1, \dots, u_d) = \prod_{j=1}^{d-1} \prod_{k=j+1}^d c_{m_{k,j}, m_{j,j}; m_{k+1,j}, \dots, m_{d,j}} \left( C_{m_{k,j}|m_{k+1,j}, \dots, m_{d,j}}(u_{m_{k,j}}|u_{m_{k+1,j}}, \dots, u_{m_{d,j}}), \right. \\ \left. C_{m_{j,j}|m_{k+1,j}, \dots, m_{d,j}}(u_{m_{j,j}}|u_{m_{k+1,j}}, \dots, u_{m_{d,j}}); u_{m_{k+1,j}}, \dots, u_{m_{d,j}} \right). \quad (2.4)$$

The resemblance of Equation 2.4 and Equation 2.2 is obvious. The only difference between the two formulas is that in Equation 2.4 the indices of the pair-copulas are denoted by the entries of the structure matrix, whereas in Equation 2.2 they are represented by the edges of the tree representation. Both notations have their advantages: The tree representation is easy to interpret and can be illustrated graphically; the matrix representation is very concise and is particularly useful for programming-related purposes. We will use both representations depending on which one is better suited for the respective objective.

In our three-dimensional example (Equation 2.3) the structure matrix looks as follows:

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

The entries  $m_{3,1} = 2$  (together with  $m_{1,1} = 1$ ) and  $m_{3,2} = 3$  (together with  $m_{2,2} = 2$ ) in the last row represent  $c_{1,2}(u_1, u_2)$  and  $c_{2,3}(u_2, u_3)$ , respectively. In both cases, the conditioning set is empty because the considered entries are the last ones in their columns. The entry  $m_{2,1}$  (together with  $m_{1,1}$  and  $m_{3,1}$ ) encodes the expression  $c_{1,3;2}(C_{1|2}(u_1|u_2), C_{3|2}(u_3|u_2); u_2)$  since the indices of the conditioned variables are given by  $m_{2,1} = 3$  and  $m_{1,1} = 1$  and the conditioning variable is  $m_{3,1} = 2$ . Multiplying these three factors leads to the expression from Equation 2.3. Note that there is not a unique way of encoding a given vine decomposition into a structure matrix. For instance, exchanging  $m_{2,2}$  and  $m_{3,2}$  in the above example yields the same vine decomposition.

### 2.2.3. Simplifying assumption

When it comes to modeling, for tractability reasons most authors assume that for pair-copulas with a non-empty conditioning set the copula itself does not depend on the conditioning variables (e.g.  $c_{1,3;2}(\cdot, \cdot; u_2) = c_{1,3;2}(\cdot, \cdot)$  for any  $u_2 \in [0, 1]$ ). This assumption is referred to as the *simplifying assumption*. Among others, Hobæk Haff et al. (2010), Acar et al. (2012), Stöber et al. (2013), Spanhel and Kurz (2015) and Killiches et al. (2017a) discuss when this assumption is justified. Since *simplified vines*, i.e. vine copulas satisfying the simplifying assumption, are in practice the most relevant class of vine copulas—especially in high dimensions—, all examples and applications in this thesis use simplified vines. Nevertheless, all of the presented concepts are also applicable to non-simplified vines.

Stöber et al. (2013) investigated which multivariate copulas could be represented as simplified vines: Similar to the relationship between correlation matrices and partial correlations (Bedford and Cooke, 2002), every Gaussian copula can be written as a simplified Gaussian vine, i.e. a vine copula with only bivariate Gaussian pair-copulas, where any (valid) vine structure can be used and the parameters are the corresponding partial correlations. Vice versa, every Gaussian vine represents a Gaussian copula. Further, t copulas can also be decomposed into simplified vines with arbitrary (valid) vine structure. The pair-copulas are then bivariate t copulas, the association parameters are the corresponding partial correlations and the degrees of freedom in tree  $T_m$  are  $\nu + (m - 1)$ , where  $\nu$  is the degrees of freedom parameter of the t copula. However, a regular vine copula with only bivariate t copulas, called a t vine, does not necessarily represent a t copula. Moreover, Stöber et al. (2013) proved that the only Archimedean copula that can be decomposed into a simplified vine copula is the Clayton copula. The pair-copulas in the vine copula representation are then bivariate Clayton copulas with associated parameters  $\theta/(m\theta + 1)$  for all pairs in the  $m$ th tree, where  $\theta$  is the parameter of the Clayton copula. Similarly as for the t copula, a regular vine copula with only bivariate Clayton copulas (a Clayton vine) does not necessarily represent a Clayton copula.

### 2.2.4. Notation of simplified parametric vine copulas

Since we typically work in a simplified parametric framework we specify each pair-copula of the vine decomposition as a parametric bivariate copula (with up to two parameters). In order to represent a  $d$ -dimensional vine copula using the tree notation we specify a triplet  $\mathcal{R} = (\mathcal{V}, \mathcal{B}, \Theta)$ , where  $\mathcal{V} = (T_1, \dots, T_{d-1})$  denotes the tree sequence defining the vine structure,  $\mathcal{B}$  is the set of pair-copula families and  $\Theta$  are the corresponding parameters.

Alternatively, for the matrix representation, we borrow the concept of the vine structure matrix to introduce a lower-triangular family matrix  $B = (b_{i,j})_{i,j=1}^d$  and two lower-

triangular parameter matrices  $P^{(k)} = (p_{i,j}^{(k)})_{i,j=1}^d$ ,  $k = 1, 2$ , containing the pair-copula families and associated parameters of  $c_{m_{i,j}, m_{j,j} | m_{i+1,j}, \dots, m_{d,j}}$ , respectively. Since we only use one- and two-parametric copula families, two parameter matrices are sufficient. The entries of the family and parameter matrices,  $b_{i,j}$ ,  $p_{i,j}^{(1)}$  and  $p_{i,j}^{(2)}$ , specify the pair-copula corresponding to the entry  $m_{i,j}$ . For one-parametric families we set the corresponding entry in the second parameter matrix to zero. In order to compare the strengths of dependence of different copula families, we also compute the Kendall's  $\tau$  values  $k_{i,j}$  corresponding to pair-copulas with family  $b_{i,j}$  and parameters  $p_{i,j}^{(1)}$  and  $p_{i,j}^{(2)}$  and store them in a lower-triangular matrix  $K = (k_{i,j})_{i,j=1}^d$ . Note that  $k_{i,j}$  is associated with a pair-copula and does in general not represent the Kendall'  $\tau$  between  $U_i$  and  $U_j$ . A simplified vine copula can then be written as the quadruple  $\mathcal{R} = (M, B, P^{(1)}, P^{(2)})$ .

### 2.2.5. Simulation and fitting of vine copulas

One of the main reasons why vine copula are considered to be a very useful tool for modeling dependence in practice is that there is software available that can be used for example for simulation and fitting. All these implementations are contained in the R library `VineCopula` (Schepsmeier et al., 2017) for a parametric simplified framework. Handling non-parametric simplified and parametric non-simplified vines is numerically challenging but there is software available: `kdevine` (Nagler, 2017) and `gamCopula` (Vatter and Nagler, 2016), respectively. We use `VineCopula` for all numerical vine copula related applications throughout this thesis. The parametric bivariate copulas used as candidate models are Gaussian, Student t, Clayton, Gumbel, Frank, Joe, BB1, BB6, BB7, BB8, Tawn type 1 and Tawn type 2 as well as their survival versions and 90/270 degree rotations (for details see Schepsmeier et al., 2017).

For simulation and Monte Carlo integration it is important that we can sample from vine copula distributions. Stöber and Czado (2012) and Joe (2014) provide sampling algorithms for arbitrary vine copulas. They are based on the inverse Rosenblatt transformation (Rosenblatt, 1952), which is given by  $T_c: [0, 1]^d \rightarrow [0, 1]^d$ ,  $\mathbf{w} = (w_1, \dots, w_d)^\top \mapsto (T_{c,1}(\mathbf{w}), \dots, T_{c,d}(\mathbf{w}))^\top$ . The components of  $T_c(\mathbf{w})$  can recursively be defined by  $T_{c,m_{d,d}}(\mathbf{w}) = w_{m_{d,d}}$  and

$$T_{c,m_{j,j}}(\mathbf{w}) = C_{m_{j,j} | m_{j+1,j+1}, \dots, m_{d,d}}^{-1}(w_{m_{j,j}} | T_{c,m_{j+1,j+1}}(\mathbf{w}), \dots, T_{c,m_{d,d}}(\mathbf{w})) \quad (2.5)$$

for  $j = 1, \dots, d-1$ , where  $m_{j,j}$  denotes the  $j$ th diagonal entry of the structure matrix of the vine copula. The corresponding Rosenblatt transform is given by  $T_c^{-1}: [0, 1]^d \rightarrow [0, 1]^d$ ,  $\mathbf{u} = (u_1, \dots, u_d)^\top \mapsto (T_{c,1}^{-1}(\mathbf{u}), \dots, T_{c,d}^{-1}(\mathbf{u}))^\top$ , where  $T_{c,m_{d,d}}^{-1}(\mathbf{u}) = u_{m_{d,d}}$  and

$$T_{c,m_{j,j}}^{-1}(\mathbf{u}) = C_{m_{j,j} | m_{j+1,j+1}, \dots, m_{d,d}}(u_{m_{j,j}} | u_{m_{j+1,j+1}}, \dots, u_{m_{d,d}}). \quad (2.6)$$

The sampling algorithm then works as follows: First, sample  $w_j \sim \text{uniform}(0, 1)$  for  $j = 1, \dots, d$ . Then, apply an inverse Rosenblatt transform  $T_c$  to the uniform sample, i.e.  $\mathbf{u} = (u_1, \dots, u_d)^\top = T_c(\mathbf{w})$ , where  $\mathbf{w} = (w_1, \dots, w_d)^\top$  is mapped from the (uniform) w-scale to the (warped) u-scale in the following way:

- $u_{m_{d,d}} := w_{m_{d,d}}$ ,
- $u_{m_{d-1,d-1}} = C_{m_{d-1,d-1}|m_{d,d}}^{-1}(w_{m_{d-1,d-1}}|u_{m_{d,d}})$ ,
- $\vdots$
- $u_{m_{1,1}} = C_{m_{1,1}|m_{2,2}, \dots, m_{d,d}}^{-1}(w_{m_{1,1}}|u_{m_{2,2}}, \dots, u_{m_{d,d}})$ .

Note that the appearing (inverse) conditional distribution functions can be obtained easily for vine copulas (Stöber and Czado, 2012, Section 5.3). This sampling algorithm is implemented in `VineCopula` as `RVineSim`.

Using the tree representation of vine copulas, Dißmann et al. (2013) developed a sequential estimation method that fits a simplified parametric vine, i.e. the structure as well the corresponding pair-copula families and parameters, to a given data set tree-by-tree. *Dißmann's algorithm* is the most frequently used procedure for fitting vine copulas and works as follows: First, the empirical Kendall's  $\tau$  values are calculated for all pairs. Then, a spanning tree maximizing the sum of absolute Kendall's  $\tau$  values is determined such that most dependence is captured in the first tree of the vine. For every edge the maximum-likelihood estimate for each possible pair-copula from the candidate set is determined. Then, the pair-copula with the highest likelihood, AIC or BIC is assigned to the edge. Having specified the first tree the pseudo-data for the second tree is determined by applying the fitted conditional distribution functions. For the second tree, the empirical Kendall's  $\tau$  values for all edges admissible with respect to the proximity condition are determined. Then, as for the first tree, a maximal spanning tree with corresponding optimal pair-copulas is selected. This procedure is repeated until all  $d - 1$  trees of the vine copula are specified. For a more detailed description see Dißmann et al. (2013). This algorithm is also implemented in `VineCopula` as the function `RVineStructureSelect`.

# Chapter 3

---

## *Model distances for vine copulas*

---

The contents of this chapter are a lightly edited reproduction of the published contents in Killiches et al. (2017b) and of parts of the submitted contents in Killiches et al. (2017c). Sections 3.1 and 3.7 consist of modified parts of both Killiches et al. (2017b) and Killiches et al. (2017c). Sections 3.2 and 3.3 are based on Killiches et al. (2017b) and Sections 3.4 to 3.6 present contents of Killiches et al. (2017c).

### 3.1. Introduction

In the course of growing data sets and increasing computing power, statistical data analysis has considerably developed within the last decade. The necessity of proper dependence modeling has become evident at least since the financial crisis of 2007. Using vine copulas is a popular option to approach this task. The advantage of these models is that they are flexible and numerically tractable even in high dimensions.

Since it is interesting in many cases to determine how much two models differ, some authors like Stöber et al. (2013) and Schepsmeier (2015) use the *Kullback–Leibler (KL) distance* (Kullback and Leibler, 1951), also known as *KL divergence*, as a model distance between vines. A symmetrized version of the KL distance is given by the *Jeffreys distance (JD)* (Jeffreys, 1946). In model selection for copulas the KL distance is frequently used (see for example Chen and Fan, 2005, 2006; Diks et al., 2010). In the context of vine copulas, Joe (2014, Section 5.7) used the KL distance to calculate the sample size necessary to discriminate between two densities. Investigating the simplifying assumption Hobæk Haff et al. (2010) used the KL distance to find the simplified vine closest to a given non-simplified vine and Stöber et al. (2013) gage the strength of non-simplifiedness of the trivariate Farlie-Gumbel-Morgenstern (FGM) copula for different dependence parameters. Similarly, Spanhel and Kurz (2015) use the KL distance to assess the quality of simplified

vine copula approximations. However, all popular distance measures require multivariate integration, which is why they can only deal with up to three- or four-dimensional models in a reasonable amount of time.

In this chapter we will address the question of how to measure the distance between two vine copulas even for high dimensions and show how to use distance measures for model selection in two applications. For this purpose, we develop methods based on the Kullback–Leibler distance, where we use the fact that it can be expressed as the sum over expectations of KL distances between univariate conditional densities. By cleverly approximating these expectations in different ways, we introduce three new distance measures with varying focuses. The *approximate Kullback–Leibler distance* (aKL) aims to approximate the true Kullback–Leibler distance via structured Monte Carlo integration and is a computationally tractable distance measure in up to five dimensions. The *diagonal Kullback–Leibler distance* (dKL) focuses on the distance between two vine copulas on specific conditioning vectors, namely those lying on certain diagonals in the space. We show that even though the resulting distance measure does not approximate the KL distance in a classical sense, it still reproduces its qualitative behavior quite well. While this way of measuring distances between vines is fast in up to ten dimensions, we still have to reduce the number of evaluation points in order to get a numerically tractable distance measure for dimensions 30 and higher. By concentrating on only one specific diagonal we achieve this, defining the *single diagonal Kullback–Leibler distance* (sdKL). The lack of symmetry of the KL distance and its substitutes is overcome by developing similar approximations to the Jeffreys distance. In numerous examples and applications we illustrate that the proposed methods are valid distance measures and outperform benchmark approaches like Monte Carlo integration regarding computational time. Moreover, in order to enable the assessment of the size of our developed distance measures we provide a baseline calibration based on the comparison of specific Gaussian copulas to the independence copula. Further, we show possible fields of applications for the dKL and sdKL in model selection. For this purpose we develop a hypothesis test that answers the question if the distance between two models from nested model classes is significant. Then we show how to select the best model out of a list of candidate models with the help of a model distance based measure. Finally, we also use the new distance measures and the developed hypothesis test to answer the question how to determine the optimal truncation level of a fitted vine copula, a task already recently discussed by Brechmann et al. (2012) and Brechmann and Joe (2015). Truncation methods have the aim of enabling high-dimensional vine copula modeling by severely reducing the number of used parameters without changing the fit of the resulting model too much.

The remainder of this chapter is organized as follows: In Section 3.2 we develop the above mentioned modified model distances for vine copulas and perform several plausibility

checks on their performance. Section 3.3 contains a simulation studies comparing the performances of all introduced distance measures. In order to facilitate model selection using model distances we provide a hypothesis test based on parametric bootstrapping in Section 3.4. In Section 3.5 we show how the model distances can be used to assess the best model fit out of a set of candidate models. As a final application the determination of the optimal truncation level of a vine copula is discussed in Section 3.6. Section 3.7 concludes the chapter with some summarizing comments.

## 3.2. Model distances for vines

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

In order to simplify notation, for the remainder of this chapter we assume that the diagonal of a  $d$ -dimensional structure matrix is given by  $1:d$ . This assumption comes without any loss of generality: Property 2 from Definition 2.1 implies that the diagonal of any vine structure matrix is a permutation of  $1:d$ , where we use the notation  $r:s$  to describe the vector  $(r, r+1, \dots, s)^\top$  for  $r \leq s$ . Hence, relabeling of the variables suffices to obtain the desired property.

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

**Definition 3.1** (Matched Gaussian vine). For a simplified vine copula  $\mathcal{R} = (M, B, P^{(1)}, P^{(2)})$  let  $K = (k_{i,j})_{i,j=1}^d$  denote the lower-triangular matrix that contains the corresponding Kendall's  $\tau$  values. Then, the *matched Gaussian vine* of  $\mathcal{R}$  is given by the vine copula  $\tilde{\mathcal{R}} = (M, \tilde{B}, \tilde{P}^{(1)}, \tilde{P}^{(2)})$ , where  $\tilde{B}$  is a family matrix where all entries are Gaussian pair-copulas, parameter matrix  $\tilde{P}^{(1)} = (\tilde{p}_{i,j}^{(1)})_{i,j=1}^d$  with  $\tilde{p}_{i,j}^{(1)} = \sin(\frac{\pi}{2}k_{i,j})$  and  $\tilde{P}^{(2)}$  is a zero-matrix.

### 3.2.1. Kullback–Leibler distance

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

Leibler distance between  $f$  and  $g$  is defined as

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

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

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

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

$$\text{JD}(f, g) = \text{KL}(f, g) + \text{KL}(g, f). \quad (3.3)$$

Since the Jeffreys distance is just a sum of two Kullback–Leibler distances, we will in the following sections concentrate on the KL distance and apply our results to the Jeffreys distance in Section 3.3.2.

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

$$\text{KL}(f, g) = \text{KL}(c^f, c^g). \quad (3.4)$$

In this chapter we are mainly interested in comparing different models that are obtained by fitting a data set. Since we usually first estimate the margins and afterwards the dependence structure (cf. IFM method in Joe, 1997, Section 10.1), the assumption of identical margins is always fulfilled. Hence, we will in the following concentrate on calculating the Kullback–Leibler distance between copula densities.

Having a closer look at the definition of the KL distance, we see that for its calculation a  $d$ -dimensional integral has to be evaluated. In general, this cannot be done analytically and, further, is numerically infeasible in high dimensions. For example, Schepsmeier (2015) stresses the difficulty of numerical integration in dimensions 8 and higher. In this section,

we propose modifications of the Kullback–Leibler distance designed to be computationally tractable and still measure model distances adequately. These modifications are all based on the following proposition that shows that the KL distance between  $d$ -dimensional copula densities  $c^f$  and  $c^g$  can be expressed as the sum over expectations of KL distances between univariate conditional densities.

**Proposition 3.2.** For two copula densities  $c^f$  and  $c^g$  it holds:

$$\text{KL}(c^f, c^g) = \sum_{j=1}^d \mathbb{E}_{c_{(j+1):d}^f} \left[ \text{KL} \left( c_{j|(j+1):d}^f(\cdot | \mathbf{U}_{(j+1):d}), c_{j|(j+1):d}^g(\cdot | \mathbf{U}_{(j+1):d}) \right) \right], \quad (3.5)$$

where  $\mathbf{U}_{(j+1):d} \sim c_{(j+1):d}^f$  and  $(d+1):d := \emptyset$ . Further,  $c_{j|(j+1):d}^f(\cdot | u_{j+1}, \dots, u_d)$  denotes the univariate conditional density of  $U_j | (U_{j+1}, \dots, U_d)^\top = (u_{j+1}, \dots, u_d)^\top$  implied by the density  $c^f$ .

We will prove an even more general version of Proposition 3.2 that holds for arbitrary densities  $f$  and  $g$ :

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

where  $\mathbf{X}_{(j+1):d} \sim f_{(j+1):d}$  and  $f_{j|(j+1):d}(\cdot | x_{j+1}, \dots, x_d)$  denotes the univariate conditional density of  $X_j | (X_{j+1}, \dots, X_d)^\top = (x_{j+1}, \dots, x_d)^\top$  implied by  $f$ . Proposition 3.2 then follows directly from this statement.

*Proof.* Recall that using recursive conditioning we can obtain for density  $f$

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

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

$$\begin{aligned} \text{KL}(f, g) &= \int_{\mathbf{x} \in \mathbb{R}^d} \ln \left( \frac{f(\mathbf{x})}{g(\mathbf{x})} \right) f(\mathbf{x}) \, d\mathbf{x} \\ &= \int_{\mathbf{x} \in \mathbb{R}^d} \sum_{j=1}^d \ln \left( \frac{f_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d})}{g_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d})} \right) f(\mathbf{x}) \, d\mathbf{x} \\ &= \sum_{j=1}^d \int_{x_d \in \mathbb{R}} \cdots \int_{x_1 \in \mathbb{R}} \ln \left( \frac{f_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d})}{g_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d})} \right) f(x_1, \dots, x_d) \, dx_1 \cdots dx_d \end{aligned}$$

$$\begin{aligned}
&= \sum_{j=1}^d \int_{x_d \in \mathbb{R}} \cdots \int_{x_j \in \mathbb{R}} \ln \left( \frac{f_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d})}{g_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d})} \right) \\
&\quad \times \left\{ \int_{x_{j-1} \in \mathbb{R}} \cdots \int_{x_1 \in \mathbb{R}} f(x_1, \dots, x_d) dx_1 \cdots dx_{j-1} \right\} dx_j \cdots dx_d \\
&= \sum_{j=1}^d \int_{x_d \in \mathbb{R}} \cdots \int_{x_j \in \mathbb{R}} \ln \left( \frac{f_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d})}{g_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d})} \right) f_{j,\dots,d}(x_j, \dots, x_d) dx_j \cdots dx_d \\
&= \sum_{j=1}^d \int_{x_d \in \mathbb{R}} \cdots \int_{x_{j+1} \in \mathbb{R}} \left\{ \int_{x_j \in \mathbb{R}} \ln \left( \frac{f_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d})}{g_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d})} \right) \right. \\
&\quad \left. \times f_{j|(j+1):d}(x_j | \mathbf{x}_{(j+1):d}) dx_j \right\} f_{(j+1):d}(\mathbf{x}_{(j+1):d}) dx_{j+1} \cdots dx_d \\
&= \sum_{j=1}^d \mathbb{E}_{f_{(j+1):d}} \left[ \text{KL} \left( f_{j|(j+1):d}(\cdot | \mathbf{X}_{(j+1):d}), g_{j|(j+1):d}(\cdot | \mathbf{X}_{(j+1):d}) \right) \right].
\end{aligned}$$

□

Proposition 3.2 is especially useful if  $c^f$  and  $c^g$  are vine copula densities since for a vine copula with structure matrix  $M$  the appearing (univariate) conditional density  $c_{j|(j+1):d}$  of  $U_j | (U_{j+1}, \dots, U_d)^\top = (u_{j+1}, \dots, u_d)^\top$  can be easily obtained by taking the product over all pair-copula expressions corresponding to the entries in the  $j$ th column of  $M$ . We will prove this in Proposition 3.3.

**Proposition 3.3.** Let  $\mathbf{U} = (U_1, \dots, U_d)^\top$  be a random vector with vine copula density  $c$  and corresponding structure matrix  $M = (m_{i,j})_{i,j=1}^d$ . Then, for  $j < d$

$$\begin{aligned}
c_{j|(j+1):d}(u_j | u_{j+1}, \dots, u_d) &= \prod_{k=j+1}^d c_{m_{k,j}, m_{j,j}; m_{k+1,j}, \dots, m_{d,j}} \left( \right. \\
&\quad C_{m_{k,j} | m_{k+1,j}, \dots, m_{d,j}}(u_{m_{k,j}} | u_{m_{k+1,j}}, \dots, u_{m_{d,j}}), \\
&\quad \left. C_{m_{j,j} | m_{k+1,j}, \dots, m_{d,j}}(u_{m_{j,j}} | u_{m_{k+1,j}}, \dots, u_{m_{d,j}}); u_{m_{k+1,j}}, \dots, u_{m_{d,j}} \right).
\end{aligned} \tag{3.6}$$

*Proof.* From Equation 2.4 we know that the vine copula density can be written as a product over the pair-copula expressions corresponding to the matrix entries. In Property 2.8 (ii), Dißmann et al. (2013) state that deleting the first row and column from a  $d$ -dimensional structure matrix yields a  $(d-1)$ -dimensional trimmed structure matrix. Due to Property 2 from Definition 2.1 the entry  $m_{1,1} = 1$  does not appear in the remaining

matrix. Hence, we obtain the density  $c_{2:d}$  by taking the product over all pair-copula expressions corresponding to the entries in the trimmed matrix. Iterating this argument yields that the entries of matrix  $M_k := (m_{i,j})_{i,j=k+1}^d$  resulting from cutting the first  $k$  rows and columns from  $M$  represent the density  $c_{(k+1):d}$ . In general, we have

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

The numerator and denominator can be obtained as the product over all pair-copula expressions corresponding to the entries of  $M_{j-1}$  and  $M_j$ . Thus,  $c_{j|(j+1):d}$  is simply the product over the expressions corresponding to the entries from the first column of  $M_{j-1}$ . This proves Equation 3.6.  $\square$

As an example, combining the results from Proposition 3.2 and Proposition 3.3, for four-dimensional copula densities  $c^f$  and  $c^g$  we can write:

$$\begin{aligned} \text{KL}(c^f, c^g) &= \mathbb{E}_{c_{2:4}^f} \left[ \text{KL} \left( c_{1|2:4}^f(\cdot | \mathbf{U}_{2:4}), c_{1|2:4}^g(\cdot | \mathbf{U}_{2:4}) \right) \right] \\ &\quad + \mathbb{E}_{c_{3,4}^f} \left[ \text{KL} \left( c_{2|3,4}^f(\cdot | \mathbf{U}_{3,4}), c_{2|3,4}^g(\cdot | \mathbf{U}_{3,4}) \right) \right] \\ &\quad + \mathbb{E}_{c_4^f} \left[ \text{KL} \left( c_{3|4}^f(\cdot | U_4), c_{3|4}^g(\cdot | U_4) \right) \right] + 0 \end{aligned} \quad (3.7)$$

where for instance

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

The zero in the last line of Equation 3.7 results from the fact that  $c_4^f(u_4) = c_4^g(u_4) = 1$  for all  $u_4 \in [0, 1]$ . This is generally the case for the  $d$ th summand in Equation 3.5, which will therefore be omitted in the following. Further note that the last non-zero term of Equation 3.7 can also be written as  $\text{KL} \left( c_{3,4}^f(\cdot, \cdot), c_{3,4}^g(\cdot, \cdot) \right)$ .

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

$$\mathbb{E}_f[h(\mathbf{X})] \approx \frac{1}{N_{\text{MC}}} \sum_{i=1}^{N_{\text{MC}}} h(\mathbf{x}_i), \quad (3.8)$$

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

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

$$\text{KL}(c^f, c^g) \approx \sum_{j=1}^{d-1} \frac{1}{|\mathcal{U}_j|} \sum_{\mathbf{u}_{(j+1):d} \in \mathcal{U}_j} \text{KL}\left(c_{j|(j+1):d}^f(\cdot | \mathbf{u}_{(j+1):d}), c_{j|(j+1):d}^g(\cdot | \mathbf{u}_{(j+1):d})\right). \quad (3.9)$$

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

Throughout the subsequent sections we assume the following setting: Let  $\mathcal{R}^f$  and  $\mathcal{R}^g$  be two  $d$ -dimensional vines with copula densities  $c^f$  and  $c^g$ , respectively. We assume that their vine structure matrices have the same entries on the diagonals, i.e.  $\text{diag}(M^f) = \text{diag}(M^g)$ . Note that, although this assumption is a restriction, there are still  $2^{\binom{d-2}{2} + d - 2}$  different vine decompositions with equal diagonals of the structure matrix, which is shown in Proposition 3.4.<sup>2</sup> As before, without loss of generality we set the diagonals equal to  $1:d$ .

**Proposition 3.4.** Let  $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_d)^\top$  be a permutation of  $1:d$ . Then, there exist  $2^{\binom{d-2}{2} + d - 2}$  different vine decompositions whose structure matrix has the diagonal  $\boldsymbol{\sigma}$ .

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

---

<sup>2</sup>This includes, for example, C- and D-vines having the same diagonal.

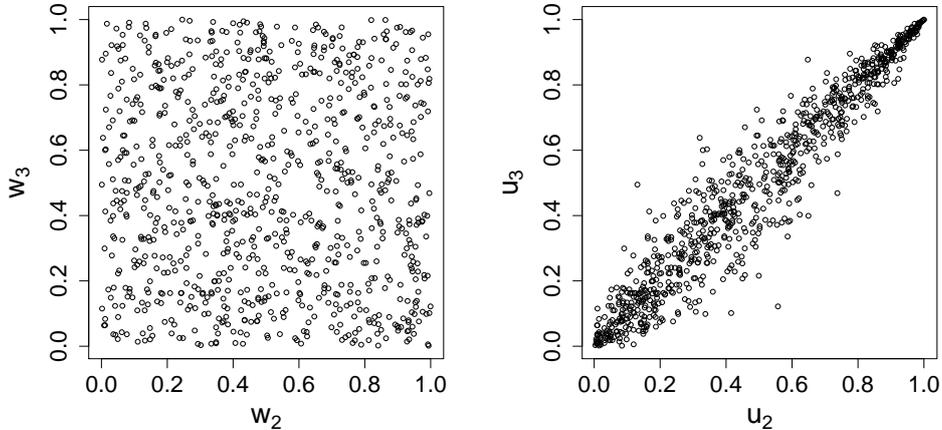
there are  $d!$  different diagonals. Thus, for a fixed diagonal  $\sigma$  there exist

$$\frac{\frac{d!}{2} \cdot 2^{\binom{d-2}{d}} \cdot 2^{d-1}}{d!} = 2^{\binom{d-2}{2} + d - 2}$$

different vine decompositions. □

### 3.2.2. Approximate Kullback–Leibler distance

We illustrate the idea of the approximate Kullback–Leibler distance at the example of two three-dimensional vines  $\mathcal{R}^f$  and  $\mathcal{R}^g$ . For the first summand ( $j = 1$ ) of Equation 3.9, the KL distance between  $c_{1|2,3}^f(\cdot | u_2, u_3)$  and  $c_{1|2,3}^g(\cdot | u_2, u_3)$  is calculated for all pairs  $(u_2, u_3)^\top$  contained in the grid  $\mathcal{U}_1$ . In this example we assume that the pair-copula  $c_{2,3}^f$  is a Gumbel copula with parameter  $\theta = 6$  (implying a Kendall’s  $\tau$  value of 0.83). Regarding the choice of the grid, if we used the Monte Carlo method,  $\mathcal{U}_1$  would contain a random sample of  $c_{2,3}^f$ . Recall from Section 2.2.5 that such a sample can be generated by simulating from a uniform distribution on  $[0, 1]^2$  and applying the inverse Rosenblatt transformation  $T_{c_{2,3}^f}$ . Figure 3.1 displays a sample of size 900 on the (uniform) w-scale and its transformation via  $T_{c_{2,3}^f}$  to the (warped) u-scale.

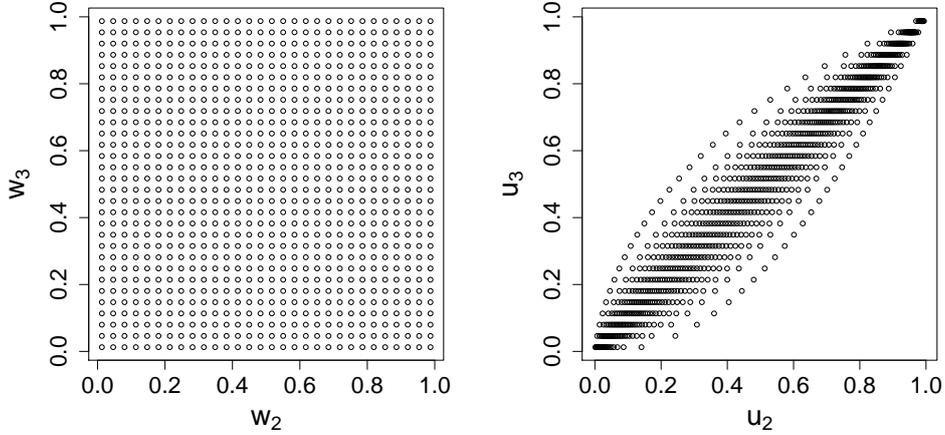


**Figure 3.1.:** Sample of size 900 from the uniform distribution (left) and corresponding warped sample under transformation  $T_{c_{2,3}^f}$ , which is a sample from a Gumbel copula with  $\theta = 6$  (right).

As mentioned before we do not want our distance measure to be random. This motivates us to introduce the concept of *structured Monte Carlo integration*: Instead of sampling from the uniform distribution on the w-scale, we use a *structured grid*  $\mathcal{W}$ , which is an equidistant lattice on the two-dimensional unit cube<sup>3</sup>, and transform it to the warped u-

<sup>3</sup>Since most copulas have an infinite value at the boundary of the unit cube, we usually restrict ourselves to  $[\varepsilon, 1 - \varepsilon]^d$  for a small  $\varepsilon > 0$ .

scale by applying the inverse Rosenblatt transformation  $T_{c_{2,3}^f}$  (cf. Equation 2.5). Figure 3.2 shows an exemplary structured grid with 30 grid points per margin.



**Figure 3.2.:** Structured grid with 30 grid points per margin (left) and corresponding warped grid under transformation  $T_{c_{2,3}^f}$  (right).

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

**Definition 3.5** (Approximate Kullback–Leibler distance). Let  $\mathcal{R}^f$  and  $\mathcal{R}^g$  be as described above. Further, let  $n \in \mathbb{N}$  be the number of grid points per margin and  $\varepsilon > 0$ . Then, the *approximate Kullback–Leibler distance* (aKL) between  $\mathcal{R}^f$  (reference vine) and  $\mathcal{R}^g$  is defined as

$$\text{aKL}(\mathcal{R}^f, \mathcal{R}^g) := \sum_{j=1}^{d-1} \frac{1}{|\mathcal{G}_j|} \sum_{\mathbf{u}_{(j+1):d} \in \mathcal{G}_j} \text{KL} \left( c_{j|(j+1):d}^f(\cdot | \mathbf{u}_{(j+1):d}), c_{j|(j+1):d}^g(\cdot | \mathbf{u}_{(j+1):d}) \right),$$

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

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

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

Proposition 3.6 shows that the approximate KL distance in fact approximates the true KL distance in the sense that the aKL converges to the KL for  $\varepsilon \rightarrow 0$  and  $n \rightarrow \infty$ . A proof can be found in Appendix A.1.

**Proposition 3.6.** Let  $\mathcal{R}^f$  and  $\mathcal{R}^g$  be as described above. Then,

$$\lim_{\varepsilon \rightarrow 0} \lim_{n \rightarrow \infty} \text{aKL}(\mathcal{R}^f, \mathcal{R}^g) = \text{KL}(c^f, c^g).$$

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

**Example 3.7 (Four-dimensional aKL-example).** We consider a data set from the Euro Stoxx 50, already used in Brechmann and Czado (2013). It covers a 4-year period (May 22, 2006 to April 29, 2010) containing 985 daily observations. The Euro Stoxx 50 is a major index consisting of the stocks of 50 large European companies. In order to obtain copula data Brechmann and Czado (2013, Appendix A) first fitted ARMA(1,1)-GARCH(1,1), AR(1)-GARCH(1,1), MA(1)-GARCH(1,1) and GARCH(1,1) models, respectively, to the univariate marginal time series. In this example we consider the copula data of the following four national indices: the Dutch AEX ( $U_1$ ), the Italian FTSE MIB ( $U_2$ ), the German DAX ( $U_3$ ) and the Spanish IBEX 35 ( $U_4$ ). Fitting a simplified vine to the data yields:

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

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

As usual for financial data, most of the pair-copulas selected by the fitting algorithm are t copulas with rather high dependence; only  $c_{14;23}$  is modeled as a Frank copula. Now we compute the approximate KL distance between this (reference) vine and its matched Gaussian vine (see Definition 3.1) and compare it to the numerically integrated KL distance. The latter limits our example to low dimensions because numerical integration becomes very slow in higher dimensions. Even in four dimensions we have to set the tolerance level of the integration routine `adaptIntegrate` of the package `cubature` from its default level of  $10^{-5}$  to  $10^{-4}$  to obtain results within less than 10 days. Throughout the chapter we will also consider examples for  $d \geq 5$ , where numerical integration becomes (almost) infeasible. As a substitute benchmark for the numerically integrated KL distance, we compare our approximated KL values to the corresponding Monte Carlo Kullback–

Leibler (MCKL) values, where the expectation in Equation 3.2 is approximated by Monte Carlo integration, i.e.

$$\text{MCKL}(c^f, c^g) := \frac{1}{N_{\text{MC}}} \sum_{i=1}^{N_{\text{MC}}} \ln \left( \frac{c^f(\mathbf{u}^i)}{c^g(\mathbf{u}^i)} \right), \quad (3.10)$$

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

Table 3.1 displays the approximate Kullback–Leibler distance between the fitted (reference) vine and its matched Gaussian vine for different values of  $\beta$  and  $n$  together with the corresponding computational time (in hours).<sup>4</sup> We further present the numerically and Monte Carlo integrated KL distances. In order to facilitate comparability, for each value of  $\beta$  we compute the integrals on the corresponding domain of integration with volume  $\beta$ .

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

**Table 3.1.:** Approximate, numerically integrated and Monte Carlo integrated KL distances for different parameter settings with corresponding computational times (in hours).

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

**Remark 3.8.** During the review process of Killiches et al. (2017b) an anonymous referee suggested to compare our approach of the (structured) warped grid to using the Latin

<sup>4</sup>All numerical calculations in this chapter were performed on a Linux computer (8-way Opteron) with 32 cores (each with 2.6 GHz and 3.9 GB of memory).

Hypercube sampling (LHS) method, which is a quasi-random sampler guaranteeing that the sample points are more evenly spread across the unit hypercube compared to standard Monte Carlo-methods (cf. McKay et al., 1979). Section 3.3.2 contains a simulation study assessing the performance of the introduced distance measures. There, we also implemented LHS and compared its performance. The results showed no improvement and still had the disadvantage of being random with a rather high volatility. This is a property we wanted to avoid with our approach. The weaker performance may result from the fact that tail behavior cannot be captured sufficiently by LHS with a small sample size; for larger sample sizes LHS loses its competitiveness due to very long computational times. For these reason we omit a thorough discussion of the LHS.

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

### 3.2.3. Diagonal Kullback–Leibler distance

In order to illustrate the idea behind the *diagonal Kullback–Leibler distance* we continue our example from Section 3.2.2. Figure 3.3 shows the structured and warped grids used for the aKL (gray circles). Additionally, the diagonal grid points are highlighted by filled diamonds.

The idea is now to reduce the evaluation grids  $\mathcal{U}_j$  to the diagonal grids in order to define a distance measure related to the original KL distance with the advantage of reduced computational costs. For this, we formally define the sets of diagonals and warped discretized diagonals.

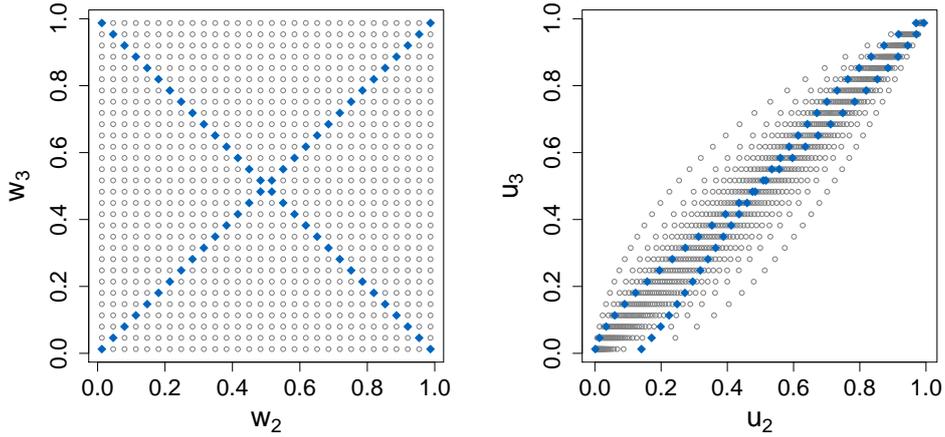
**Definition 3.9** (Diagonals and warped discretized diagonals). For  $j = 1, \dots, d-1$ , we define the set of diagonals in  $[0, 1]^{d-j}$ :

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

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

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

is the direction vector corresponding to the corner point  $\mathbf{r}$ . Note that the set of diagonals  $D_j$  only contains  $2^{d-j-1}$  elements since every diagonal is implied by two corner points (e.g. the diagonals  $\{(0, \dots, 0)^\top + t(1, \dots, 1)^\top \mid t \in [0, 1]\}$  and  $\{(1, \dots, 1)^\top + t(-1, \dots, -1)^\top \mid t \in [0, 1]\}$  coincide). Further, let  $D_{j,1}, \dots, D_{j,2^{d-j-1}}$  be an arbitrary ordering of the  $2^{d-j-1}$  diagonals. We define the  $k$ th *discretized diagonal* on the w-scale as  $\mathcal{D}_{j,k}^w := D_{j,k} \cap \mathcal{W}_j$ , where  $\mathcal{W}_j$  is the structured grid in  $[0, 1]^{d-j}$  defined in Definition 3.5, such that it contains  $n$  grid points (cf. left panel of Figure 3.3). Finally, the  $k$ th *warped discretized diagonal* on the u-scale is defined as  $\mathcal{D}_{j,k}^u := T_{c_{(j+1):d}^f}(\mathcal{D}_{j,k}^w)$ , where  $T_{c_{(j+1):d}^f}$  is defined as in Definition 3.5 (cf. right panel of Figure 3.3).



**Figure 3.3.:** Structured grid with highlighted diagonals consisting of 30 evaluation points (left) and corresponding warped grid under transformation  $T_{c_{2,3}^f}$  (right).

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

$$\mathcal{D}_j^u := \bigcup_{k=1}^{2^{d-j-1}} \mathcal{D}_{j,k}^u$$

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

**Definition 3.10** (Diagonal Kullback–Leibler distance). Let  $\mathcal{R}^f$ ,  $\mathcal{R}^g$  and  $\mathcal{D}_j^u$  be as described above. Then, the *diagonal Kullback–Leibler distance* dKL between  $\mathcal{R}^f$  (reference vine) and  $\mathcal{R}^g$  is defined as

$$\text{dKL}(\mathcal{R}^f, \mathcal{R}^g) := \sum_{j=1}^{d-1} \frac{1}{|\mathcal{D}_j^u|} \sum_{\mathbf{u}_{(j+1):d} \in \mathcal{D}_j^u} \text{KL} \left( c_{j|(j+1):d}^f(\cdot | \mathbf{u}_{(j+1):d}), c_{j|(j+1):d}^g(\cdot | \mathbf{u}_{(j+1):d}) \right),$$

where  $|\mathcal{D}_j^u| = n \cdot 2^{d-j-1}$ .

**Remark 3.11.** Similar to Proposition 3.6 one can show (see Appendix A.2.1) that for each of the  $2^{d-j-1}$  diagonals  $\mathcal{D}_{j,k}^u$  it holds

$$\begin{aligned} \lim_{\varepsilon \rightarrow 0} \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{\mathbf{u}_{(j+1):d} \in \mathcal{D}_{j,k}^u} \text{KL} \left( c_{j|(j+1):d}^f(\cdot | \mathbf{u}_{(j+1):d}), c_{j|(j+1):d}^g(\cdot | \mathbf{u}_{(j+1):d}) \right) \\ = \frac{1}{\sqrt{d-j}} \int_{\mathbf{u}_{(j+1):d} \in D_{j,k}^u} \text{KL} \left( c_{j|(j+1):d}^f(\cdot | \mathbf{u}_{(j+1):d}), c_{j|(j+1):d}^g(\cdot | \mathbf{u}_{(j+1):d}) \right) \\ \times c_{(j+1):d}^f(\mathbf{u}_{(j+1):d}) \, d\mathbf{u}_{(j+1):d}, \end{aligned}$$

where  $D_{j,k}^u := T_{c_{(j+1):d}^f}(D_{j,k})$ . Hence, the diagonal Kullback–Leibler distance can be interpreted as a sum of scaled approximated line integrals over weighted univariate KL distances between conditional densities, which is exactly the integrand appearing in Proposition 3.2. Having the infeasibility of the aKL in higher dimensions in mind, the reduction to points on lines seems to be a good choice in order to reduce the approximation of multivariate integrals to one-dimensional ones. We choose the warped diagonals as these lines since they on the one hand contain points with high density values due to the warping and on the other hand each let all components of the conditioning vector take values on the whole range from 0 to 1. Since in practice vine models tend to differ most in the tails of the distributions, we increase the concentration of evaluation points in the tails by transforming the discretized diagonal with the method described at the end of Appendix A.2.2.

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

### Applications of the dKL

**Example 3.12** (Example 3.7 continued). We continue Example 3.7 and apply the dKL to measure the distance between the fitted four-dimensional (reference) vine and its matched Gaussian vine using different numbers of grid points  $n$  per diagonal and  $\beta = 95\%$  as usual. The results and computational times (in seconds) are displayed in Table 3.2.

We observe that the dKL values seem to converge quite fast and are already quite close to their limit for small  $n$ . Of course, we cannot expect them to converge to the original KL distance, but we see that the order of magnitude is the same as the values



$\nu$	$\tau$	dKL	MCKL	ratio	$\nu$	$\tau$	dKL	MCKL	ratio
3	0.5	0.857	0.374	2.29	3	-0.7	4.702	3.226	1.46
5	0.5	0.376	0.162	2.32	3	-0.5	2.106	1.431	1.47
7	0.5	0.209	0.091	2.30	3	-0.3	0.740	0.473	1.56
10	0.5	0.109	0.047	2.31	3	-0.1	0.077	0.050	1.54
15	0.5	0.051	0.023	2.26	3	0.1	0.067	0.048	1.41
20	0.5	0.029	0.013	2.20	3	0.3	0.561	0.423	1.33
25	0.5	0.019	0.008	2.25	3	0.5	1.740	1.262	1.38
30	0.5	0.013	0.006	2.23	3	0.7	4.590	2.982	1.54

**Table 3.3.:** Left table: dKL ( $n = 10$ ) and MCKL ( $N_{MC} = 10^6$ ) values between five-dimensional t copulas with  $K = K_5(0.5)$  and  $P^{(2)} = V_5(\nu)$  and their matched Gaussian vines. Right table: dKL and MCKL values between five-dimensional t copulas with  $K = K_5(\tau)$  and  $P^{(2)} = V_5(3)$  and their matched Gaussian vines.

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

In the second plausibility check we also deal with five-dimensional t copulas decomposed as above. However, in this scenario the degrees of freedom are fixed to be equal to 3 and the value for  $\tau$  in  $K_5(\tau)$  is ranging between  $-0.7$  and  $0.7$ . All these vines are compared to a t copula with Kendall's  $\tau$  matrix  $K_5(0)$  and degrees of freedom  $\nu = 3$ . The dKL and MCKL distances between the resulting eight (reference) t copulas and the t copula with Kendall's  $\tau$  matrix  $K_5(0)$  is shown in Table 3.3 (right table).

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

In the third plausibility check we consider five-dimensional Gumbel vines (i.e. every pair-copula is a bivariate Gumbel copula having upper tail dependence) with the same structure matrix  $D_d$  and Kendall's  $\tau$  matrix  $K_5(0.5)$ . In Table 3.4 we compare this reference vine to its matched Gaussian vine and other vines constructed similarly using one copula family only but retaining the same dependence in terms of the Kendall's  $\tau$  matrix. As other copula families we choose the Clayton copula exhibiting lower tail dependence, the

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

Family	Reference family	dKL	MCKL	ratio
Gaussian	Gumbel	0.369	0.205	1.80
Clayton	Gumbel	2.987	1.780	1.68
Survival Clayton	Gumbel	0.322	0.158	2.04
Joe	Gumbel	0.483	0.249	1.94

**Table 3.4.:** dKL and MCKL values between a five-dimensional (reference) Gumbel D-vine and D-vines with the same Kendall's  $\tau$  matrix constructed using one copula family only. The last column contains the ratio between dKL and MCKL.

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

**Example 3.14** (Performance in different dimensions). In order to assess the performance of the diagonal KL distance regarding computational time in different dimensions, we again make use of the Euro Stoxx 50 data Brechmann and Czado (2013). We take the copula data of the 12 German stocks (with ticker symbols ALV, BAS, BAYN, DAI, DB1, DBK, DTE, EOAN, MUV2, RWE, SIE, SAP, corresponding to  $U_1, \dots, U_{12}$ ) and fit vines to the first  $d$  variables ( $d = 3, \dots, 12$ ). We display the dKL distance ( $n = 10$ ) between these (reference) vines and their matched Gaussian vines with corresponding computational times (in seconds) in Table 3.5. Again, we also present the approximated KL values using Monte Carlo integration ( $N_{MC} = 10^6$ ) and the ratio between dKL and MCKL values. While we observe that the dKL is exceptionally fast in low dimensions we note that computational times more than double when moving up one dimension. This is reasonable since the total number of diagonals in all evaluation grids is equal to

$$\sum_{j=1}^{d-1} |\mathcal{D}_j^u| = \sum_{j=1}^{d-1} 2^{d-j-1} = 2^{d-1} - 1 \quad (3.13)$$

and thus grows exponentially in  $d$ . Further, the evaluations of the conditional copula densities become more costly in higher dimensions, which can also be seen by the fact that

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

**Table 3.5.:** dKL values ( $n = 10$ ) and MCKL values ( $N_{\text{MC}} = 10^6$ ) with corresponding computational times (in seconds) for vines with different dimensions based on the stock exchange data. The fifth row contains the ratio between dKL and MCKL.

the computational times for the MCKL increase even though the number of evaluations  $N_{\text{MC}}$  stays constant. Comparing the computational times of the dKL and MCKL one notices that the dKL is considerably faster than the MCKL in lower dimensions. Only in dimensions 10 and higher it loses this competitive advantage. Considering the ratio between dKL and MCKL values it seems that the dKL values increase slightly faster with the dimension  $d$  than the MCKL values.

The preceding examples suggest that with the dKL distance we have found a valid distance measure between vines with reasonable computational times for up to ten dimensions. However, we are still interested in finding a distance measure computable in dimensions of order 30 to 50. To achieve this, the number of grid points should not depend on the dimension of the evaluation grid, implying a constant number of grid points. Hence, we choose only one of the  $2^{d-j-1}$  warped discrete diagonals in  $\mathcal{D}_j^y$  to be the evaluation grid. While this may seem like a very severe restriction (with the curse of dimensionality in mind), two heuristic observations justify this approach. On the one hand we observe that most of the  $2^{d-j-1}$  diagonals contain many grid points with density values close to zero while there is always one diagonal whose points have very large density values. On the other hand we will see that the properties of the distance measure using only this single diagonal for the evaluation grid still pass the plausibility checks with values behaving closely to those of the dKL and the MCKL.

### 3.2.4. Single diagonal Kullback–Leibler distance

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

**Definition 3.15** (Diagonal weighting measure). Assume we can parametrize a diagonal  $D \subseteq [0, 1]^d$  (on the u-scale) by the mapping  $\gamma: [0, 1] \rightarrow [0, 1]^d$ . Let  $c: [0, 1]^d \rightarrow [0, \infty)$  be

a copula density. Then, we define

$$\lambda_c(D) := \int_{\boldsymbol{\xi} \in D} c(\boldsymbol{\xi}) \, d\boldsymbol{\xi} = \int_{t \in [0,1]} c(\boldsymbol{\gamma}(t)) \|\dot{\boldsymbol{\gamma}}(t)\| \, dt \quad (3.14)$$

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

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

**Definition 3.16** (Single diagonal Kullback–Leibler distance). Let  $\mathcal{R}^f, \mathcal{R}^g$  be as before and  $k_j^* := \arg \max_k \lambda_{c_{(j+1):d}^f}(D_{j,k}^u)$  be the index of the diagonal  $D_{j,k}^u$  with the highest weight according to  $\lambda_{c_{(j+1):d}^f}$ ,  $j = 1, \dots, d-1$ . Further, let  $\mathcal{D}_{j,k_j^*}^u$  with  $|\mathcal{D}_{j,k_j^*}^u| = n$  be a discretization of  $D_{j,k_j^*}^u$ . Then, the *single diagonal Kullback–Leibler distance* (sdKL) between  $\mathcal{R}^f$  (reference vine) and  $\mathcal{R}^g$  is defined as

$$\text{sdKL}(\mathcal{R}^f, \mathcal{R}^g) := \sum_{j=1}^{d-1} \frac{1}{|\mathcal{D}_{j,k_j^*}^u|} \sum_{\mathbf{u}_{(j+1):d} \in \mathcal{D}_{j,k_j^*}^u} \text{KL} \left( c_{j|(j+1):d}^f(\cdot | \mathbf{u}_{(j+1):d}), c_{j|(j+1):d}^g(\cdot | \mathbf{u}_{(j+1):d}) \right).$$

**Remark 3.17.** From Remark 3.11 we know that the single diagonal Kullback–Leibler distance approximates a scaled line integral over weighted univariate KL distances between conditional densities along the diagonal with the highest weight.

To find this diagonal we actually would have to calculate the integral of  $c_{(j+1):d}^f$  over each of the  $2^{d-j-1}$  diagonals. In practice, this may be infeasible for high dimensions. Therefore, we propose a more sophisticated method to find a candidate for the diagonal with the highest weight. Similar to the hill-climbing algorithm used to find optimal graph structures in Bayesian networks (see Tsamardinos et al., 2006), we choose a starting value in form of a certain diagonal implied by the vine’s unconditional dependencies and look in the “neighborhood” of this diagonal for another diagonal with higher weight. This procedure is repeated until a (local) maximum is found. The two procedures of finding a suitable starting diagonal and locally searching for better candidates are described in Appendix A.3.

In the following we continue the plausibility checks from Example 3.13 in order to demonstrate that the restriction to a single diagonal is still enough for the resulting distance measure to generate reasonable results.

**Example 3.18** (Plausibility checks (Example 3.13 continued)). Table 3.6 and Table 3.7 repeat the three plausibility checks of Example 3.13. The resulting sdKL values obviously also pass these tests resembling the behavior of the MCKL values quite closely with

### 3.3. Comparison of all introduced model distances

relatively steady sdKL/MCKL-ratios. Evaluating at only one diagonal in each grid reduces computational times even more such that the sdKL is roughly 180 times faster than the MCKL.

$\nu$	$\tau$	sdKL	MCKL	ratio	$\nu$	$\tau$	sdKL	MCKL	ratio
3	0.5	0.754	0.374	2.02	3	-0.7	7.534	3.226	2.34
5	0.5	0.330	0.162	2.04	3	-0.5	3.100	1.431	2.17
7	0.5	0.184	0.091	2.03	3	-0.3	0.773	0.473	1.63
10	0.5	0.097	0.047	2.06	3	-0.1	0.053	0.050	1.05
15	0.5	0.046	0.023	2.04	3	0.1	0.157	0.048	3.30
20	0.5	0.026	0.013	1.98	3	0.3	1.322	0.423	3.12
25	0.5	0.017	0.008	2.04	3	0.5	3.226	1.262	2.56
30	0.5	0.012	0.006	2.03	3	0.7	6.193	2.982	2.08

**Table 3.6.:** Left table: sdKL ( $n = 10$ ) and MCKL ( $N_{MC} = 10^6$ ) values between five-dimensional t copulas with  $P^{(2)} = V_5(\nu)$  and their matched Gaussian vines. Right table: sdKL and MCKL values between five-dimensional t copulas with  $K = K_5(\tau)$  and their matched Gaussian vines.

Family	Reference family	sdKL	MCKL	ratio
Gaussian	Gumbel	0.394	0.205	1.92
Clayton	Gumbel	3.557	1.780	2.00
Survival Clayton	Gumbel	0.421	0.158	2.66
Joe	Gumbel	0.576	0.249	2.32

**Table 3.7.:** sdKL and MCKL values between a five-dimensional Gumbel vine and vines constructed using one copula family only. The last column contains the ratio between sdKL and MCKL.

These five-dimensional plausibility checks empirically show that the reduction from all to one diagonal still yields viable results for our modified version of the KL distance. In the following section we will compare all introduced distance measures.

## 3.3. Comparison of all introduced model distances

### 3.3.1. Comparison of all introduced KL approximations

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

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

where  $\det(\cdot)$  denotes the determinant and  $\text{tr}(\cdot)$  the trace of a matrix. For each dimension  $d$  we use a reference Gaussian vine  $\mathcal{R}^0$  (which is also a Gaussian copula) with the (D-vine) structure matrix  $M = D_d$  and Kendall's  $\tau$  matrix  $K = K_d(0.5)$  (cf. Equation 3.11 and Equation 3.12, respectively).

We generate another  $m = 50$  Gaussian vines  $\mathcal{R}^r$ ,  $r = 1, \dots, m$ , with the same structure matrix  $M = D_d$  and a parameter matrix  $P^{(1)}$ , where the  $d(d-1)/2$  partial correlations are simulated such that the corresponding correlation matrix is uniform over the space of valid correlation matrices. For this purpose, we follow Joe (2006): For  $i = 2, \dots, d$  and  $j = 1, \dots, i-1$  we draw  $q_{i,j}$  from a Beta( $i/2, i/2$ ) distribution. The parameter  $p_{i,j}^{(1)}$  is then obtained as the linear transformation of  $q_{i,j}$  to  $[-1, 1]$ :  $p_{i,j}^{(1)} := 2(q_{i,j} - 0.5)$ .

We compare the reference vine  $\mathcal{R}^0$  to each  $\mathcal{R}^r$  using the model distances KL, MCKL, aKL, dKL and sdKL. Since the Kullback–Leibler distance is exact in these cases, we can assess the performance of the remaining distance measures by comparing their  $m = 50$  distance values to the ones of the true KL. As the scale of the KL and related distance measures cannot be interpreted in a sensible way, we are only interested in how well the ordering suggested by the KL is reproduced by aKL ( $n = 20$ ), dKL ( $n = 10$ ), sdKL ( $n = 10$ ) and MCKL ( $N_{\text{MC}} = 10^6$ ), respectively. Hence, we consider the respective rank correlations to the KL values in order to assess their performances. The results and average computation times are displayed in Table 3.8 and Table 3.9, respectively. For illustration, the true KL values (between reference vine  $\mathcal{R}^0$  and each  $\mathcal{R}^r$ ,  $r = 1, \dots, 50$ ) are plotted against the corresponding sdKL values for  $d = 20$  in the left plot Figure 3.4. The corresponding right plot shows the same results but the KL and sdKL values were transformed to the rank level first.

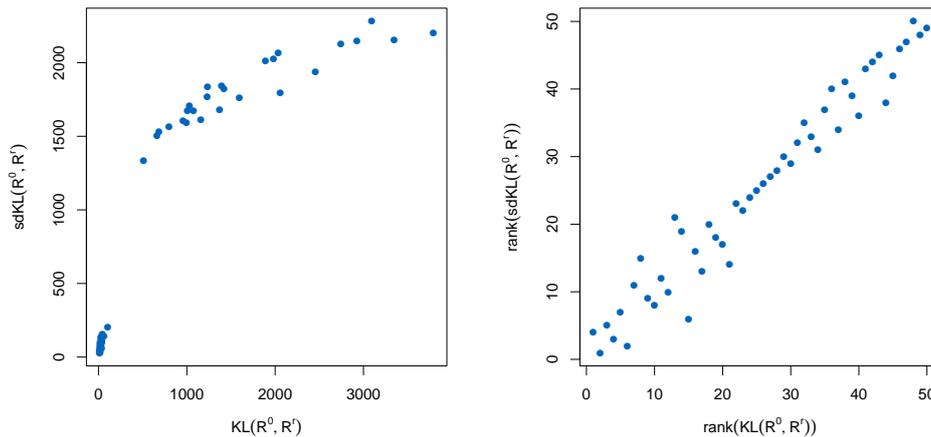
$d$	3	4	5	7	10	15	20	30
aKL	98.4	98.5	98.4	–	–	–	–	–
dKL	96.7	97.4	98.7	98.7	97.2	–	–	–
sdKL	93.3	90.2	91.5	89.7	82.9	84.8	84.5	80.4
MCKL	99.8	99.5	99.5	99.7	98.5	97.2	92.3	91.7

**Table 3.8.:** Rank correlations (in percent) between the true KL distance values (between reference vine  $\mathcal{R}^0$  and each  $\mathcal{R}^r$ ,  $r = 1, \dots, 50$ ) and the corresponding distance values obtained by aKL, dKL, sdKL and MCKL, respectively, for  $d = 3, 4, 5, 7, 10, 15, 20, 30$ .

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

**Table 3.9.:** Average computational times (in seconds) of aKL, dKL, sdKL and MCKL (from Table 3.8) for  $d = 3, 4, 5, 7, 10, 15, 20, 30$ .

### 3.3. Comparison of all introduced model distances



**Figure 3.4.:** Left: Plot of the true KL distance values (between reference vine  $\mathcal{R}^0$  and each  $\mathcal{R}^r$ ,  $r = 1, \dots, 50$ ) and the corresponding distance values obtained by sdKL for  $d = 20$ . Right: Plot of the ranks of the true KL distance values (between reference vine  $\mathcal{R}^0$  and each  $\mathcal{R}^r$ ,  $r = 1, \dots, 50$ ) and the ranks of the corresponding distance values obtained by sdKL for  $d = 20$ .

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

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

### 3.3.2. Comparison of the resulting JD approximations

All approximations of the KL distance discussed in this chapter can be easily used to define corresponding approximations of the Jeffreys distance (see Equation 3.3). We will call these aJD, dJD, sdJD and MCJD, where for example  $\text{sdJD}(f, g) = \text{sdKL}(f, g) + \text{sdKL}(g, f)$ . We repeat the simulation study from Section 3.3.1 for the comparison of the approximated JD values to the true one. Table 3.10 displays the results.

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

**Table 3.10.:** Rank correlations (in percent) between the true JD values (between reference vine  $\mathcal{R}^0$  and each  $\mathcal{R}^r$ ,  $r = 1, \dots, 50$ ) and the corresponding distance values obtained by aJD, dJD, sdJD and MCJD, respectively, for  $d = 3, 4, 5, 7, 10, 15, 20, 30$ .

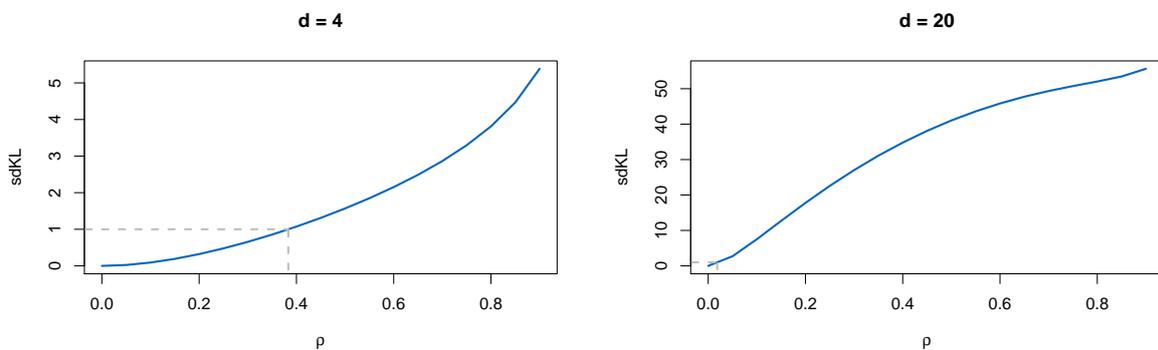
We see that the results are similar to the ones where we just considered the KL distance. Of course, being sums of two approximated KL distances the approximations of the Jeffreys distance are more volatile and therefore perform slightly worse than their KL counterparts. However, we still have aJD and dJD values close to 100% and sdJD values around 85%. As one would have expected, computational times of the JD substitutes are simply (approximately) twice as long as the ones of their KL counterparts (cf. Table 3.9). So we can conclude that in low dimensions suitable substitutes for the Jeffreys Distance are given by the aJD and dJD with better computational times than those of the MCJD. In dimensions 10 and higher the sdJD would be the measure of choice with low computational times and high correlations to the true Jeffreys distance. In practice users can decide if they want to apply substitutes of the Jeffreys distance or the Kullback–Leibler distance, depending on whether the focus is on symmetry of the distance measure or computational times.

### 3.3.3. Calibration

The results in Section 3.3.1 showed that the sdKL is a valid substitute for the Kullback–Leibler distance since it ranks the differences between the considered models very similarly. However, what still remains is the drawback that any divergence or distance measure shares: one distance value alone cannot be interpreted properly. Therefore, we will provide a baseline comparison in this section. Based on these results one can assess whether a certain sdKL-value is small or large. Of course, this procedure can similarly be used to calibrate any of the other distance measures. As reference vines we take exchangeable  $d$ -dimensional Gaussian copulas with correlation matrix  $\Sigma(\rho) = (\sigma_{i,j}(\rho))_{i,j=1,\dots,d}$  with

### 3.3. Comparison of all introduced model distances

$\sigma_{i,j}(\rho) = 1$  for  $i = j$  and  $\sigma_{i,j}(\rho) = \rho$  for  $i \neq j$ , i.e. every pair of variables has the same correlation coefficient  $\rho$ . All copulas are written as D-vines with structure matrix  $D_d$  (cf. Equation 3.11), where all parameters, i.e. the corresponding partial correlations, in the  $\ell$ th tree are equal and recursively given by  $\rho^{(\ell)} = \rho^{(\ell-1)}/(1 + \rho^{(\ell-1)})$  for  $\ell = 2, \dots, d-1$  and  $\rho^{(1)} = \rho$ . This implies for example that for  $\rho = 0.5$  the partial correlations of the pairs in the  $\ell$ th tree are given by  $\rho^{(\ell)} = 1/(\ell + 1)$  such that the strength of dependence decreases with the tree level. The above specified (reference) Gaussian copulas are compared to the  $d$ -dimensional independence copula using the sdKL. Figure 3.5 shows the sdKL-values for  $d = 4$  and  $d = 20$  for values of  $\rho$  between 0 and 0.90.



**Figure 3.5.:** Plots of the sdKL between the exchangeable (reference) Gaussian copula with joint correlation  $\rho$  and the independence copula against  $\rho$  for  $d = 4$  (left) and  $d = 20$  (right).

Of course, both graphs start at 0 as for  $\rho = 0$  the exchangeable Gaussian copula is simply the independence copula such that the compared models are the same. As one would expect, the distance increases as  $\rho$  increases, regardless of the dimension. However, we see that the scale for  $d = 4$  is very different from that of  $d = 20$ . Whereas a sdKL-value of 1 corresponds to a  $\rho$  of roughly 0.4 in 4 dimensions, for  $d = 20$  it corresponds to a  $\rho$  of approximately 0.02 (see gray dashed lines in Figure 3.5). Plots like the ones in Figure 3.5 can now be used as a baseline comparison: If we obtain an sdKL-value of 1 between two four-dimensional vine copulas, we know that this is comparable to how much an exchangeable Gaussian copula with  $\rho = 0.4$  differs from the independence copula, which is in fact considerable. If we get the same sdKL-value for  $d = 20$ , this corresponds to the difference between an exchangeable Gaussian copula with  $\rho = 0.02$  and the independence copula, which is not too extreme.

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

Gaussian copulas and the independence copula to classify the sdJD value as small or large.

Since we have seen that the substitutes for KL and JD perform similarly well, we will focus on dKL and sdKL for the remainder of the chapter.

### 3.4. Hypothesis test for model selection

In this section we provide a procedure based on parametric bootstrapping (see Efron and Tibshirani, 1994) for choosing between a parsimonious and a more complex model.

Assume we have two nested classes of  $d$ -dimensional parametric copula models  $\mathbb{C}^f \subseteq \mathbb{C}^g$  and a copula data set  $\mathbf{u}_i^0 \in [0, 1]^d$ ,  $i = 1, \dots, N$ , with true underlying distribution  $\mathcal{R}^g \in \mathbb{C}^g$ . We want to investigate whether a model from  $\mathbb{C}^f$  suffices to describe the data. In other words, we want to test the null hypothesis  $H_0: \mathcal{R}^g \in \mathbb{C}^f$ , which means that there exists  $\mathcal{R}^f \in \mathbb{C}^f$  such that  $\mathcal{R}^f = \mathcal{R}^g$ . Due to the identity of indiscernibles (i.e.  $\text{KL}(\mathcal{R}^f, \mathcal{R}^g) = 0$  if and only if  $\mathcal{R}^f = \mathcal{R}^g$ ) of the Kullback–Leibler distance this is equivalent to  $\text{KL}(\mathcal{R}^f, \mathcal{R}^g) = 0$ . Hence, for testing  $H_0$  we can examine whether the KL distance between  $\mathcal{R}^f$  and  $\mathcal{R}^g$  is significantly different from zero. In practice,  $\mathcal{R}^f$  and  $\mathcal{R}^g$  are unknown and have to be estimated from the data  $\mathbf{u}_i^0 \in [0, 1]^d$ . Consider the KL distance  $d_0 = \text{KL}(\hat{\mathcal{R}}_0^f, \hat{\mathcal{R}}_0^g)$  between the two fitted models  $\hat{\mathcal{R}}_0^f$  and  $\hat{\mathcal{R}}_0^g$  as the test statistic. Since the distribution of  $d_0$  cannot be derived analytically we use a parametric bootstrapping scheme to retrieve it:

For  $j = 1, \dots, M$ , where  $M$  is the number of bootstrap iterations, generate a sample  $\mathbf{u}_i^j \in [0, 1]^d$ ,  $i = 1, \dots, N$ , from  $\hat{\mathcal{R}}_0^f$ . Fit copulas  $\hat{\mathcal{R}}_j^f \in \mathbb{C}^f$  and  $\hat{\mathcal{R}}_j^g \in \mathbb{C}^g$  to the generated sample. Calculate the distance between  $\hat{\mathcal{R}}_j^f$  and  $\hat{\mathcal{R}}_j^g$ :

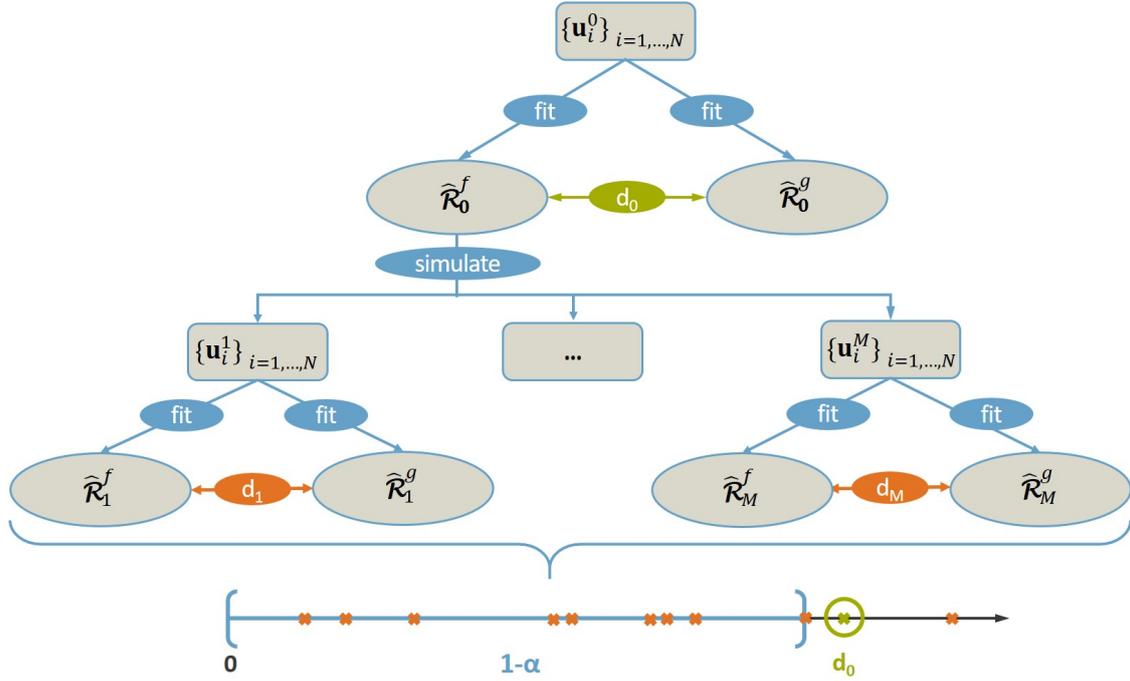
$$d_j = \text{KL}(\hat{\mathcal{R}}_j^f, \hat{\mathcal{R}}_j^g).$$

Now reorder the set  $\{d_j \mid j = 1, \dots, M\}$  such that  $d_1 < d_2 < \dots < d_M$ . For a significance level  $\alpha \in (0, 1)$ , we can determine an empirical confidence interval  $I_{1-\alpha}^M \subseteq [0, \infty)$  with confidence level  $1 - \alpha$  by

$$I_{1-\alpha}^M = [0, d_{\lceil M(1-\alpha) \rceil}],$$

where  $\lceil \cdot \rceil$  denotes the ceiling function. Finally, we can reject  $H_0$  if  $d_0 \notin I_{1-\alpha}^M$ . Figure 3.6 illustrates the above procedure in a flow chart. At the bottom the resulting distances are plotted on the positive real line. In this exemplary case,  $d_0$  (circled cross) lies outside the range of the empirical  $100(1 - \alpha)\%$  confidence interval and therefore  $H_0$  can be rejected at the  $100\alpha\%$  level, i.e. there is a significant difference between  $\hat{\mathcal{R}}_0^f$  and  $\hat{\mathcal{R}}_0^g$ .

Since in higher dimensions the KL distance cannot be calculated in a reasonable amount of time, we use the distance measures dKL (for  $d < 10$ ) and sdKL (for  $d \geq 10$ ), introduced in Section 3.2, as substitutes for the KL distance. The above bootstrapping scheme works



**Figure 3.6.:** Scheme of the testing procedure based on parametric bootstrapping.

similarly using the substitutes.

Of course, it is not obvious per se how to choose the number of bootstrap samples  $M$ . On the one hand we want to choose  $M$  as small as possible (due to computational time); on the other hand we want the estimate of  $d_{\lceil M(1-\alpha) \rceil}$  to be as precise as possible in order to avoid false decisions with respect to the null hypothesis (the upper bound  $d_{\lceil M(1-\alpha) \rceil}$  of the confidence interval  $I_{1-\alpha}^M$  is random with variance decreasing in  $M$ ). Therefore, we choose  $M$  so large that we can decide if  $d_0$  is significantly larger (smaller) than  $d_{\lceil M(1-\alpha) \rceil}$ . For this purpose, we construct a  $100(1 - \beta)\%$  confidence interval for  $d_{\lceil M(1-\alpha) \rceil}$  by considering the distribution of the  $\lceil M(1 - \alpha) \rceil$ th order statistic, which is given as follows (see for example Casella and Berger, 2002, page 232):

$$F_{d_{\lceil M(1-\alpha) \rceil}}(x) = \mathbb{P}(d_{\lceil M(1-\alpha) \rceil} \leq x) = \sum_{k=\lceil M(1-\alpha) \rceil}^M \binom{M}{k} [F_d(x)]^k [1 - F_d(x)]^{M-k}, \quad (3.15)$$

where  $F_d$  denotes the true common underlying distribution of  $d_j$ ,  $j = 1, \dots, d_M$ . Since  $F_d$  is not known we replace it by the empirical distribution function implied by our sample  $\{d_1, \dots, d_M\}$ . This way we can estimate the  $100(\beta/2)\%$  and the  $100(1 - \beta/2)\%$  quantile for  $d_{\lceil M(1-\alpha) \rceil}$ , i.e.  $\hat{q}(\beta/2)$  and  $\hat{q}(1 - \beta/2)$ , respectively. If  $d_0$  lies outside the interval  $(\hat{q}(\beta/2), \hat{q}(1 - \beta/2))$  we know that it is significantly larger or smaller than  $d_{\lceil M(1-\alpha) \rceil}$ . In all applications of the test contained in this chapter we found that for  $\alpha = 5\%$  and  $\beta = 1\%$  a bootstrap sample size of  $M = 100$  was enough.

### Validity of the parametric bootstrap for the hypothesis test

In order to establish the validity of the parametric bootstrap for the above hypothesis test we will argue that under the null hypothesis  $H_0$  the bootstrapped distances  $d_j$ ,  $j = 1, \dots, M$ , are i.i.d. with a common distribution that is close to  $F_{d_0}$ , i.e. the distribution of  $d_0$ , for large sample size  $N$ : If we have a consistent estimator for  $\mathcal{R}^f$ , we know that under  $H_0$  the estimate  $\hat{\mathcal{R}}^f$  is close to  $\mathcal{R}^f$  for large  $N$ . Since the bootstrap samples  $\mathbf{u}_i^j$ ,  $i = 1, \dots, N$ ,  $j = 1, \dots, M$  are generated from  $\hat{\mathcal{R}}^f$ , they can be assumed to be approximate samples from  $\mathcal{R}^f$ . Since  $\hat{\mathcal{R}}_j^f$  and  $\hat{\mathcal{R}}_j^g$  are estimated based on the  $j$ th bootstrap sample  $\mathbf{u}_i^j$ ,  $i = 1, \dots, N$ , the KL between  $\hat{\mathcal{R}}_j^f$  and  $\hat{\mathcal{R}}_j^g$ , i.e.  $d_j$ , has the same distribution as the KL between  $\hat{\mathcal{R}}^f$  and  $\hat{\mathcal{R}}^g$ , i.e.  $d_0$ , for large  $N$ . Therefore, we can construct empirical confidence intervals for  $d_0$  based on the bootstrapped distances  $d_j$ ,  $j = 1, \dots, M$ .

Of course this argumentation is not a strict proof but rather makes the proposed approach plausible. An example for a mathematical justification of the parametric bootstrap in the copula context can be found in Genest and Rémillard (2008). In Section 3.1 of Killiches et al. (2017c) a simulation study can be found showing that our proposed test holds its level under the null hypothesis (for different sample sizes) when investigating the power of the test in a simplified/non-simplified vine copula framework.

## 3.5. Model selection

A typical application of model distance measures is model selection. Given a certain data set one often has to choose between several models with different complexity and features. Distance measures are a convenient tool that can help with the decision for the “best” or “most suitable” model out of a set of candidate models.

### 3.5.1. KL based model selection

The Kullback–Leibler distance is of particular interest for model selection because of the following relationship: For given copula data  $\mathbf{u}_i \in [0, 1]^d$ ,  $i = 1, \dots, N$ , from a  $d$ -dimensional copula model  $c: [0, 1]^d \rightarrow [0, \infty)$  we have

$$\text{KL}(c, c^\perp) \approx \sum_{i=1}^N \log \left( \frac{c(\mathbf{u}_i)}{c^\perp(\mathbf{u}_i)} \right) = \sum_{i=1}^N \log (c(\mathbf{u}_i)) = \log \ell(c),$$

where  $c^\perp$  denotes the density of the  $d$ -dimensional independence copula and  $\log \ell(c)$  is the log-likelihood of the model  $c$ . This means that the log-likelihood of a model can be approximated by calculating its Kullback–Leibler distance from the corresponding independence model (also known as *mutual information* in the bivariate case; see e.g. Cover and Thomas, 2012). The log-likelihood itself as well as the information criteria

AIC and BIC (Akaike, 1998; Schwarz, 1978), which are based on the log-likelihood but penalize the use of too many parameters, can be used to assess how well a certain model fits the data. The higher (lower) the log-likelihood (AIC/BIC) is, the better the model fit. Thus, a high Kullback–Leibler distance from the independence copula also corresponds to a good model fit. Note that this approximation only holds if data in fact was generated from  $c$ . Since in applications the true underlying distribution  $c$  is unknown, we can use the KL distance between a fitted copula and the independence copula as a proxy for the quality of the fit. Therefore, having fitted different models to a data set it is advisable to choose the one with the largest KL distance. Since dKL and sdKL are modifications of the original KL distance, it is natural to use them as substitutes for the model selection procedure.

In the following subsections we provide two examples, where dKL and sdKL based measures are applied for model selection. For this purpose we perform the following procedure 100 times: We fix a vine copula model and generate a sample of size  $N = 3000$  from it. Then, we fit different models on the generated sample and calculate the distance to the independence model with respect to dKL and sdKL, respectively. The results are compared to AIC and BIC.

### 3.5.2. Five-dimensional mixed vine

As a first example we consider a five-dimensional vine copula with the following vine tree structure and pair-copulas:

- Tree 1:  $c_{1,5}$  is a Gumbel copula with  $\tau_{1,5} = 0.6$ ,  $c_{2,4}$  is a BB1 copula with  $\tau_{2,4} = 0.83$ ,  $c_{3,4}$  is a BB7 copula with  $\tau_{3,4} = 0.74$  and  $c_{4,5}$  is a Tawn copula with  $\tau_{4,5} = 0.72$ ;
- Tree 2:  $c_{1,4;5}$  is a Clayton copula with  $\tau_{1,4;5} = 0.5$ ,  $c_{2,5;4}$  is a Joe copula with  $\tau_{2,5;4} = 0.45$  and  $c_{3,5;4}$  is a BB6 copula with  $\tau_{3,5;4} = 0.48$ ;
- Tree 3:  $c_{1,3;4,5}$  is a t copula with  $\tau_{1,3;4,5} = -0.19$  and  $\nu_{1,3;4,5} = 3$  degrees of freedom and  $c_{2,3;4,5}$  is a Frank copula with  $\tau_{2,3;4,5} = -0.31$ ;
- Tree 4:  $c_{1,2;3,4,5}$  is a Gaussian copula with  $\tau_{1,2;3,4,5} = -0.13$ .

As described above, we perform the following steps 100 times: Generate a sample of size  $N = 3000$  from the specified vine copula and fit four different models to the data sample (a Gaussian copula, a C-vine, a D-vine and an R-vine with the true structure) using the sequential fitting routine from `VineCopula`, where all available bivariate families are allowed as candidate pair-copulas. Table 3.11 displays the number of parameters, the dKL to the five-dimensional independence copula and the AIC and BIC values of the

four fitted models, all averaged over the 100 replications. The corresponding estimated standard errors are given in brackets.

	# par	dKL( $\cdot, c^\perp$ )	AIC	BIC
Gaussian copula	<b>10.00</b> (0.00)	6.24 (0.09)	-23669 (377)	-23609 (377)
C-vine	15.57 (0.64)	7.10 (0.06)	-28317 (333)	-28223 (333)
D-vine	19.77 (0.47)	7.41 (0.08)	-30320 (354)	-30201 (354)
R-vine	15.68 (0.78)	<b>8.37</b> (0.06)	<b>-33843</b> (344)	<b>-33749</b> (345)

**Table 3.11.:** Average number of parameters, dKL to the five-dimensional independence copula, AIC and BIC of the fitted Gaussian copula, C-vine, D-vine and R-vine (with true structure). Standard errors are given in brackets. Best values per column are marked in bold.

Compared to the 15 parameters of the true R-vine model, the Gaussian copula has only 10 parameters but also exhibits the poorest fit of all considered models with respect to any of the decision criteria. The C-vine (between 15 and 16 parameters on average) is ranked third by dKL, AIC and BIC. The D-vine model uses the most parameters (almost 20) but also performs better than the C-vine. With just under 16 parameters on average the R-vine copula is rated best by all three measures. We see that the ranking of the four fitted models is the same for dKL, AIC and BIC. We also checked that all 100 cases yielded this ranking. Considering the empirical ‘noise-to-signal’ ratio, i.e. the quotient of the standard errors and the absolute estimated mean, we obtain that the dKL performs better than AIC and BIC (e.g. for the R-vine we have  $0.06/8.37 < 344/33843 < 345/33479$ ).

### 3.5.3. 20-dimensional t vine

In order to show a high-dimensional example, we consider a 20-dimensional D-vine being also a t vine, i.e. a vine copula with only bivariate t copulas as pair copulas. The association parameter is chosen constant for all pair-copulas in one tree: Kendall’s  $\tau$  in Tree  $m$  is  $0.8^m$ ,  $m = 1, \dots, 19$ . Further, all pairs are heavy-tailed, having  $\nu = 3$  degrees of freedom. Due to the overall constant degrees of freedom the resulting t vine with its 380 parameters is not a t copula (cf. Section 2.2.3). Now we repeat the following procedure 100 times: Generate a sample of size  $N = 3000$  from the t vine with above specified parameters and fit a Gaussian copula, a t copula, a t vine and an R-vine to the simulated data. For the t and the R-vine we assume the true D-vine structure and use the sequential fitting routine from `VineCopula`, where for the R-vine all available bivariate families are allowed as candidate pair-copulas. Since the calculation of the dKL in  $d = 20$  dimensions would be rather time-consuming, we use the sdKL instead. We present the number of parameters, the sdKL to the 20-dimensional independence copula and the AIC and BIC values of the four fitted models (again averaged over the 100 replications) in Table 3.12. The estimated standard errors are given in brackets.

	# par	sdKL( $\cdot, c^\perp$ )	AIC	BIC
Gaussian copula	<b>190.00</b> (0.00)	84.11 (0.75)	-271610 (2962)	-270468 (2962)
t copula	191.00 (0.00)	93.95 (0.67)	-299703 (1929)	-298556 (1929)
t vine	380.00 (0.00)	96.72 (0.82)	<b>-309647</b> (2112)	<b>-307365</b> (2112)
R-vine	379.87 (0.60)	<b>96.80</b> (0.99)	-309337 (2579)	-307056 (2579)

**Table 3.12.:** Average number of parameters, sdKL to the 20-dimensional independence copula, AIC and BIC of the fitted Gaussian copula, t copula, t vine (with true structure) and R-vine (with true structure). Standard errors are given in brackets. Best values per column are marked in bold.

The Gaussian copula has the least parameters (190) but also the worst sdKL, AIC and BIC values. Adding a single additional parameter already causes an enormous improvement of all three measures for the t copula. The t vine is more flexible but has considerably more parameters than the t copula (380); nevertheless all three decision criteria prefer the t vine over the t copula. Surprisingly, the t vine is even ranked a little bit higher by AIC and BIC than the R-vine, which also has roughly 380 parameters on average. This ranking might seem illogical at first because the class of R-vines is a superset of the class of t vines such that one would expect the fit of the R-vine to be at least as good as the fit of the t vine. The reason for this alleged contradiction is that the fitting procedure that is implemented in the R package `VineCopula` (Schepsmeier et al., 2017) is not optimizing globally but tree-by-tree (cf. Section 2.2.5). Therefore, it is possible that fitting a non-t copula in one of the lower trees might be optimal but cause poorer fits in some of the higher trees. However, the difference between the fit of the t vine and the R-vine is very small and for 83 of the 100 samples both procedures fit the same model. Therefore, we want to test whether this difference is significant at all. For this purpose, we perform a parametric bootstrapping based test as described in Section 3.4 at the level  $\alpha = 5\%$  with  $M = 100$  replications. With p-values between 0.32 and 0.78 we cannot even reject the null hypothesis that the two underlying models coincide in any of the remaining 17 cases, where different models were fitted. Hence we would prefer to use the simpler t vine model which is in the same model class as the true underlying model. In a similar manner we check whether the difference between the t copula and the t vine is significant. Here, however, we find out that the model can indeed be distinguished at a 5% confidence level for all 100 samples (p-values range between 0 and 0.03). Considering the empirical noise-to-signal ratio we see that sdKL is a bit more dependent on the sample compared to dKL such that AIC, BIC and sdKL have roughly the same noise-to-signal ratio, where the values of AIC/BIC are slightly lower for the t copula, the t vine and the R-vine.

## 3.6. Determination of the optimal truncation level

As the dimension  $d$  of a vine copula increases, the number of parameters  $d(d-1)/2$  grows quadratically. For example, a 50-dimensional R-vine consists of 1225 (conditional) pair-copulas, each with one or more copula parameters. This on the one hand can create computational problems, while on the other hand the resulting model is difficult to interpret. Given an  $d$ -dimensional data set ( $d$  large), it has been proposed (see Brechmann et al., 2012; Brechmann and Joe, 2015) to fit a so-called  $k$ -truncated vine, where the pair-copulas of all trees of higher order than some truncation level  $k \leq d-1$  are modeled as independence copulas. This reduces the number of pair-copulas to be estimated from  $d(d-1)/2$  to  $d(d-1)/2 - (d-k)(d-k-1)/2$ , where  $k$  is chosen as small as can be justified. The heuristic behind this approach is that the sequential fitting procedure of regular vines captures most of the dependence in the lower trees, such that the dependence in the higher trees might be negligible and therefore the approximation error caused by using an independence copula is rather small. The task of finding the optimal truncation level  $k^*$  has already been tackled in the recent literature. Brechmann et al. (2012) use likelihood based criteria such as the AIC, BIC and Vuong test for the selection of  $k^*$ , while Brechmann and Joe (2015) propose an approach based on fit indices that measure the distance between fitted and observed correlation matrices.

### 3.6.1. Algorithms for the determination of optimal truncation levels

Using the proposed distance measures we can directly compare several truncated vines with different truncation levels. With the bootstrapped confidence intervals described in Section 3.4 we can assess whether the distances are significant in order to find the optimal truncation level. To be precise, in the following we present two algorithms that use the sdKL for the determination of the optimal truncation level, a global one (Algorithm 1) and a sequential one (Algorithm 2).

In Algorithm 1,  $\text{tRV}(k)$  denotes the  $k$ -truncated version of R-vine RV. Since a full  $d$ -dimensional R-vine consists of  $d-1$  trees,  $\text{tRV}(d-1)$  and RV coincide. The algorithm starts with the full model RV and, going backwards, truncates the vine tree-by-tree until the distance between the  $m$ -truncated vine and the full model is significantly larger than 0. Hence, the truncation at level  $m$  is too restrictive such that we select the level  $k^* = m+1$ , for which the distance was still insignificant. For the testing procedure we can use the test from Section 3.4 since the class of truncated vine copula models is nested in the general class of all vine copulas.

Since fitting a full vine copula model might be rather time-consuming in high dimensions with Algorithm 2 we propose another procedure of determining the truncation level, which

---

**Algorithm 1** Global determination of the optimal truncation level

---

**Input:**  $d$ -dimensional copula data, significance level  $\alpha$ .

- 1: Fit full (non-truncated) regular vine  $\text{RV} = \text{tRV}(d-1)$  to the data set.
- 2: **for**  $m = d - 2, \dots, 0$  **do**
- 3:     Specify the truncated vine  $\text{tRV}(m)$  by setting all pair-copulas of trees  $m + 1$  and higher to the independence copula.
- 4:     Calculate the sdKL between  $\text{RV}$  and  $\text{tRV}(m)$  and use the parametric bootstrap to check whether the distance is significantly different from zero.
- 5:     **if** distance is significant **then**
- 6:         **break** the **for**-loop and **return** the optimal truncation level  $k^* = m + 1$ .

**Output:** Optimal truncation level  $k^* = m + 1$ .

---

---

**Algorithm 2** Sequential determination of the optimal truncation level

---

**Input:**  $d$ -dimensional copula data, significance level  $\alpha$ .

- 1: Set  $\text{tRV}(0)$  to be a truncated vine with truncation level 0, i.e. an independence copula.
- 2: **for**  $m = 1, \dots, d - 1$  **do**
- 3:     Specify the truncated vine  $\text{tRV}(m)$  by taking the truncated vine from the previous step  $\text{tRV}(m - 1)$  and estimating the pair-copulas from tree  $m$ .
- 4:     Calculate the sdKL between  $\text{tRV}(m - 1)$  and  $\text{tRV}(m)$  and use the parametric bootstrap to check whether the distance is significantly different from zero.
- 5:     **if** distance is not significant **then**
- 6:         **break** the **for**-loop and **return** the optimal truncation level  $k^* = m - 1$

**Output:** Optimal truncation level  $k^* = m - 1$ .

---

builds the R-vine sequentially tree by tree, starting with the first tree. In each step we check whether the additionally modeled tree significantly changes the resulting model in comparison to the previous one. As long as it does, the vine is updated to one with an additionally modeled tree. Only when the addition of a new tree of order  $m$  results in a model that is statistically indistinguishable from the previous one, the algorithm stops and returns the optimal truncation level  $k^* = m - 1$ . The heuristic behind Algorithm 2 is that since the vine is estimated sequentially maximizing the sum of absolute (conditional) Kendall's  $\tau$  values in each tree (for details see Dißmann et al., 2013), we can expect the distance between two subsequent truncated vines to be decreasing. Therefore, if the distance between  $\text{tRV}(k^*)$  and  $\text{tRV}(k^* + 1)$  is not significant, the distances between  $\text{tRV}(m)$  and  $\text{tRV}(m + 1)$  for  $m > k^*$  should be not significant either.

Comparing Algorithm 1 and Algorithm 2, we note that in general they do not find the same truncation level. For example, consider the case where for some  $m$  the distances between  $\text{tRV}(m)$  and  $\text{tRV}(m + 1)$ ,  $\text{tRV}(m + 1)$  and  $\text{tRV}(m + 2)$ , until  $\text{tRV}(d - 2)$  and  $\text{tRV}(d - 1)$  are not significant, while the distance between  $\text{tRV}(m)$  and  $\text{tRV}(d - 1)$  is. Then, Algorithm 2 would return an optimal truncation level of  $m$ , whereas we would obtain a higher truncation level by Algorithm 1. So in general we see that Algorithm 2 finds more

parsimonious models than Algorithm 1.

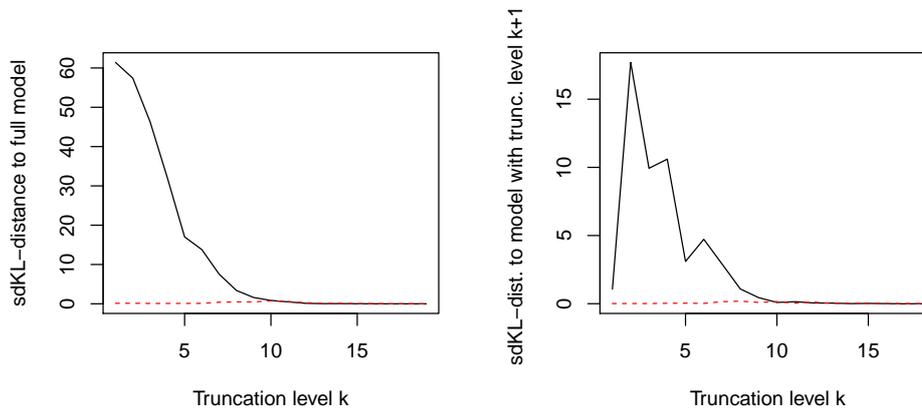
In the following we examine how well the proposed algorithms for finding optimal truncation levels for R-vines work in several simulated scenarios as well as real data examples. We compare our results to the existing methodology of Brechmann et al. (2012), who use a Vuong test (with and without AIC/BIC correction) to check whether there is a significant difference between a certain  $k$ -truncated vine and the corresponding vine with truncation level  $k - 1$ , for  $k = 1, \dots, d - 1$ . Starting with the lowest truncation levels, once the difference is not significant for some  $m$ , the algorithm stops and returns the optimal truncation level  $k^* = m - 1$ .

### 3.6.2. Simulation study

#### 20-dimensional t copula truncated at level 10

In the first simulated example, we consider a scenario where the data comes from a 20-dimensional t copula truncated at level 10. For this, we set the degrees of freedom to 3 and produce a random correlation matrix sampled from the uniform distribution on the space of correlation matrices according to the procedure of Joe (2006) described in Section 3.3.1, page 36. In this example, the resulting correlations range between  $-1$  and  $1$  with a higher concentration on correlations with small absolute values. After sampling the correlation matrix, we express the corresponding t copula as a D-vine (cf. Section 2.2.1) and truncate it at level 10, i.e. the pair-copulas of trees 11 to 19 are set to the independence copula. From this truncated D-vine we generate a sample of size  $N = 2000$  and use the R function `RVineStructureSelect` from the package `VineCopula` to fit a vine copula to the sample with the Dißmann algorithm (see Section 2.2.5). The question is now if our algorithms can detect the true truncation level underlying the generated data. For this we visualize the steps of the two algorithms. Concerning Algorithm 1, in the left panel of Figure 3.7 we plot the sdKL-distances between the truncated vines and the full (non-truncated) vine against the 19 truncation levels together with the bootstrapped upper 95% confidence bounds ( $d_{95}$  from Section 3.4) under the null hypothesis that the truncated vine coincides with the full model (dashed line).

Naturally, the curve corresponding to Algorithm 1 is decreasing with an extremely large distance between the one-truncated vine and the full model and a vanishingly small distance between the 18-truncated vine and the full model, which only differ in the specification of one pair-copula. In order to determine the smallest truncation level whose distance to the full model is insignificantly large, the algorithm compares these distances to the bootstrapped upper 95% confidence bounds. In this example we see that the smallest truncation level for which the sdKL-distance to the full model drops below the confidence bound is 10, such that the algorithm is able to detect the true truncation level. In order



**Figure 3.7.:** Visualization of the algorithms for data generated from a 20-dimensional  $t$  copula truncated at level 10. Left (Algorithm 1): sdKL-distance to full model with dashed bootstrapped upper 95% confidence bounds. Right (Algorithm 2): sdKL-distance to model with truncation level  $k + 1$  with dashed bootstrapped upper 95% confidence bounds.

to check, whether this was not just a coincidence we repeated this procedure 100 times and found that the optimal truncation level found by the algorithm averages to 10.5 with a standard deviation of 0.81.

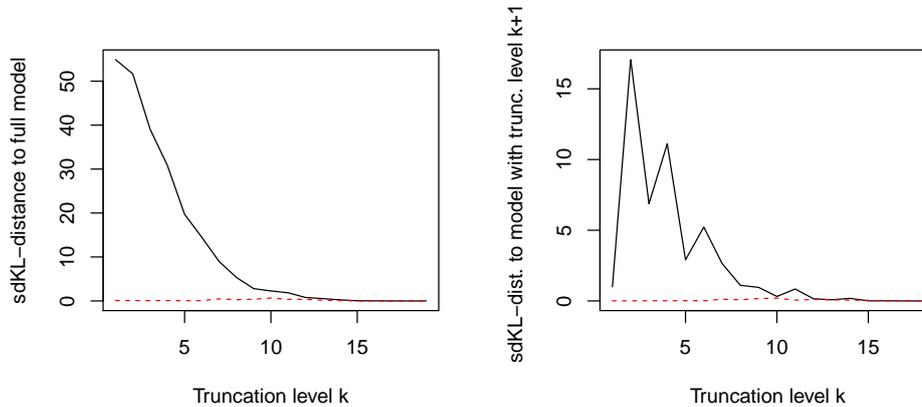
The right panel of Figure 3.7 displays the results for Algorithm 2. For each truncation level  $k$ , the sdKL-distance between the vine truncated at level  $k$  and the vine truncated at level  $k+1$  is plotted, again together with bootstrapped upper 95% confidence bounds under the null hypothesis that this distance is 0, i.e. the true model is the one with truncation level  $k$ . We observe that the largest sdKL-distance is given between the vine copulas truncated at levels 2 and 3, 3 and 4, and 4 and 5, respectively. This is in line with the results from Algorithm 1 (left panel of Figure 3.7), where we observe the steepest decrease in sdKL to the full model from truncation level 2 to 5. In this example Algorithm 2 would also detect the true truncation level 10. In the 100 simulated repetitions of this scenario, the average optimal truncation level was 10.2 with a standard deviation of 0.41.

In each of the 100 repetitions, we also used the Vuong test based algorithms without/with AIC/with BIC correction from Brechmann et al. (2012) to compare our results. They yielded average truncation levels of 14.6, 12.6 and 10.8 with standard deviations of 1.68, 1.52 and 0.85 (without/with AIC/with BIC correction), respectively, depending on the correction method. So all three methods overestimate the truncation level, in particular the first two.

Thus we have seen that in a scenario where the data is generated from a truncated vine both proposed algorithms manage to detect the truncation level very well. Next, we investigate how the algorithms perform when the true underlying copula is not truncated.

### 20-dimensional t copula (non-truncated)

In this example we generate data from the same 20-dimensional t copula as before, this time without truncating it. The results of the algorithms are displayed in Figure 3.8.



**Figure 3.8.:** Visualization of the algorithms for data generated from a 20-dimensional t copula (non-truncated). Left (Algorithm 1): sdKL-distance to full model with dashed bootstrapped upper 95% confidence bounds. Right (Algorithm 2): sdKL-distance to model with truncation level  $k + 1$  with dashed bootstrapped upper 95% confidence bounds.

At first sight the plots look quite similar to those of Figure 3.7. Due to the sequential fitting algorithm of Dißmann et al. (2013), which tries to capture large dependencies as early as possible (i.e. in the lower trees), the sdKL distance to the full model (left panel of Figure 3.8) is strongly decreasing in the truncation level. However, for truncation levels 10 to 15 this distance is still significantly different from zero (albeit very close to the upper 95% confidence bounds for  $k \geq 12$ ) such that the optimal truncation level is found to be 16. The right panel of Figure 3.8 tells us that the distance between the 11- and 12-truncated vine copulas is still fairly large and all subsequent distances between the  $k$ - and  $(k + 1)$ -truncated models are very small. However, Algorithm 2 also detects 16 to be the optimal truncation level because the distances are still slightly larger than the upper 95% confidence bounds for smaller  $k$ . In the 100 simulated repetitions the detected optimal truncation level was between 14 and 18 with an average of 16.2 for Algorithm 1 and 15.4 for Algorithm 2.

Again, we used the algorithms from Brechmann et al. (2012) in each of the 100 repetitions. From the different correction methods we obtained the following average truncation levels: 18.6, 18.3 and 17.6 with standard deviations 0.62, 0.94 and 1.32 (without/with AIC/with BIC correction), respectively.

Hence we can conclude that for vine copulas fitted by the Dißmann's algorithm our algorithms decide for a little more parsimonious models than the ones from Brechmann

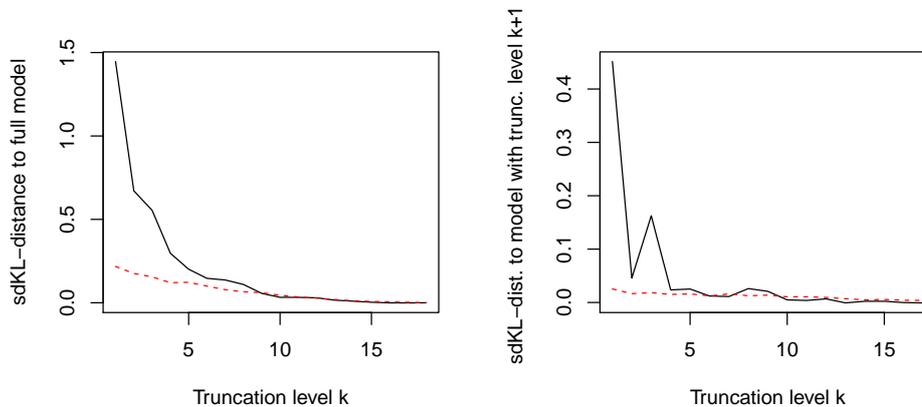
et al. (2012). This can even be desirable since Dißmann’s algorithm selects vines such that there is only little dependence in higher trees. Hence, we do not necessarily need to model all pair-copulas of the vine specification explicitly and a truncated vine often suffices.

### 3.6.3. Real data examples

Having seen that the algorithms seem to work properly for simulated data we now want to turn our attention to real data examples. First we revisit the example considered in Brechmann et al. (2012) concerning 19-dimensional Norwegian finance data.

#### 19-dimensional Norwegian finance data

The data set consists of 1107 observations of 19 financial quantities such as interest rates, exchange rates and financial indices for the period 2003–2008 (for more details refer to Brechmann et al., 2012). Figure 3.9 shows the visualization of the two algorithms for this data set.



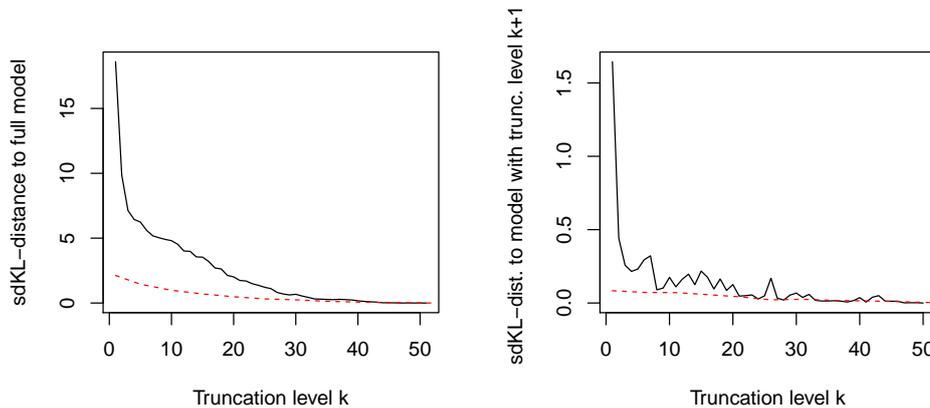
**Figure 3.9.:** Visualization of the algorithms for the 19-dimensional Norwegian finance data. Left (Algorithm 1): sdKL-distance to full model with dashed bootstrapped upper 95% confidence bounds. Right (Algorithm 2): sdKL-distance to model with truncation level  $k + 1$  with dashed bootstrapped upper 95% confidence bounds.

We see that the sdKL-distance to the full model is rapidly decreasing in the truncation level  $k$ , being quite close to the upper 95% confidence bound for  $k \geq 4$ , very close for  $k \geq 6$  and dropping below it for  $k = 10$ . Hence we can conclude that the optimal truncation level found by Algorithm 1 is 10, while a truncation level of 6 or even 4 may also be justified if one seeks more parsimonious models. This is exactly in line with the findings of Brechmann et al. (2012), who ascertained that depending on the favored degree of parsimony both truncation levels 4 and 6 may be justified. Yet, they find that there still

are significant dependencies beyond the sixth tree. This can also be seen from the right plot of Figure 3.9, which visualizes the results from Algorithm 2. We see that the distance between two subsequent truncated vines first falls below the upper 95% confidence bound for  $k = 6$ , after being close to it for  $k = 4$  and  $k = 5$ . Thus we see that in this example Algorithm 2 indeed finds a more parsimonious model than Algorithm 1. If we took the distances between all subsequent truncated vines into account, we would see that trees 9 and 10 still contribute significant dependencies, such that the “global” optimal truncation level again would be 10. If a data analyst decided that the parsimonious model truncated at level 6 or 4 would suffice for modeling this 19-dimensional data set, he or she would be able to reduce the number of pair-copulas to be modeled from 171 of the full model to 93 or 66, respectively, and thus greatly improve model interpretation and simplify further computations involving the model (e.g. value-at-risk simulations).

### 52-dimensional EuroStoxx50 data

Since the positive effect of truncating vine copulas intensifies with increasing dimensions, we revisit the EuroStoxx50 data set from Example 3.7 but this time we consider the entire 52-dimensional data set instead of only a four-dimensional subset. For risk managers it is a relevant task to correctly assess the interdependencies between these variables since they are included in most international banking portfolios. Figure 3.10 shows the results of the algorithms for this data set.



**Figure 3.10.:** Visualization of the algorithms for the 52-dimensional EuroStoxx50 data. Left (Algorithm 1): sdKL-distance to full model with dashed bootstrapped upper 95% confidence bounds. Right (Algorithm 2): sdKL-distance to model with truncation level  $k + 1$  with dashed bootstrapped upper 95% confidence bounds.

In the left panel we see that most of the dependence is captured by the first few trees since there the sdKL-distance to the full model has its sharpest decrease in truncation level  $k$ . The distance gets very close to the dashed upper 95% confidence bound for

$k > 27$ , however crossing it not before  $k = 43$ , implying a rather high truncation level. Considering the visualized results of Algorithm 2 in the right panel of Figure 3.10 we observe that the distances between subsequent truncated vines is quite small for  $k \geq 8$ , first dropping below the upper 95% confidence bound for  $k = 24$ . However, it increases again afterwards and ultimately drops below the confidence bound for  $k = 44$ . One could argue that a truncation level of  $k = 33$  might be advisable since the treewise distance slightly exceeds the confidence bound only three times thereafter. This would reduce the number of pair-copulas to be modeled from 1326 for the full 52-dimensional model to 1155 for the 33-truncated vine copula. Thus, with the help of model distances we can find simpler models for high-dimensional data.

For comparison, the algorithm from Brechmann et al. (2012) finds optimal truncation levels of 47 (without correction), 24 (AIC correction) and 3 (BIC correction). We see that there are large differences between the three methods: Whereas a truncation level of 47 corresponds almost to the non-truncated vine, one should be skeptical whether a 3-truncated vine is apt to describe the dependence structure of 52 random variables.

### 3.7. Conclusion

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

presented in this chapter showed the necessity of calculating distances between vine copulas for model selection. The developed parametric bootstrap based testing procedure proved to be particularly useful for finding an optimal truncation level for vines. Both algorithms proposed for determining suited truncation levels proved to yield satisfactory results in simulations and real data applications. Since Algorithm 2 tends to find more parsimonious models than Algorithm 1 it might be more suited for practical applications.

While we only considered datasets with dimensions  $d \leq 52$  here, applications in even higher dimensions are possible. With the theory developed in Müller and Czado (2016) the fitting of vines with hundreds of dimensions is facilitated with the focus on sparsity, i.e. fitting as many independence copulas as justifiable in order to reduce the number of parameters. The proposed distance measures can also be applied to select between several of these high-dimensional models.

# Chapter 4

---

## *Block maxima for vine copulas*

---

Large parts of the contents of this chapter have been published in Killiches and Czado (2015). This chapter is an edited reproduction of the published results.

### **4.1. Introduction**

Basically, block maxima have been used in extreme-value theory as one approach to derive the family of generalized extreme-value (GEV) distributions (McNeil et al., 2010). In the recent past the block maxima method has been studied more thoroughly and compared to the peaks-over-threshold (POT) method in Ferreira and de Haan (2014) and Jarušková and Hanek (2006). Dombry (2015) justifies the usage of a maximum-likelihood estimator for the extreme-value index within the block maxima framework. The numerical convergence of the block maxima approach to the GEV distribution is examined in Faranda et al. (2011). Moreover, the block maxima method has found its way into many application areas: Marty and Blanchet (2012) investigate long-term changes in annual maximum snow depth and snowfall in Switzerland. Temperature, precipitation, wind extremes over Europe are analyzed in Nikulin et al. (2011). A spatial application can be found in Naveau et al. (2009). Rocco (2014) provides an overview over the concepts of extreme-value theory being used in finance. While many of the articles use univariate concepts, Bücher and Segers (2014) treat how to estimate extreme-value copulas based on block maxima of a multivariate stationary time series. In contrast to the existing literature, in the following we will consider finite block maxima of multivariate random variables focusing on the dependence structure.

The remainder of the chapter will be structured as follows: In Section 4.2 we will derive an expression for the copula density of the joint distribution of finite block maxima of multivariate random variables. We show that this result is particularly useful when it is

applied to three-dimensional vine copulas in Section 4.3, where we also present numerical examples and an application to a hydrological data set. In order to mimic the approach of Hüsler and Reiss (1989) we further consider scaled block maxima in Section 4.4 and present numerical examples in Section 4.5. Finally, Section 4.6 concludes the chapter.

## 4.2. Copula density of the distribution of block maxima

Let  $\mathbf{U} = (U_1, \dots, U_d)^\top$  be a random vector with uniform(0,1)-distributed margins, copula  $C$  and copula density  $c$ . We consider  $n \in \mathbb{N}$  i.i.d. copies  $\mathbf{U}_i = (U_{i,1}, \dots, U_{i,d})^\top$  of  $\mathbf{U}$ ,  $i = 1, \dots, n$ . We are interested in the distribution of the vector of componentwise block maxima

$$\mathbf{M}^{(n)} = \left( M_1^{(n)}, \dots, M_d^{(n)} \right)^\top \quad \text{with } M_j^{(n)} := \max_{i=1, \dots, n} U_{i,j}$$

for  $j = 1, \dots, d$ . According to Sklar (1959) the dependence structure is determined by the corresponding copula  $C_{\mathbf{M}^{(n)}}$ . Since  $U_{i,j}$ ,  $i = 1, \dots, n$ , are i.i.d., we know that the marginal distribution functions of  $M_j^{(n)}$  are given by

$$F_j^{(n)}(m_j) = \mathbb{P}(U_{1,j} \leq m_j, \dots, U_{n,j} \leq m_j) = m_j^n \quad (4.1)$$

and hence the corresponding marginal densities are

$$f_j^{(n)}(m_j) = nm_j^{n-1} \quad (4.2)$$

for  $m_j \in [0, 1]$ ,  $j = 1, \dots, d$ . Thus, the copula  $C_{\mathbf{M}^{(n)}}$  is the distribution function of the random vector

$$\mathbf{V} = (V_1, \dots, V_d)^\top \quad \text{with } V_j := \left( M_j^{(n)} \right)^n \sim \text{uniform}(0, 1).$$

For  $n \in \mathbb{N}$  the copula of the componentwise maxima  $C_{\mathbf{M}^{(n)}}$  can be expressed in terms of the underlying copula  $C$  as follows

$$C_{\mathbf{M}^{(n)}}(u_1, \dots, u_d) = C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^n, \quad (4.3)$$

where  $u_j \in [0, 1]$ ,  $j = 1, \dots, d$  (see e.g. Genest and Nešlehová, 2012). Since  $C$  is assumed to have a density  $c$ , Equation 4.3 yields that  $C_{\mathbf{M}^{(n)}}$  also has a density, denoted by  $c_{\mathbf{M}^{(n)}}$ . Many statistical methods such as likelihood-based techniques rely on the density of a distribution such that  $c_{\mathbf{M}^{(n)}}$  is in fact of great interest. In Theorem 4.1 we provide an explicit formula for the copula density of the block maxima. The corresponding proof can be found in Appendix B.1.

**Theorem 4.1.** The density of the copula of the vector of block maxima satisfies for  $u_j \in [0, 1]$ ,  $j = 1, \dots, d$ :

$$c_{\mathbf{M}^{(n)}}(u_1, \dots, u_d) = \frac{1}{n^d} \left( \prod_{j=1}^d u_j \right)^{\frac{1}{n}-1} \sum_{j=1}^{d \wedge n} \left\{ \frac{n!}{(n-j)!} C(u_1^{1/n}, \dots, u_d^{1/n})^{n-j} \times \sum_{\mathcal{P} \in \mathcal{S}_{d,j}} \prod_{I \in \mathcal{P}} \partial_I C(u_1^{1/n}, \dots, u_d^{1/n}) \right\}. \quad (4.4)$$

Here,  $d \wedge n := \min\{d, n\}$  and  $\mathcal{S}_{d,j} := \{\mathcal{P} | \mathcal{P} \text{ partition of } \{1, \dots, d\} \text{ with } |\mathcal{P}| = j\}$  represents the set of all partitions consisting of  $j$  non-empty and disjoint subsets of  $\{1, \dots, d\}$ . Further,

$$\partial_I C(u_1^{1/n}, \dots, u_d^{1/n}) := \frac{\partial^p C(v_1, \dots, v_d)}{\partial v_{i_1} \cdots \partial v_{i_p}} \Big|_{v_1=u_1^{1/n}, \dots, v_d=u_d^{1/n}}$$

denotes the mixed partial derivative of the copula function  $C$  with respect to all indices contained in  $I = \{i_1, \dots, i_p\} \subseteq \{1, \dots, d\}$ .

We see that the copula density of the block maxima depends only on the copula  $C$  itself and its partial derivatives. Thus, we can determine the copula density of the block maxima for any block size  $n$  as soon as we know the underlying dependence structure  $C$ . This is a very convenient property when it comes to estimation: Assume you are given a sample  $\mathbf{u}_1, \dots, \mathbf{u}_N \in [0, 1]^d$  from  $C$  of size  $N$ . The natural approach to determine the copula density of the corresponding block maxima (for block size  $n$ ) would be to take the maxima over blocks of observations with size  $n$  and estimate the corresponding copula density. Thus, the estimation of the copula density of the block maxima is based on only  $\lfloor N/n \rfloor$  observations, where  $\lfloor \cdot \rfloor$  denotes the floor function. However, if we use Theorem 4.1 we are able use the entire sample for the estimation of  $C$ , which results in a more precise estimate. Further, we can directly derive the copula density of the block maxima for any block size  $n$ . This way it is even possible to determine  $c_{\mathbf{M}^{(n)}}$  for block sizes  $n > N$ , where we would not be able to perform the standard estimation approach since we do not even have a single observation of the block maxima. The only challenge is that we need to be able to determine the partial derivatives of the copula function  $C$ .

In order to illustrate the result from Theorem 4.1 we investigate the three-dimensional case as an example and determine the corresponding copula density of the vector of the block maxima.

**Example 4.2.** Let  $d = 3$ ,  $n \in \mathbb{N}$ , i.e.  $\mathbf{U}_i = (U_{i,1}, U_{i,2}, U_{i,3})^\top$  and  $\mathbf{M}^{(n)} = (M_1^{(n)}, M_2^{(n)}, M_3^{(n)})^\top$ ,  $i = 1, \dots, n$ . If  $n \geq 3$  the copula density of the vector of the block maxima is given by the

following expression:

$$\begin{aligned}
 c_{\mathbf{M}^{(n)}}(u_1, u_2, u_3) &= \frac{(u_1 u_2 u_3)^{\frac{1}{n}-1}}{n^3} \left\{ nC \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right)^{n-1} c \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right) \right. \\
 &\quad + n(n-1)C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right)^{n-2} \\
 &\quad \times \left[ \partial_1 C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right) \partial_{23} C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right) \right. \\
 &\quad + \partial_2 C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right) \partial_{13} C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right) \\
 &\quad \left. + \partial_3 C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right) \partial_{12} C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right) \right] \\
 &\quad + n(n-1)(n-2)C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right)^{n-3} \partial_1 C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right) \\
 &\quad \left. \times \partial_2 C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right) \partial_3 C \left( u_1^{1/n}, u_2^{1/n}, u_3^{1/n} \right) \right\}.
 \end{aligned}$$

Furthermore, we consider the block maxima of multivariate random variables with univariate standard normal margins since in Section 4.4 we want to establish a connection to the limiting approach used to derive the multivariate Hüsler–Reiss extreme-value distribution (Hüsler and Reiss, 1989). For this purpose, we apply the inverse probability integral transform to each component of  $\mathbf{U}_i$  to obtain marginally standard normally distributed data:

$$\mathbf{Z}_i = (Z_{i,1}, \dots, Z_{i,d})^\top \text{ with } Z_{i,j} := \Phi^{-1}(U_{i,j}) \sim \mathcal{N}(0, 1),$$

for  $i = 1, \dots, n, j = 1, \dots, d$ , where  $\Phi^{-1}$  is the inverse distribution function of the standard normal distribution  $\mathcal{N}(0, 1)$ . The corresponding componentwise block maxima are defined by

$$\tilde{\mathbf{M}}^{(n)} = \left( \tilde{M}_1^{(n)}, \dots, \tilde{M}_d^{(n)} \right)^\top \text{ with } \tilde{M}_j^{(n)} := \max_{i=1, \dots, n} Z_{i,j}$$

for  $i = 1, \dots, n, j = 1, \dots, d$ . Since we only perform a strictly increasing transformation of the margins the copula of  $\mathbf{Z}_i$  is the same as the one of  $\mathbf{U}_i$ , i.e.  $C$ . Similarly the copula of the block maxima over  $\mathbf{Z}_i$  is also given by  $C_{\mathbf{M}^{(n)}}$  with density  $c_{\mathbf{M}^{(n)}}$ . Note that this argument holds for any random variable with a strictly increasing distribution function. The marginal distributions of the componentwise block maxima are given by

$$G_j^{(n)}(m_j) = \mathbb{P}(\Phi(Z_{1,j}) \leq m_j, \dots, \Phi(Z_{n,j}) \leq m_j) = \Phi(m_j)^n \quad (4.5)$$

with corresponding marginal densities

$$g_j^{(n)}(m_j) = n\Phi(m_j)^{n-1}\varphi(m_j). \quad (4.6)$$

for  $m_j \in \mathbb{R}, j = 1, \dots, d$ . Here  $\Phi$  and  $\varphi$  denote the distribution function and the density of the standard normal distribution, respectively. Thus, the joint distribution function of

the block maxima over  $\mathbf{Z}_i$  is given by

$$G^{(n)}(m_1, \dots, m_d) = C_{\mathbf{M}^{(n)}} \left( G_1^{(n)}(m_1), \dots, G_d^{(n)}(m_d) \right) = C(\Phi(m_1), \dots, \Phi(m_d))^n.$$

For the corresponding joint density  $g^{(n)}$  of the block maxima over  $\mathbf{Z}_i$  we also obtain an explicit expression as a corollary of Theorem 4.1, where once again Sklar's Theorem is applied.

**Corollary 4.3.** For  $m_j \in \mathbb{R}$ ,  $j = 1, \dots, d$ , we have

$$g^{(n)}(m_1, \dots, m_d) = \left( \prod_{j=1}^d \varphi(m_j) \right) \cdot \sum_{j=1}^{d \wedge n} \left\{ \frac{n!}{(n-j)!} \cdot C(\Phi(m_1), \dots, \Phi(m_d))^{n-j} \right. \\ \left. \times \sum_{\mathcal{P} \in \mathcal{S}_{d,j}} \prod_{I \in \mathcal{P}} \partial_I C(\Phi(m_1), \dots, \Phi(m_d)) \right\}. \quad (4.7)$$

### 4.3. Application to three-dimensional vine copulas

As an application of the results in the previous section we will consider vine copulas in three dimensions and show why they are a particularly useful class of model for this purpose. We have seen in Chapter 2 that vine copulas allow for product expressions of the density. Since in three dimensions all (three) structures are equivalent in the sense that they can be obtained by relabeling the variables, we only consider the following decomposition for the remainder of the chapter:

$$c(u_1, u_2, u_3) = c_{1,2}(u_1, u_2) c_{2,3}(u_2, u_3) c_{1,3;2}(C_{1|2}(u_1|u_2), C_{3|2}(u_3|u_2); u_2). \quad (4.8)$$

We have seen that we need the copula function  $C$  and its partial derivatives in order to calculate the copula density of the block maxima using in Theorem 4.1. These expressions are derived in Proposition 4.4. The corresponding proof can be found in Appendix B.2.

**Proposition 4.4.** For a three-dimensional vine copula with decomposition as in Equation 4.8 the partial derivatives of the copula function can be obtained as follows:

1.  $C(u_1, u_2, u_3) = \int_0^{u_2} C_{1,3;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) dv_2;$
2. a)  $\partial_1 C(u_1, u_2, u_3) = \int_0^{u_2} C_{3|1;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) c_{1,2}(u_1, v_2) dv_2;$   
b)  $\partial_2 C(u_1, u_2, u_3) = C_{1,3;2}(C_{1|2}(u_1|u_2), C_{3|2}(u_3|u_2); u_2);$   
c)  $\partial_3 C(u_1, u_2, u_3) = \int_0^{u_2} C_{1|3;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) c_{2,3}(v_2, u_3) dv_2;$
3. a)  $\partial_{12} C(u_1, u_2, u_3) = C_{3|1;2}(C_{1|2}(u_1|u_2), C_{3|2}(u_3|u_2); u_2) c_{1,2}(u_1, u_2);$

$$\text{b) } \partial_{13}C(u_1, u_2, u_3) = \int_0^{u_2} c_{1,3;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) c_{2,3}(v_2, u_3) c_{1,2}(u_1, v_2) dv_2;$$

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

$$4. \ c(u_1, u_2, u_3) = c_{1,2}(u_1, u_2) c_{2,3}(u_2, u_3) c_{1,3;2}(C_{1|2}(u_1|u_2), C_{3|2}(u_3|u_2); u_2).$$

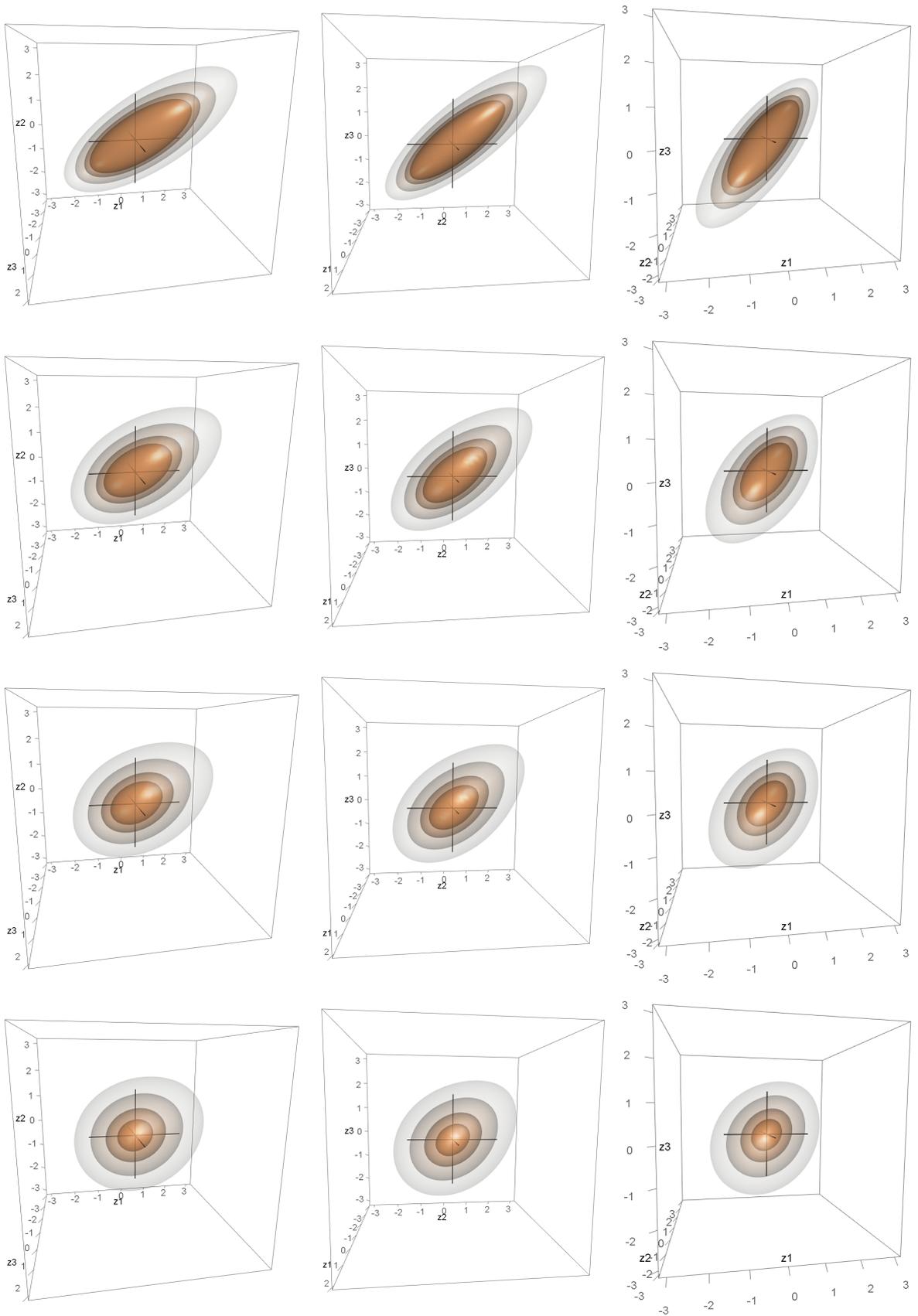
Proposition 4.4 shows that the copula density corresponding to the three-dimensional vector of block maxima based on an arbitrary vine copula is numerically tractable since only one-dimensional integration is required. In particular this allows a numerical treatment for the block size  $n$  in a finite setting.

In the following three examples we will work with simplified vine copulas. In order to illustrate the influence of the block size on the dependence structure of the block maxima we consider the corresponding copula density  $c_{\mathbf{M}(n)}$  for block sizes  $n = 1, 10, 50, 1000$ . Note that for a block size of  $n = 1$  the copula density of the block maxima simply is the copula density  $c$  of the underlying distribution, i.e.  $c_{\mathbf{M}(1)} = c$ , since taking maxima over blocks of size one does not have any impact. To visualize the copula density of the block maxima  $c_{\mathbf{M}(n)}$  for different block sizes  $n$  we consider the contour surfaces of these three-dimensional copula densities. This approach is the natural extension of considering contour lines for bivariate copulas. Since copula densities on the original  $[0,1]$ -scale are difficult to interpret and comparisons between different models would hardly be possible one usually uses univariate standard normal margins when considering contour plots. Therefore, we also use univariate standard normal margins for our three-dimensional contour surface plots. The contour levels are fixed to be 0.015, 0.035, 0.075 and 0.110 for each plot (from outer to inner surface) in order to enable comparisons between different block sizes and examples. We show the contour surfaces from three different angles such that we get a representative impression of the shape of the contours. This way of illustrating three-dimensional copula densities is inspired by Killiches et al. (2017a). Furthermore, we apply our approach to a three-dimensional hydrological data set.

**Example 4.5.** The first example we present is a three-dimensional Gaussian vine, i.e. all three pair-copulas are bivariate Gaussian copulas. As parameters we choose  $\rho_{1,2} = 0.71$ ,  $\rho_{2,3} = 0.78$  and  $\rho_{1,3;2} = 0.52$  corresponding to Kendall's  $\tau$  values of  $\tau_{1,2} = 0.50$  and  $\tau_{2,3} = 0.57$ ,  $\tau_{1,3;2} = 0.35$ .

Figure 4.1 shows the copula density of the block maxima of this vine copula with standard normal margins for block sizes  $n = 1, 10, 50, 1000$ . Each row represents one block size and contains three plots from different angles. We see that for increasing block size the contours tend to concentric spheres around the origin. This is what the contour surfaces of the independence copula (with standard normal margins) look like in three dimensions. This is also what one would expect: Recall from Section 2.2.3 that every Gaussian vine is also a Gaussian copula. Sibuya (1960) proved that any Gaussian copula

### 4.3. Application to three-dimensional vine copulas



**Figure 4.1.:** Plot of the contour surfaces of the copula density of block maxima of a three-dimensional Gaussian vine with standard normal margins ( $\tau_{1,2} = 0.50$ ,  $\tau_{2,3} = 0.57$ ,  $\tau_{1,3;2} = 0.35$ ) from different angles (columns). The rows correspond to block sizes  $n = 1, 10, 50$  and  $1000$ .

lies in the domain of attraction of the independence copula, i.e. the block maxima of any Gaussian copula converge to the independence copula. Thus, our empirical findings are in line with the theory. Hüsler and Reiss (1989) showed that in order to achieve that the distribution of the block maxima of a multivariate Gaussian distribution converges to a non-trivial limiting distribution, a proper scaling of the margins and the correlation coefficients is necessary. This will be discussed in Section 4.4.

**Example 4.6.** As a second example we take a three-dimensional Clayton vine, i.e. all three pair-copulas are now bivariate Clayton copulas. As parameters we choose  $\delta_{1,2} = 6$ ,  $\delta_{2,3} = 7.09$  and  $\delta_{1,3;2} = 4.67$  corresponding to Kendall's  $\tau$  values of  $\tau_{1,2} = 0.75$  and  $\tau_{2,3} = 0.78$ ,  $\tau_{1,3;2} = 0.70$ .

Similar to the Gaussian example Figure 4.2 shows the copula density of the block maxima of this vine copula with standard normal margins for block sizes  $n = 1, 10, 50, 1000$ . Each row represents one block size and contains three plots from different angles. We see that for increasing block size the contours tend to concentric spheres around the origin corresponding to the contour surfaces of the independence copula, which are already (approximately) reached for  $n = 1000$ .

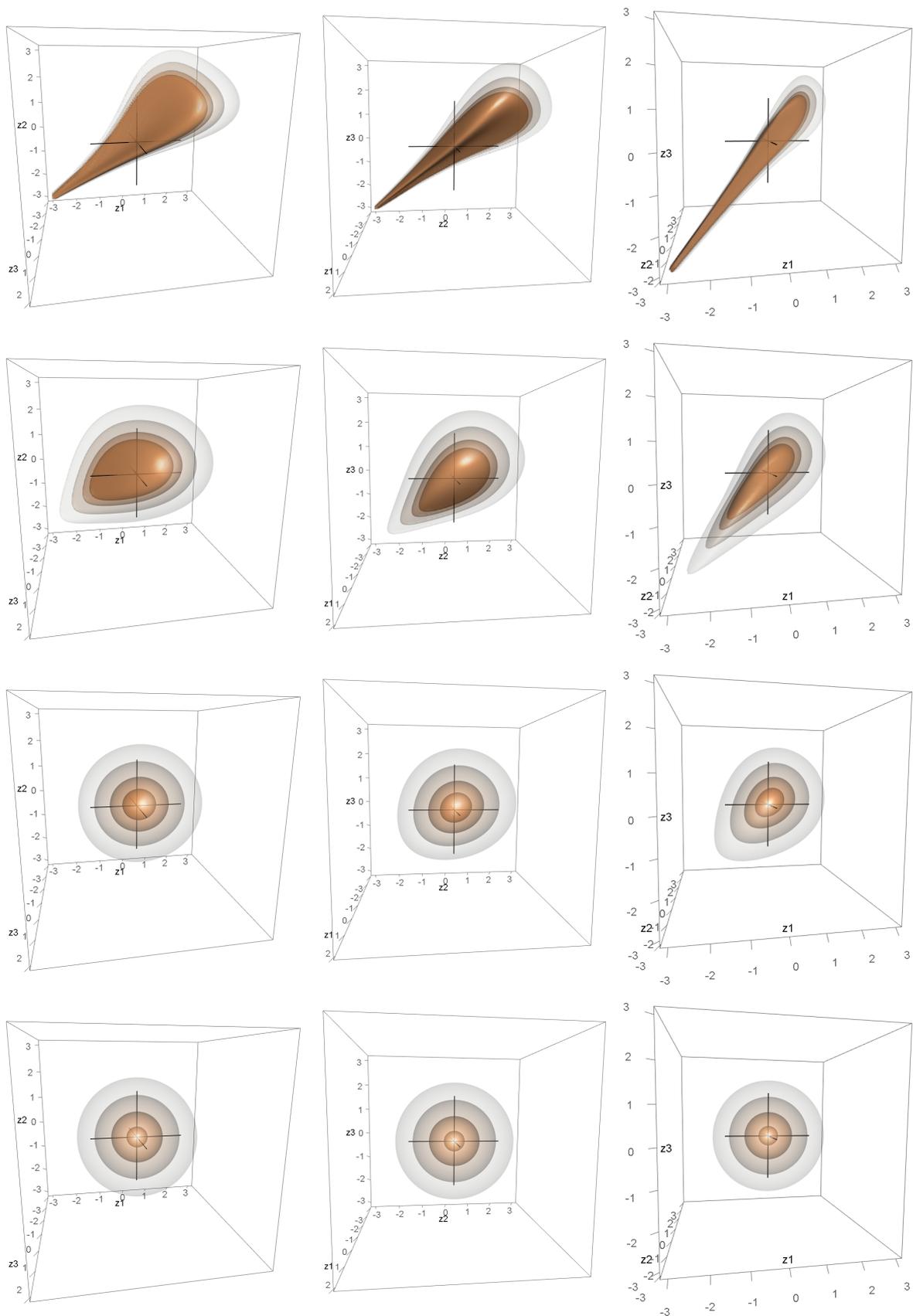
**Remark 4.7.** Even though it is not known whether all Clayton vines lie in the domain of attraction of the independence copula, one can show that the Clayton copula, which can be represented as a Clayton vine with specific parameter restrictions (cf. Section 2.2.3), lies in the domain of attraction of the independence copula. According to Gudendorf and Segers (2010) an Archimedean copula with generator  $\varphi$  lies in the domain of attraction of the Gumbel copula with parameter

$$\theta := -\lim_{s \downarrow 0} \frac{s\varphi'(1-s)}{\varphi(1-s)} \in [1, \infty)$$

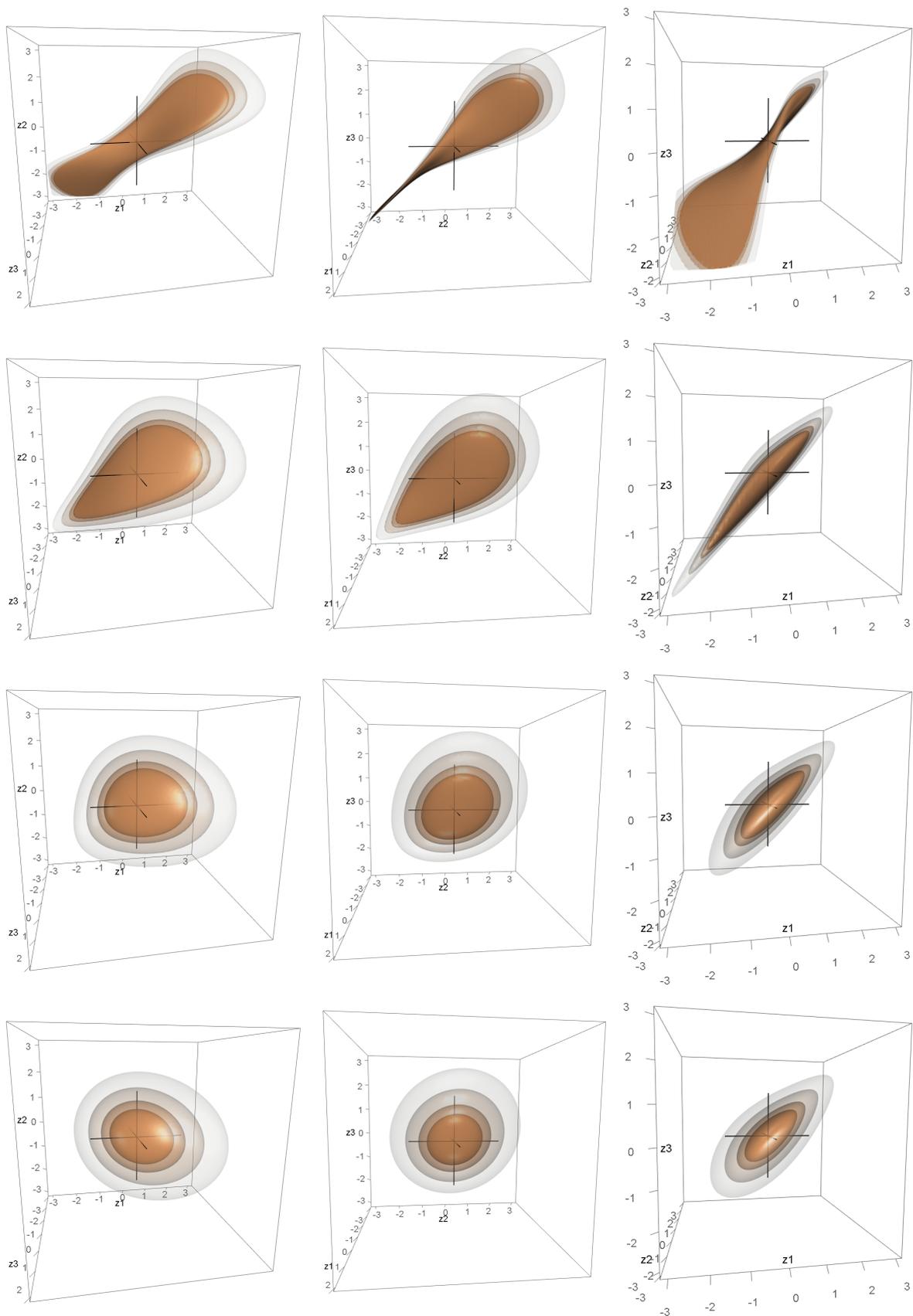
if the limit exists. For the Clayton copula this limit is equal to 1. Therefore, the copula of the block maxima of a Clayton copula converges to the Gumbel copula with  $\theta = 1$ , which corresponds to the independence copula.

**Example 4.8.** Having investigated two examples where all three pair-copulas belonged to the same family we will now consider a mixed vine copula with very high strengths of dependence specified as follows:  $c_{1,2}$  is a Frank copula with parameter 8 ( $\tau_{1,2} = 0.8$ ),  $c_{2,3}$  is a Clayton copula with parameter 18.19 ( $\tau_{2,3} = 0.8$ ) and  $c_{1,3;2}$  is a Gaussian copula with parameter 0.95 ( $\tau_{1,3;2} = 0.8$ ). As before we show the copula density of the block maxima of this vine copula with standard normally distributed margins for block sizes  $n = 1, 10, 50, 1000$  in Figure 4.3. Each row represents one block size and contains three plots from different angles. We see that due to the high parameter values there can still be detected some non-negligible dependence for block sizes  $n = 50$  and even  $n = 1000$ .

### 4.3. Application to three-dimensional vine copulas



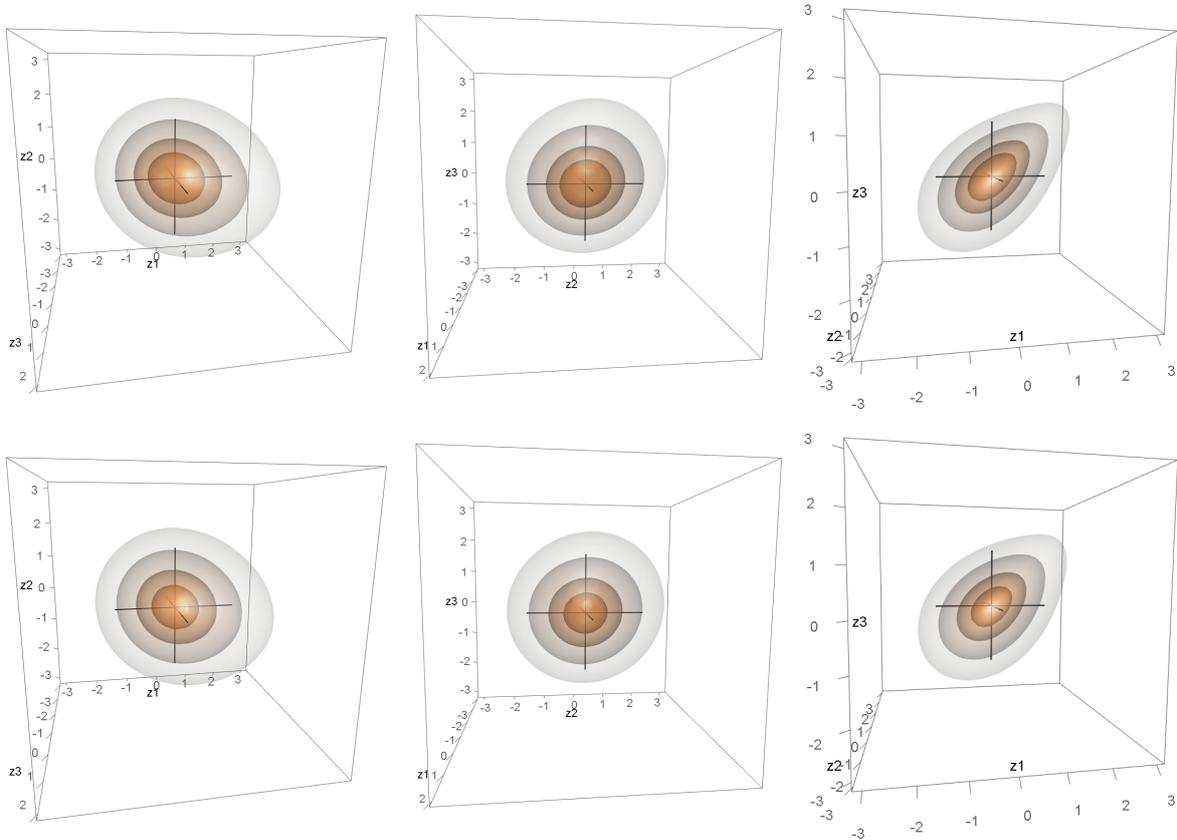
**Figure 4.2.:** Plot of the contour surfaces of the copula density of block maxima of a three-dimensional Clayton vine with standard normal margins ( $\tau_{1,2} = 0.75$ ,  $\tau_{2,3} = 0.78$ ,  $\tau_{1,3;2} = 0.70$ ) from different angles (columns). The rows correspond to block sizes  $n = 1, 10, 50$  and  $1000$ .



**Figure 4.3.:** Plot of the contour surfaces of the copula density of block maxima of a three-dimensional Frank-Clayton-Gaussian vine with standard normal margins ( $\tau_{1,2} = 0.80$ ,  $\tau_{2,3} = 0.80$ ,  $\tau_{1,3;2} = 0.80$ ) from different angles (columns). The rows correspond to block sizes  $n = 1, 10, 50$  and  $1000$ .

### 4.3. Application to three-dimensional vine copulas

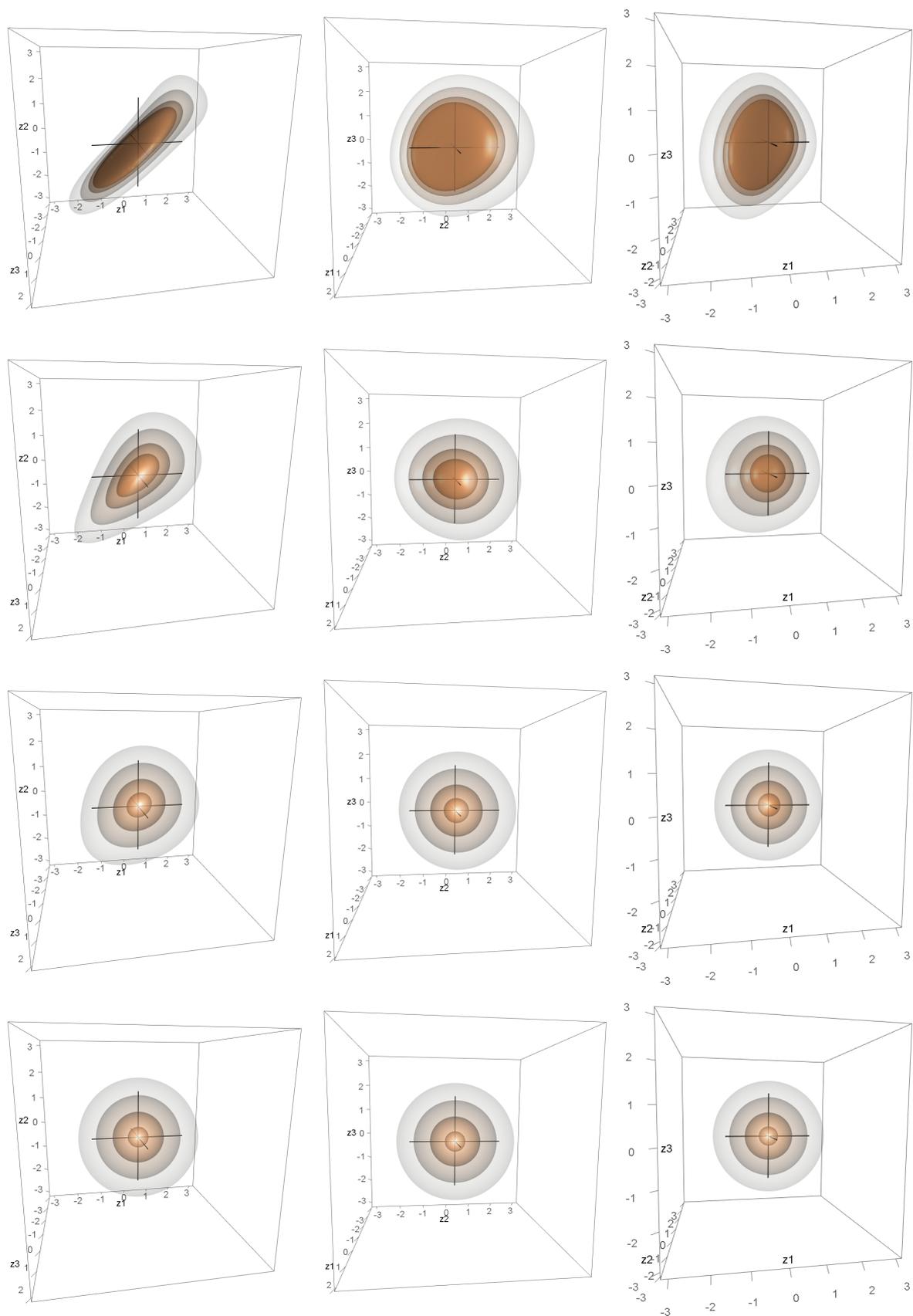
Therefore, we further increased block sizes to  $n = 10^6$  and  $n = 10^8$  in Figure 4.4. Although there is still some dependence left, as before we can see the clear tendency to the contours of the independence copula.



**Figure 4.4.:** Plot of the contour surfaces of the copula density of block maxima of a three-dimensional Frank-Clayton-Gaussian vine with standard normal margins ( $\tau_{1,2} = 0.80$ ,  $\tau_{2,3} = 0.80$ ,  $\tau_{1,3;2} = 0.80$ ) from different angles (columns). The rows correspond to block sizes  $n = 10^6, 10^8$ .

**Example 4.9.** Hydrology is one of the areas where block maxima are important. Especially, the water levels of rivers can be interesting when it comes to analyzing the risk of floods. We consider a three-dimensional data set containing the water levels of rivers in and around Munich, Germany, from August 1, 2007 to July 31, 2013. The data has been taken from Bavarian Hydrological Service (<http://www.gkd.bayern.de>). The three variables denote the differences of the 12 hour average water levels at the following three measuring points: the Isar measured in Munich, the Isar measured in Baierbrunn (south of Munich) and the Schwabinger Bach measured in Munich (a small stream entering the Isar in Garching, north of Munich). Since we only consider the hydrological winter (November 1 to April 30), we have 2176 data points.

First, we transform the margins to the unit interval applying the probability integral transform with the empirical marginal distribution functions. Then, we fit the dependence



**Figure 4.5.:** Plot of the contour surfaces of the copula density for block maxima of the water level differences for one day (first row), one week (second row), one month (third row) and one winter (fourth row) with standard normal margins.

structure with a vine copula using `RVineStructureSelect`<sup>5</sup>:  $c_{1,2}$  is estimated to be a Frank copula with a Kendall's  $\tau$  of  $\tau_{1,2} = 0.76$ ,  $c_{2,3}$  is a Frank copula with  $\tau_{2,3} = 0.23$  and  $c_{1,3;2}$  is a Gaussian copula with  $\tau_{1,3;2} = -0.18$ . Now we are interested in the resulting copula density of the maxima for one day ( $n = 2$ ), one week ( $n = 14$ ), one month ( $n = 60$ ) and one winter ( $n = 362$ ). The corresponding contour surfaces are plotted in Figure 4.5. Similar to the examples from above we see that with increasing  $n$  the observed dependence structure tends to the independence copula. In case of the considered rivers this means that the maximal differences of the 12 hour average water levels over the entire winter are (almost) independent.

## 4.4. Copula density of scaled block maxima

The examples from Section 4.3 all seemed to converge to the independence copula for block sizes going to infinity. Therefore, we will investigate if some sort of scaling of the block maxima can help to achieve non-trivial limiting copulas. Such limiting copulas are called extreme value copulas and are characterized by max-stability. A recent introduction to extreme value copulas is given by Gudendorf and Segers (2010).

Since Hüsler and Reiss (1989) derived the scaling for the block maxima of the multivariate normal distribution with standard normally distributed margins to a non-trivial extreme value copula, we use the same marginal scaling for the block maxima  $\tilde{\mathbf{M}}^{(n)}$ . They are given by

$$W_j^{(n)} := b_n \left( \tilde{M}_j^{(n)} - b_n \right),$$

where  $b_n$  satisfies  $b_n = n \cdot \varphi(b_n)$  for the standard normal density  $\varphi$ . Univariate extreme value theory gives that

$$F_{W_j^{(n)}}(w_j) = P \left( W_j^{(n)} \leq w_j \right) = \Phi^n \left( b_n + \frac{w_j}{b_n} \right) \rightarrow \exp\{-\exp\{w_j\}\} \text{ as } n \rightarrow \infty$$

for  $w_j \in \mathbb{R}$ . The marginal density of  $W_j^{(n)}$  is given by

$$f_{W_j^{(n)}}(w_j) = \frac{n}{b_n} \Phi^{n-1} \left( b_n + \frac{w_j}{b_n} \right) \varphi \left( b_n + \frac{w_j}{b_n} \right) \quad (4.9)$$

for  $w_j \in \mathbb{R}$ ,  $j = 1, \dots, d$ . Since  $W_j^{(n)}$  is a strictly increasing transformation of  $\tilde{M}_j^{(n)}$ , the copula of  $\mathbf{W}^{(n)} := \left( W_1^{(n)}, \dots, W_d^{(n)} \right)$  is the same as the one of  $\tilde{\mathbf{M}}^{(n)}$ , which is given by  $C_{\tilde{\mathbf{M}}^{(n)}}$ . Therefore, using Equation 4.3 we obtain the following expression for the joint

---

<sup>5</sup>In order to assure that the necessary integrals were numerically tractable we had to exclude some pair-copula families (e.g. the t copula).

distribution of  $\mathbf{W}^{(n)}$ :

$$\begin{aligned} F_{\mathbf{W}^{(n)}}(w_1, \dots, w_d) &= P\left(W_1^{(n)} \leq w_1, \dots, W_d^{(n)} \leq w_d\right) \\ &= C_{\mathbf{M}^{(n)}}\left(\Phi^n\left(b_n + \frac{w_1}{b_n}\right), \dots, \Phi^n\left(b_n + \frac{w_d}{b_n}\right)\right) \\ &= \left[C\left(\Phi\left(b_n + \frac{w_1}{b_n}\right), \dots, \Phi\left(b_n + \frac{w_d}{b_n}\right)\right)\right]^n. \end{aligned}$$

Similar arguments as in Corollary 4.3 can be used to express the joint density of  $\mathbf{W}^{(n)}$  in three dimensions for  $n \geq 3$  as

$$\begin{aligned} f_{\mathbf{W}^{(n)}}(w_1, w_2, w_3) &= \frac{1}{b_n^3} \prod_{j=1}^3 \varphi\left(b_n + \frac{w_j}{b_n}\right) \left\{ nC(u_1, u_2, u_3)^{n-1} c(u_1, u_2, u_3) \right. \\ &\quad + n(n-1)C(u_1, u_2, u_3)^{n-2} \\ &\quad \times \left[ \partial_1 C(u_1, u_2, u_3) \partial_{23} C(u_1, u_2, u_3) \right. \\ &\quad \quad + \partial_2 C(u_1, u_2, u_3) \partial_{13} C(u_1, u_2, u_3) \\ &\quad \quad \left. + \partial_3 C(u_1, u_2, u_3) \partial_{12} C(u_1, u_2, u_3) \right] \\ &\quad \left. + n(n-1)(n-2)C(u_1, u_2, u_3)^{n-3} \partial_1 C(u_1, u_2, u_3) \right. \\ &\quad \quad \left. \times \partial_2 C(u_1, u_2, u_3) \partial_3 C(u_1, u_2, u_3) \right\}, \end{aligned} \tag{4.10}$$

where  $u_j := \Phi\left(b_n + \frac{w_j}{b_n}\right)$  for  $j = 1, 2, 3$ .

According to Hüsler and Reiss (1989), besides scaling the maxima  $M_1^{(n)}, \dots, M_d^{(n)}$ , it is also necessary to change the correlation matrix  $\Sigma(n) = (\rho_{i,j}(n))_{1 \leq i, j \leq d}$  of the underlying joint distribution of standard normal random variables  $Z_1, \dots, Z_n$ , over whose i.i.d. copies  $Z_{i,1}, \dots, Z_{i,d}$ ,  $i = 1, \dots, n$ , we take the maximum. These correlation matrices  $\Sigma(n)$  need to satisfy the following condition

$$(1 - \rho_{i,j}(n)) \log(n) \rightarrow \lambda_{i,j}^2 \text{ as } n \rightarrow \infty, \tag{4.11}$$

where  $\lambda_{i,j} \in (0, \infty)$  are some constants for  $1 \leq i, j \leq d$ ,  $i \neq j$  and  $\lambda_{i,i} = 0$  for  $i = 1, \dots, d$ . Since  $\rho_{i,j}(n) = \rho_{j,i}(n)$ , we also have  $\lambda_{i,j} = \lambda_{j,i}$  for  $1 \leq i, j \leq d$ . Note that Equation 4.11 implies that  $\rho_{i,j}(n) \rightarrow 1$  as  $n \rightarrow \infty$ . The limiting distribution  $H_\Lambda$  of the scaled maxima depends on  $\Lambda := (\lambda_{i,j})_{1 \leq i, j \leq d}$ .

In the following we will focus on the three-dimensional case and investigate the influence

of  $\lambda_{1,2}, \lambda_{1,3}, \lambda_{2,3} \in (0, \infty)$ . For simplicity, we set

$$\rho_{i,j}(n) := 1 - \frac{\lambda_{i,j}^2}{\log(n)} \quad (4.12)$$

for  $1 \leq i, j \leq 3$ ,  $n \in \mathbb{N}$ , such that Equation 4.11 is always satisfied. However, for arbitrary  $\lambda_{i,j}$  it is not trivial to decide whether we obtain a valid correlation matrix through this particular choice of  $\rho_{i,j}(n)$  for any  $n \in \mathbb{N}$ . By construction the matrices

$$\Sigma(n) = \begin{pmatrix} 1 & \rho_{1,2}(n) & \rho_{1,3}(n) \\ \rho_{1,2}(n) & 1 & \rho_{2,3}(n) \\ \rho_{1,3}(n) & \rho_{2,3}(n) & 1 \end{pmatrix}$$

are symmetric and have ones on their diagonals. Thus, the only property we have to verify is if  $\Sigma(n)$  is positive definite. For this we only need to check if the determinant of each leading principal minor is positive. Since  $1 > 0$  and  $1 - \rho_{1,2}(n)^2 > 0$  are trivially satisfied, the only real requirement is that

$$|\Sigma(n)| = 1 - \rho_{1,2}(n)\rho_{1,3}(n)\rho_{2,3}(n) - \rho_{1,2}(n)^2 - \rho_{1,3}(n)^2 - \rho_{2,3}(n)^2 > 0.$$

Using Equation 4.12 we obtain that  $|\Sigma(n)| > 0$  if and only if the following condition is satisfied:

$$2(\lambda_{1,2}^2\lambda_{1,3}^2 + \lambda_{1,2}^2\lambda_{2,3}^2 + \lambda_{1,3}^2\lambda_{2,3}^2) - (\lambda_{1,2}^4 + \lambda_{1,3}^4 + \lambda_{2,3}^4) > \frac{2\lambda_{1,2}^2\lambda_{1,3}^2\lambda_{2,3}^2}{\log(n)}. \quad (4.13)$$

We denote the left-hand side of Equation 4.13 by  $h(\lambda_{1,2}^2, \lambda_{1,3}^2, \lambda_{2,3}^2)$ . Since the right-hand side of Equation 4.13 is always positive, the condition can only be satisfied if also the left-hand side of Equation 4.13 is positive, i.e.  $h(\lambda_{1,2}^2, \lambda_{1,3}^2, \lambda_{2,3}^2) > 0$ . If this is the case, then Equation 4.13 is satisfied for all  $n \in \mathbb{N}$  with

$$n \geq n^* := \left\lfloor \exp \left\{ \frac{2\lambda_{1,2}^2\lambda_{1,3}^2\lambda_{2,3}^2}{h(\lambda_{1,2}^2, \lambda_{1,3}^2, \lambda_{2,3}^2)} \right\} + 1 \right\rfloor,$$

where  $\lfloor \cdot \rfloor$  denotes the floor function. Table 4.1 shows the values of  $h$  and  $n^*$  (if existing) for 10 different combinations of  $\lambda_{1,2}, \lambda_{1,3}, \lambda_{2,3}$ . Later we will use combinations 9 and 10 for numerical examples.

Hüsler and Reiss (1989) derived this scaling for multivariate normal distributions. Since we want to apply the scaling to vine copulas, we need to transform the parameters of the normal distribution (correlations) to the parameters of the vine. This will be done with the help of Kendall's  $\tau$ .

#	$\lambda_{1,2}^2$	$\lambda_{1,3}^2$	$\lambda_{2,3}^2$	$h(\lambda_{1,2}^2, \lambda_{1,3}^2, \lambda_{2,3}^2)$	$n^*$
1	1	1	1	3	2
2	2	2	2	12	4
3	1	2	3	8	5
4	0.5	0.5	0.5	0.75	2
5	0.3	0.2	0.1	0.08	2
6	0.2	5	0.75	-15.8	–
7	15	20	15	800	76880
8	100	0.1	20	-6376.01	–
9	1.05	0.21	0.84	0.71	2
10	4	3	3	32	10

**Table 4.1.:** Different combinations  $\lambda_{1,2}$ ,  $\lambda_{1,3}$  and  $\lambda_{2,3}$  and the corresponding values of  $h$  and  $n^*$ .

## 4.5. Application to scaled three-dimensional vine copulas

Considering the vine structure from Equation 4.8 we further assume that the pair-copulas are one-parametric. Having fixed  $\lambda_{1,2}$ ,  $\lambda_{1,3}$ ,  $\lambda_{2,3}$  such that  $h(\lambda_{1,2}^2, \lambda_{1,3}^2, \lambda_{2,3}^2) > 0$ , we can perform the following procedure for  $n \geq n^*$ :

1. Calculate  $\rho_{1,2}(n)$ ,  $\rho_{1,3}(n)$  and  $\rho_{2,3}(n)$  with the help of Equation 4.12.
2. Determine the corresponding partial correlation using

$$\rho_{1,3;2}(n) = \frac{\rho_{1,3}(n) - \rho_{1,2}(n)\rho_{2,3}(n)}{\sqrt{1 - \rho_{1,2}(n)^2}\sqrt{1 - \rho_{2,3}(n)^2}}.$$

3. Transform the (partial) correlations  $\rho_{1,2}(n)$ ,  $\rho_{2,3}(n)$  and  $\rho_{1,3;2}(n)$  into the corresponding (partial) Kendall's  $\tau$  values  $\tau_{1,2}(n)$ ,  $\tau_{2,3}(n)$  and  $\tau_{1,3;2}(n)$  using the relation for elliptical distributions

$$\tau = \frac{2}{\pi} \arcsin(\rho).$$

4. Determine the parameters  $\theta_{1,2}(n)$ ,  $\theta_{2,3}(n)$  and  $\theta_{1,3;2}(n)$  of the pair copulas from the corresponding  $\tau$  values.<sup>6</sup>

Recall that  $\rho_{1,2}(n) \rightarrow 1$ ,  $\rho_{1,3}(n) \rightarrow 1$  and  $\rho_{2,3}(n) \rightarrow 1$  as  $n \rightarrow \infty$ . Therefore, we also have  $\tau_{1,2}(n) \rightarrow 1$ ,  $\tau_{1,3}(n) \rightarrow 1$  and  $\tau_{2,3}(n) \rightarrow 1$  as  $n \rightarrow \infty$ . However, the behavior of convergence of  $\rho_{1,3;2}(n)$  and hence  $\tau_{1,3;2}(n)$  is not trivial. We use Equation 4.12 to obtain

$$\rho_{1,3;2}(n) = \frac{\left(1 - \frac{\lambda_{1,3}^2}{\log(n)}\right) - \left(1 - \frac{\lambda_{1,2}^2}{\log(n)}\right) \left(1 - \frac{\lambda_{2,3}^2}{\log(n)}\right)}{\sqrt{1 - \left(1 - \frac{\lambda_{1,2}^2}{\log(n)}\right)^2} \sqrt{1 - \left(1 - \frac{\lambda_{2,3}^2}{\log(n)}\right)^2}} \rightarrow \frac{\lambda_{1,2}^2 + \lambda_{2,3}^2 - \lambda_{1,3}^2}{2\lambda_{1,2}\lambda_{2,3}}$$

<sup>6</sup>In the `VineCopula` package this transformation can be performed using the function `BiCopTau2Par`.

#### 4.5. Application to scaled three-dimensional vine copulas

as  $n \rightarrow \infty$ . Thus,

$$\tau_{1,3;2}(n) \rightarrow \frac{2}{\pi} \arcsin \left( \frac{\lambda_{1,2}^2 + \lambda_{2,3}^2 - \lambda_{1,3}^2}{2\lambda_{1,2}\lambda_{2,3}} \right) \text{ as } n \rightarrow \infty.$$

For illustration, we will now consider combinations 9 and 10 from Table 4.1. In Table 4.2 we show the (partial) correlations from Step 2 of the above procedure for different block sizes. We also present the corresponding (partial) Kendall's  $\tau$  values since they can be compared independently from the choice the respective pair-copulas.

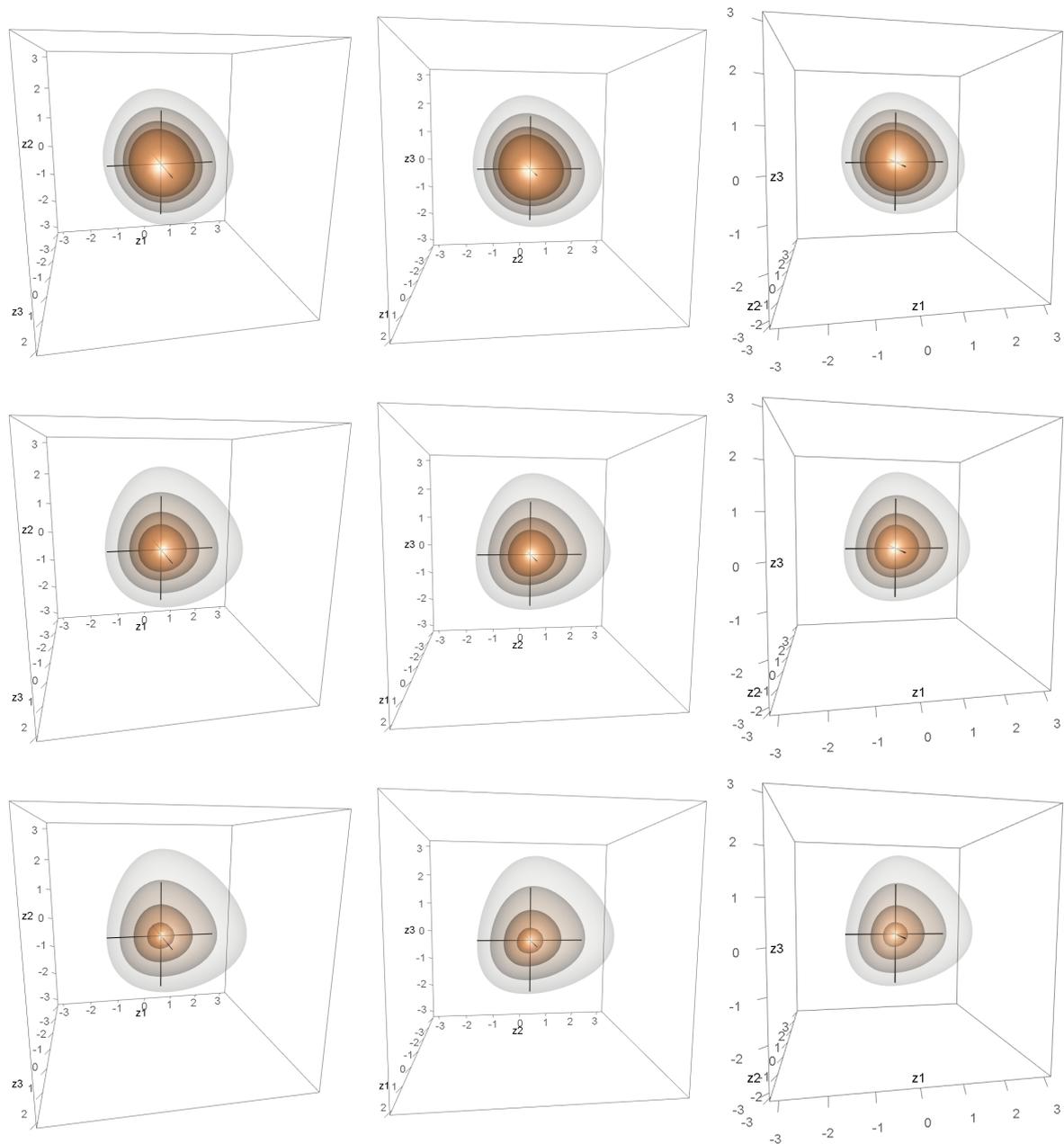
	$n$	$\rho_{1,2}(n)$	$\rho_{2,3}(n)$	$\rho_{1,3;2}(n)$	$\tau_{1,2}(n)$	$\tau_{2,3}(n)$	$\tau_{1,3;2}(n)$
Combination 9	10	0.54	0.64	0.87	0.37	0.44	0.67
	50	0.73	0.79	0.88	0.52	0.57	0.69
	1000	0.85	0.88	0.89	0.64	0.68	0.69
	$\infty$	1	1	0.89	1	1	0.70
Combination 10	10	-0.74	-0.30	-0.82	-0.53	-0.20	-0.61
	50	-0.02	0.23	0.25	-0.01	0.15	0.16
	1000	0.42	0.57	0.44	0.28	0.38	0.29
	$\infty$	1	1	0.58	1	1	0.39

**Table 4.2.:** Overview over the (partial) correlations and (partial) Kendall's  $\tau$  values for different  $n$  for combinations 9 ( $\lambda_{1,2}^2 = 1.05$ ,  $\lambda_{1,3}^2 = 0.21$ ,  $\lambda_{2,3}^2 = 0.84$ ) and 10 ( $\lambda_{1,2}^2 = 4$ ,  $\lambda_{1,3}^2 = 3$ ,  $\lambda_{2,3}^2 = 3$ ).

If we consider the values from Table 4.2 for combinations 9 and 10, it is eye-catching that the choice of  $\lambda_{1,2}$ ,  $\lambda_{1,3}$  and  $\lambda_{2,3}$  has a crucial influence on the behavior of the (partial) correlations and the (partial) Kendall's  $\tau$  values. In the first case the parameters are already relatively close to their limiting values for  $n = 1000$ , whereas in the second case they are still rather far from their limits for  $n = 1000$ . Further, we see that the limiting values of  $\rho_{1,3;2}(n)$  and  $\tau_{1,3;2}(n)$  can be very different depending on the choice of  $\lambda_{1,2}$ ,  $\lambda_{1,3}$

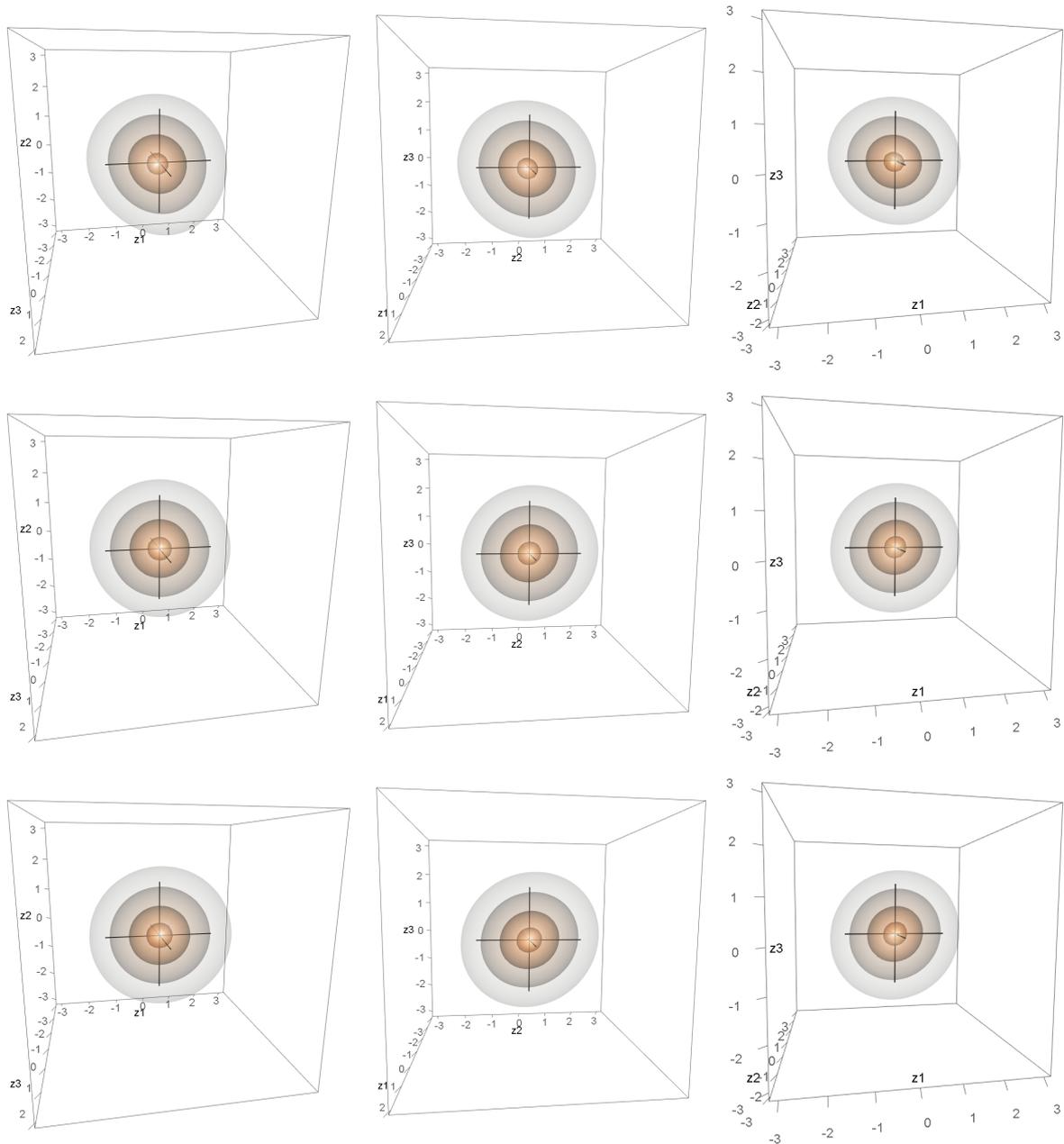
Now we examine the behavior of the three-dimensional density of the scaled block maxima  $f_{\mathbf{W}(n)}$  (cf. Equation 4.10) for different values of  $n$ .

**Example 4.10.** Again we start with a Gaussian vine and the parameters of combination 10:  $\lambda_{1,2}^2 = 4$ ,  $\lambda_{1,3}^2 = 3$  and  $\lambda_{2,3}^2 = 3$ . Figure 4.6 shows the contour surfaces of the density  $f_{\mathbf{W}(n)}$  of the scaled block maxima of the Gaussian vine for block sizes  $n = 10, 50, 1000$  (rows) from different angles (columns). The contour levels are fixed to be 0.015, 0.035, 0.075 and 0.110 (from outer to inner surface) as in Section 4.3. The Kendall's  $\tau$  values corresponding to the three block size can be found in Table 4.2. Since the margins in the plots of Figure 4.6 are given by Equation 4.9 we can hardly assess the underlying dependence structure. The underlying copulas, however, are the main interest in this chapter such that we additionally consider similar plots, where the only difference is that the margins are set to be standard normal (Figure 4.7). We see that with increasing block



**Figure 4.6.:** Plot of the contour surfaces of the density of the scaled block maxima of a three-dimensional Gaussian vine from different angles (columns). The rows correspond to block sizes  $n = 10, 50$  and  $1000$ .

#### 4.5. Application to scaled three-dimensional vine copulas



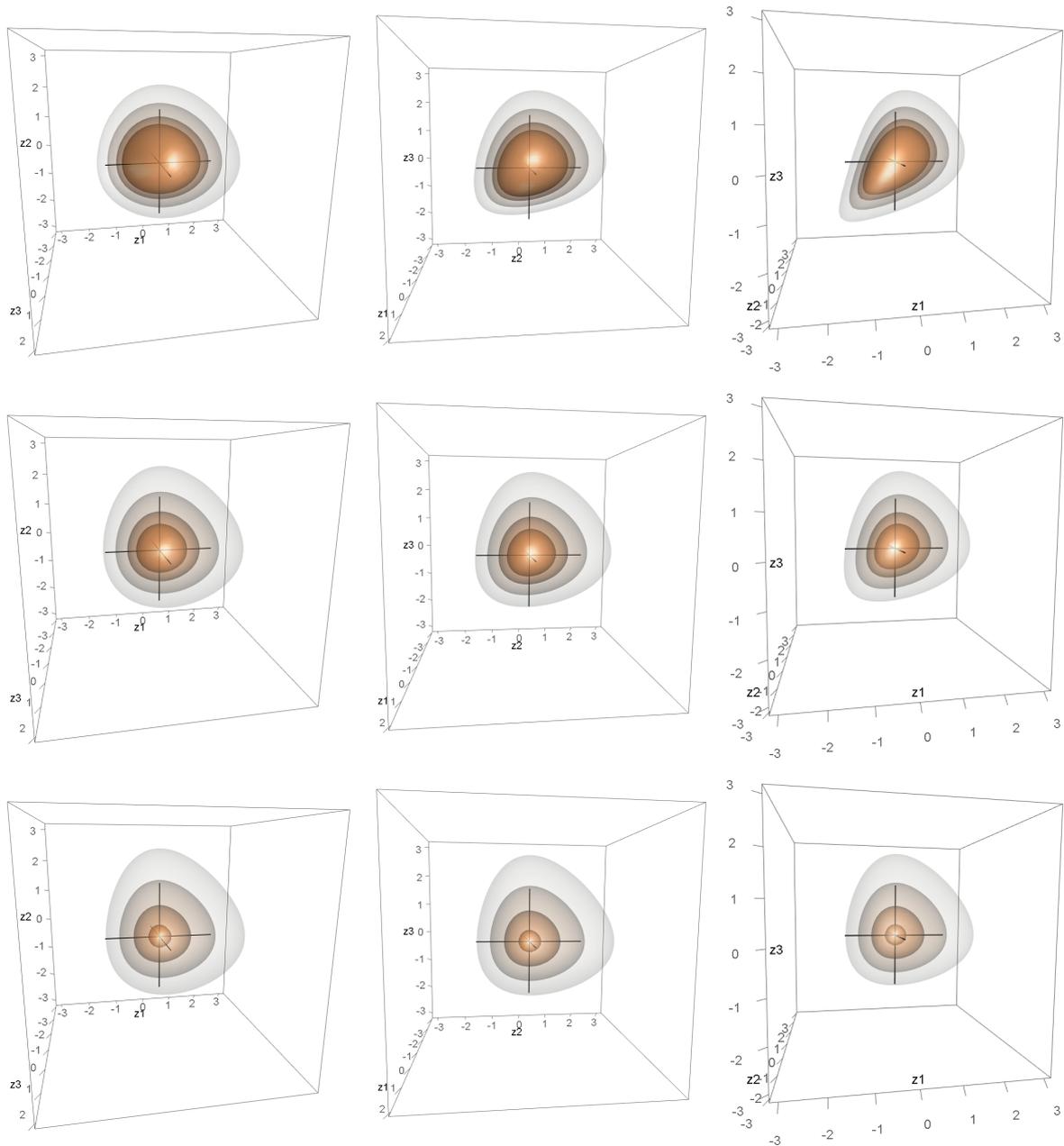
**Figure 4.7.:** Plot of the contour surfaces of the copula density of the scaled block maxima of a three-dimensional Gaussian vine with standard normal margins from different angles (columns). The rows correspond to block sizes  $n = 10, 50$  and  $1000$ .

size the contour surfaces again seem to tend to the contour surfaces of the independence copula. Having the results of Hüsler and Reiss (1989) in mind this might appear contradictory. However, if we look at the Kendall's  $\tau$  values for  $n = 1000$  we see that they are still extremely small:  $\tau_{1,2}(n) = 0.28$ ,  $\tau_{2,3}(n) = 0.38$  and  $\tau_{1,3;2}(n) = 0.29$ . They are still extremely far from their theoretical limits of 1, 1 and 0.39, respectively. Thus, we simply have the density of the block maxima of a Gaussian copula with rather low strengths of dependence, for which we already know that it lies in the domain of attraction of the independence copula (Sibuya, 1960). If we increase  $n$  further, the Kendall's  $\tau$  values will converge to their theoretical limits and we know from Hüsler and Reiss (1989) that the copula of the block maxima will approach a Hüsler–Reiss copula. This, however, requires very large block size as even for  $n = 10^{50}$  the corresponding Kendall's  $\tau$  values are given by only 0.83, 0.85 and 0.39, respectively. On the one hand, this is still far from the point where one could assume that the asymptotics have set in, on the other hand we cannot even handle such large values for  $n$  numerically. Therefore, the lesson of this example is that although in theory the scaling of the block maxima and the underlying dependence structure may lead to non-trivial limits, for finite block sizes this does not necessarily have an impact.

**Example 4.11.** Our final example is a Clayton vine, where we choose  $\lambda_{1,2}^2 = 1.05$ ,  $\lambda_{1,3}^2 = 0.21$  and  $\lambda_{2,3}^2 = 0.84$  (corresponding to combination 9). Figure 4.8 shows the contour surfaces of the density  $f_{\mathbf{W}^{(n)}}$  of the scaled block maxima of the Clayton vine for block sizes  $n = 10, 50, 1000$  (rows) from different angles (columns). The Kendall's  $\tau$  values corresponding to the three block size can again be found in Table 4.2. The similarity to the plots for the Gaussian vine (Figure 4.6) is obvious for  $n = 50$  and 1000. Therefore we do not additionally show the plots of the corresponding copula density of the scaled block maxima with standard normally distributed margins. The copula density of the scaled block maxima is very close to the independence copula for  $n = 50$  and 1000. Again even for  $n = 10^{50}$  the Kendall's  $\tau$  values are given by only 0.91, 0.92 and 0.70, respectively, which would not suffice to assume that the asymptotic behavior can already be observed. However, we do not have any theoretical results about a possibly non-trivial limit of a Clayton vine even when scaling is applied.

## 4.6. Conclusion

In this chapter we showed that the copula density of componentwise block maxima of multivariate distributions can be expressed explicitly using the copula function of the underlying distribution and its derivatives. For three-dimensional vine copulas we made use of the fact that computing their partial derivatives requires only one-dimensional



**Figure 4.8.:** Plot of the contour surfaces of the density of the scaled block maxima of a three-dimensional Clayton vine from different angles (columns). The rows correspond to block sizes  $n = 10, 50$  and  $1000$ .

integration, which makes the evaluation of the copula density for block maxima numerically tractable. The advantage of our method is that we can use the entire sample for estimation instead of reducing the sample size to one  $n$ th of the original sample size by taking maxima over  $n$  observations. Once we have estimated the underlying dependence structure we can derive the copula density of the block maxima for any block size (even larger than the original sample size). In three numerical examples and an application to hydrological data we saw that the block maxima did not approach a non-trivial limiting

distribution for increasing block size. Therefore, we mimicked the scaling approach of Hüsler and Reiss (1989) yielding non-trivial limits for Gaussian distributions. Numerical examples, however, showed that for finite block sizes the asymptotic behavior cannot be observed.

# Chapter 5

---

## *Modeling repeated measurements using D-vine copulas*

---

The contents of this chapter are a lightly modified reproduction of the contents of Killiches and Czado (2017).

### 5.1. Introduction

Repeated measurements that are obtained in a longitudinal study are common in many areas. Very early applications in astronomy (Airy, 1861) were followed by a vast number of studies in fields such as industry (e.g. Newbold, 1927), ecology (e.g. Potvin et al., 1990), biology (e.g. Yeung et al., 2003), psychology (e.g. Lorch and Myers, 1990), medicine (e.g. Ludbrook, 1994), education (e.g. Malin and Linnakylä, 2001) and many more.

Over the years many concepts have been developed for the analysis of such repeated measurements. A discussion of the origins of longitudinal data models can be found in Chapter 1 of Fitzmaurice et al. (2008). Davis (2002) offers a thorough introduction to the topic, starting with basic aspects of repeated measurement data. Besides foundations and different modeling aspects of repeated measurement data, Lindsey (1999) addresses the question of how to design a study. Diggle and Donnelly (1989) give an extensive review on different approaches to the analysis of repeated measurements. The most popular model class for this purpose is probably the one of *linear mixed models* (LMMs). It extends classical linear models by adding individual-specific random effects to the fixed effects. Extensive introductions to this topic can be found for example in Diggle (2002) and Verbeke and Molenberghs (2009). Although the covariance structure of linear mixed models can be fitted rather flexibly, the dependence always remains Gaussian by definition.

Within the last two decades dependence modeling has become more and more popular

in all areas of applications. Especially copulas have gained large popularity since they allow to model marginal distributions and the dependence structure separately (Sklar, 1959). Consequently, copulas were also applied for modeling repeated measurement data. This approach has first been used by Meester and MacKay (1994) who developed a model for bivariate clustered categorical data. Lambert and Vandenhende (2002) present a model for multivariate repeated measurement data, where the dependence is described by any copula (although only the Gaussian copula is used in the application). Shen and Weissfeld (2006) model serial dependence for continuous longitudinal data with a non-ignorable non-monotone missing-data process using a Gaussian copula. Another example is Lindsey and Lindsey (2006), who use the Gaussian copula among other multivariate models with correlation matrices for non-linear repeated measurements. Further, Sun et al. (2008) argue that elliptical copulas are better suited than Archimedean copulas for modeling serial dependence in the context of longitudinal data.

*D-vine copulas* are a special class of vine copulas (Bedford and Cooke, 2002; Aas et al., 2009) that are particularly suited for modeling serial dependence. Smith et al. (2010) used them to model longitudinal data in a Bayesian approach. Multivariate time series are considered in Smith (2015) and Nai Ruscone and Osmetti (2017). In Joe (2014, Chapter 7.5) discrete longitudinal count data are modeled using D-vines. Shi and Yang (2016) use a mixed D-vine to model semi-continuous longitudinal claims. All these references work in a balanced setting, i.e. each individual has the same number of measurements. An unbalanced setting is considered by Shi et al. (2016) using a Gaussian copula.

The novelty of the approach presented in this chapter is that we develop a D-vine copula based model with arbitrary margins for modeling unbalanced longitudinal data with the aim of understanding the underlying relationship among the measurements and enabling predictions for future events. For prediction we use conditional quantiles that are analytically given. For model selection we derive an adjustment of the Bayesian information criterion (BIC) for the proposed model. The model will furthermore be shown to be an extension of a very wide class of linear mixed models for which the correlation matrix of the measurements is homogeneous over the individuals.

Section 5.2 briefly introduces D-vine copulas and the proposed D-vine copula based model for repeated measurement data. Linear mixed models and their connection with the D-vine based model are developed in Section 5.3. Section 5.4 contains maximum-likelihood based estimation methods for the D-vine based model. Further, as a tool for model selection, an adjustment of the BIC for the proposed model is derived. The performance of the estimation methods is investigated in a simulation study (Section 5.5). In Section 5.6 we fit both LMMs and D-vine based models to a heart surgery data set and compare the results using likelihood based model selection criteria and performing conditional quantile prediction. Section 5.7 contains our conclusions and an outlook.

## 5.2. D-vine based repeated measurement model

### 5.2.1. Setting

Consider a repeated measurement (longitudinal) data set  $\mathcal{Y} = \{\mathbf{y}^1, \dots, \mathbf{y}^n\}$  that contains  $n \in \mathbb{N}$  observation blocks  $\mathbf{y}^i = (y_1^i, \dots, y_{d_i}^i)^\top \in \mathbb{R}^{d_i}$  associated with individual  $i$  having  $d_i \in \{1, \dots, d\}$  measurements. Here  $d \in \mathbb{N}$  denotes the maximum number of measurements per individual observed. For two different individuals the  $j$ th event does not necessarily need to have occurred at the same time  $t_j$ . We denote by  $n_j$  the number of observations of length  $j$ ,  $j = 1, \dots, d$ , where  $n_j$  is zero if  $\mathcal{Y}$  contains no observations of length  $j$ . We divide now the data set into groups of individuals with the same number of measurements. For  $j = 1, \dots, d$ , we summarize the observations of group  $j$  as  $\mathcal{Y}^j = \{\mathbf{y}^i \mid i \in I_j\}$ , where the corresponding index set is defined as  $I_j = \{i \mid \mathbf{y}^i \in \mathbb{R}^j\}$ . Table 5.1 illustrates the above notation and data structure for an exemplary data set of size  $n = 9$ , where the maximum number of measurements per individual is  $d = 4$  and we have  $n_1 = 0$  individuals with 1 measurement,  $n_2 = 3$  individuals with 2 measurements,  $n_3 = 2$  individuals with 3 measurements and  $n_4 = 4$  individuals with 4 measurements. Consequently,  $I_1 = \emptyset$ ,  $I_2 = \{1, 2, 3\}$ ,  $I_3 = \{4, 5\}$  and  $I_4 = \{6, 7, 8, 9\}$ .

observations	measurements			
	1	2	3	4
$\mathcal{Y}^2 = \{\mathbf{y}^i \mid i \in I_2\}$	$\begin{cases} \mathbf{y}^1 \\ \mathbf{y}^2 \\ \mathbf{y}^3 \end{cases}$	* * *	* * *	
$\mathcal{Y}^3 = \{\mathbf{y}^i \mid i \in I_3\}$	$\begin{cases} \mathbf{y}^4 \\ \mathbf{y}^5 \end{cases}$	* *	* *	
$\mathcal{Y}^4 = \{\mathbf{y}^i \mid i \in I_4\}$	$\begin{cases} \mathbf{y}^6 \\ \mathbf{y}^7 \\ \mathbf{y}^8 \\ \mathbf{y}^9 \end{cases}$	* * * *	* * * *	* * * *

**Table 5.1.:** Grouping of an exemplary data set of size  $n = 9$  with  $d = 4$ ,  $n_2 = 3$ ,  $n_3 = 2$  and  $n_4 = 4$ . Stars indicate observed events.

Having Sklar's Theorem (Sklar, 1959) in mind, we follow a two-stage approach, also referred to as the Inference Functions for Margins (IFM) method (cf. Joe, 1997, Section 10.1): First we use the probability integral transform and apply the univariate marginal distributions  $F_j^i$  to the measurements  $y_j^i \in \mathbb{R}$  in order to transform them to measurements  $u_j^i := F_j^i(y_j^i) \in [0, 1]$  on the uniform scale,  $j = 1, \dots, d_i$  and  $i = 1, \dots, n$ . Then we model the dependence structure of the resulting uniform scale data utilizing a copula. In the

following sections we will use a notation for the copula data that is similar to the one for the original data. The copula data  $\mathcal{U} = \{\mathbf{u}^1, \dots, \mathbf{u}^n\}$  consists of the observations  $\mathbf{u}^i = (u_1^i, \dots, u_d^i)^\top \in [0, 1]^{d_i}$ ,  $i = 1, \dots, n$ . Again, we form groups  $\mathcal{U}^j = \{\mathbf{u}^i \mid i \in I_j\}$  containing all observations of length  $j$ ,  $j = 1, \dots, d$ . Since individuals with only one measurement do not contribute to the dependence structure we will only consider  $\mathcal{U}^2, \dots, \mathcal{U}^d$ . Thus we can assume that  $n_1 = 0$ , i.e.  $\mathcal{U}^1 = \emptyset$ , without losing generality. Of course, in practice the distribution functions  $F_j^i$  are usually not known and need to be estimated (see Section 5.4).

## 5.2.2. D-vine based dependence model

### D-vine copulas

Recall from Section 2.2.1 that D-vine copulas are vine copulas with a special tree structure, which is illustrated in  $d = 4$  dimensions in Figure 5.1 on page 83. Due to their structure, following the notation of Czado (2010), the density of a D-vine copula with order 1–2– $\dots$ – $d$  can be written as

$$c_{1:d}(u_1, \dots, u_d) = \prod_{\ell=1}^{d-1} \prod_{k=1}^{d-\ell} c_{k, k+\ell; (k+1):(k+\ell-1)}(C_{k|(k+1):(k+\ell-1)}(u_k | u_{k+1}, \dots, u_{k+\ell-1}), \quad (5.1)$$

$$C_{k+\ell|(k+1):(k+\ell-1)}(u_{k+\ell} | u_{k+1}, \dots, u_{k+\ell-1}); u_{k+1}, \dots, u_{k+\ell-1}).$$

Here  $c_{k, k+\ell; (k+1):(k+\ell-1)}(\cdot, \cdot; u_{k+1}, \dots, u_{k+\ell-1})$  is the bivariate copula density associated with the distribution of  $(U_k, U_{k+\ell})^\top$  given  $(U_{k+1}, \dots, U_{k+\ell-1})^\top = (u_{k+1}, \dots, u_{k+\ell-1})^\top$  and  $C_{k|(k+1):(k+\ell-1)}(\cdot | u_{k+1}, \dots, u_{k+\ell-1})$  is the distribution function of the conditional distribution of  $U_k$  given  $(U_{k+1}, \dots, U_{k+\ell-1})^\top = (u_{k+1}, \dots, u_{k+\ell-1})^\top$ ,  $\ell = 1, \dots, d - 1$  and  $k = 1, \dots, d - \ell$ . The corresponding graphical interpretation is the tree representation, where the pair-copulas occurring in tree  $j$  have a conditioning set of size  $j - 1$ ,  $j = 1, \dots, d - 1$ .

In order to ease inference later we make the simplifying assumption (cf. Section 2.2.3) although we could set up our model without it as well.

In the following we will assume a parametric model such that a D-vine copula can be identified by the set of pair-copula families  $\mathcal{C} = (c_{k, k+\ell; (k+1):(k+\ell-1)} \mid k = 1, \dots, d - \ell \text{ and } \ell = 1, \dots, d - 1)$  and the set of associated parameters  $\boldsymbol{\theta} = (\boldsymbol{\theta}_{k, k+\ell; (k+1):(k+\ell-1)} \mid k = 1, \dots, d - \ell \text{ and } \ell = 1, \dots, d - 1)$ . In general, non-parametric pair-copulas could also be used (see Nagler and Czado, 2016).

A convenient property is that D-vine models are nested in the sense that the pair-copulas needed to describe the dependence of variables 1 to  $j$  are contained in the model describing the dependence of variables 1 to  $j + 1$ ,  $j < d$ . This is illustrated in Figure 5.1 on page 83.

### Model description

Since the data has been obtained from repeated measurements there exists a clear sequential or temporal ordering. This immediately suggests the use of D-vine copulas with order  $1-2-\dots-d$  (Smith et al., 2010; Smith, 2015; Nai Ruscone and Osmetti, 2017). Therefore, as a general approach, we assume parametric simplified D-vine models (cf. Equation 5.1) for the copula densities of all groups  $j = 2, \dots, d$ . Of course, we only consider groups for which we have observations. The copula density  $c_{1:j}^j$  of group  $j$  then can be described with the help of the set of the  $j(j-1)/2$  pair-copula families

$$\mathcal{C}^j = (c_{k,k+\ell;(k+1):(k+\ell-1)}^j \mid k = 1, \dots, j - \ell \text{ and } \ell = 1, \dots, j - 1)$$

and the set of associated parameters

$$\boldsymbol{\theta}^j = (\boldsymbol{\theta}_{k,k+\ell;(k+1):(k+\ell-1)}^j \mid k = 1, \dots, j - \ell \text{ and } \ell = 1, \dots, j - 1)$$

for  $j = 2, \dots, d$  with a non-empty  $\mathcal{U}^j$ . For the estimation of  $\mathcal{C}^j$  and  $\boldsymbol{\theta}^j$  we set up the likelihood, which is based on the subset of  $\mathcal{U}$  containing the observations of length  $j$ . The resulting likelihood and log-likelihood can be written as

$$\begin{aligned} L_j(\mathcal{C}^j, \boldsymbol{\theta}^j \mid \mathcal{U}^j) &= \prod_{i \in I_j} c_{1:j}^j(u_1^i, \dots, u_j^i \mid \mathcal{C}^j, \boldsymbol{\theta}^j) \quad \text{and} \\ \log L_j(\mathcal{C}^j, \boldsymbol{\theta}^j \mid \mathcal{U}^j) &= \sum_{i \in I_j} \log c_{1:j}^j(u_1^i, \dots, u_j^i \mid \mathcal{C}^j, \boldsymbol{\theta}^j), \end{aligned}$$

respectively. Consequently, the log-likelihood of the general model is given by

$$\log L(\mathcal{C}^2, \dots, \mathcal{C}^d, \boldsymbol{\theta}^2, \dots, \boldsymbol{\theta}^d \mid \mathcal{U}) = \sum_{j=2}^d \log L_j(\mathcal{C}^j, \boldsymbol{\theta}^j \mid \mathcal{U}^j). \quad (5.2)$$

For future reference we call this *Model A*. It is obvious by construction that the models for different groups can be estimated independently from each other since there are no intersections between the groups, neither regarding data nor pair-copula families or parameters. From a practical point of view this would correspond to the assumption that the dependence structure of two groups can be completely different such that an individual for whom we have observed  $j$  events has nothing in common with those who have had  $j+1$  events. However, one can argue that an individual from group  $j$  is basically a member of group  $j+1$  for whom the  $(j+1)$ st measurement has not been observed yet. The underlying random mechanism (i.e. the copula), however, should be the same or at least share some properties. Therefore, it makes sense to impose more restrictions on the set of pair-copula families and the associated parameters. For example, one could assume

that all groups share the same pair-copula families and only the parameters can differ between the groups. The most sensible and interesting case—which we will pursue in the following—is the one that all groups have the same pair-copula families and parameters,

$$\begin{aligned} c_{k,k+\ell;(k+1):(k+\ell-1)}^j &= c_{k,k+\ell;(k+1):(k+\ell-1)}, \\ \boldsymbol{\theta}_{k,k+\ell;(k+1):(k+\ell-1)}^j &= \boldsymbol{\theta}_{k,k+\ell;(k+1):(k+\ell-1)} \end{aligned} \quad (5.3)$$

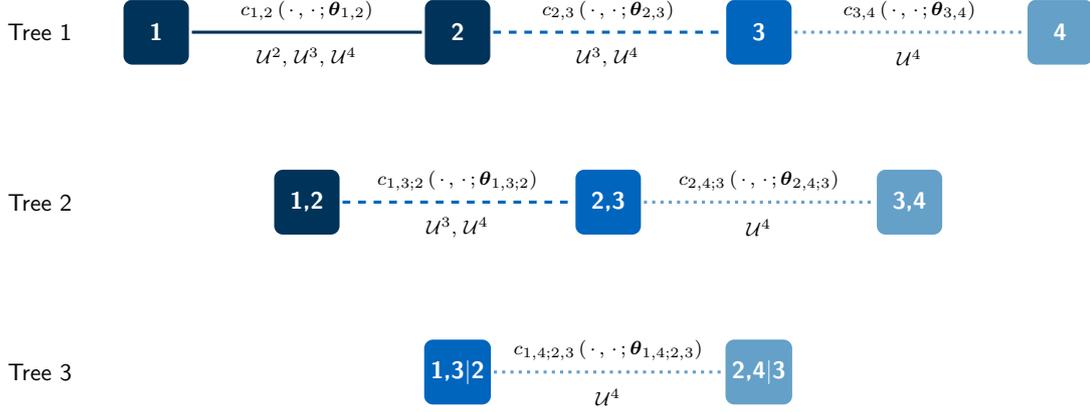
for  $k = 1, \dots, j - \ell$  and  $\ell = 1, \dots, j - 1$ . We will refer to this model as *Model B*. Using the same families and parameters for all groups implies that the  $D$ -vine describing the dependence pattern of group  $j$  is a sub-vine of the vine of groups  $j + 1, \dots, d$ . In particular, the vine copula density of group  $j$  is simply the multivariate marginal density  $c_{1:j}$  of the density  $c_{1:d}$  of the largest group  $d$ . Consequently,  $c_{1:d}$  describes the full model, from which the models of all smaller groups can be explicitly derived. Thus, the corresponding log-likelihood only depends on one set of  $d(d - 1)/2$  pair-copula families  $\mathcal{C} = (c_{k,k+\ell;(k+1):(k+\ell-1)} \mid k = 1, \dots, d - \ell \text{ and } \ell = 1, \dots, d - 1)$  and the set of associated parameters  $\boldsymbol{\theta} = (\boldsymbol{\theta}_{k,k+\ell;(k+1):(k+\ell-1)} \mid k = 1, \dots, d - \ell \text{ and } \ell = 1, \dots, d - 1)$ .

### Example

In order to illustrate the above concept we will now look at the example with at most  $d = 4$  repeated measurements. Assume we have (up to) four-dimensional repeated measurement data  $\mathcal{U} = \{\mathbf{u}^1, \dots, \mathbf{u}^n\}$  of size  $n = n_2 + n_3 + n_4$  ordered as described in Section 5.2.1, which can be partitioned into groups 2, 3 and 4 by defining  $\mathcal{U}^j = \{u^i \mid i \in I_j\}$ ,  $j = 2, 3, 4$ , where  $I_2 = \{i \mid \mathbf{u}^i \in \mathbb{R}^2\}$ ,  $I_3 = \{i \mid \mathbf{u}^i \in \mathbb{R}^3\}$  and  $I_4 = \{i \mid \mathbf{u}^i \in \mathbb{R}^4\}$ . The model and hence its log-likelihood depends on the six pair-copulas  $\mathcal{C} = (c_{1,2}, c_{2,3}, c_{3,4}, c_{1,3;2}, c_{2,4;3}, c_{1,4;2,3})$  and the associated parameters  $\boldsymbol{\theta} = (\boldsymbol{\theta}_{1,2}, \boldsymbol{\theta}_{2,3}, \boldsymbol{\theta}_{3,4}, \boldsymbol{\theta}_{1,3;2}, \boldsymbol{\theta}_{2,4;3}, \boldsymbol{\theta}_{1,4;2,3})$ . Figure 5.1 shows a schematic representation of the full model  $c_{1:4}$  with its pair-copulas and parameters. The nodes represent the measurements. Above and below each edge the associated pair-copula and the observations that can be used for estimation are denoted, respectively. The sub-vines for  $c_{1:2}$  and  $c_{1:3}$  are highlighted by different color intensities of the nodes and line types of the edges. The resulting log-likelihood is given by

$$\begin{aligned} \log L(\mathcal{C}, \boldsymbol{\theta} \mid \mathcal{U}) &= \sum_{i \in I_2} \log c_{1:2}(u_1^i, u_2^i \mid c_{1,2}, \boldsymbol{\theta}_{1,2}) \\ &+ \sum_{i \in I_3} \log c_{1:3}(u_1^i, u_2^i, u_3^i \mid c_{1,2}, c_{2,3}, c_{1,3;2}, \boldsymbol{\theta}_{1,2}, \boldsymbol{\theta}_{2,3}, \boldsymbol{\theta}_{1,3;2}) \\ &+ \sum_{i \in I_4} \log c_{1:4}(u_1^i, u_2^i, u_3^i, u_4^i \mid c_{1,2}, c_{2,3}, c_{3,4}, c_{1,3;2}, c_{2,4;3}, c_{1,4;2,3}, \\ &\quad \boldsymbol{\theta}_{1,2}, \boldsymbol{\theta}_{2,3}, \boldsymbol{\theta}_{3,4}, \boldsymbol{\theta}_{1,3;2}, \boldsymbol{\theta}_{2,4;3}, \boldsymbol{\theta}_{1,4;2,3}) \end{aligned} \quad (5.4)$$

## 5.2. D-vine based repeated measurement model



**Figure 5.1.:** Illustration of the four-dimensional D-vine describing the components of the dependence structure of the full model  $c_{1,4}$  (dark, medium and light). The sub-vines for  $c_{1,2}$  (dark) and  $c_{1,3}$  (dark and medium) are highlighted by different color intensities of the nodes and line types of the edges. Above and below each edge the associated pair-copula and the observations that can be used for estimation are denoted, respectively.

Using the vine decomposition from Equation 5.1 for  $c_{1,2}$ ,  $c_{1,3}$  and  $c_{1,4}$ , the log-likelihood associated with data  $\mathcal{U}$  (Equation 5.4) can be re-written as

$$\begin{aligned}
 \log L(\mathcal{C}, \boldsymbol{\theta} \mid \mathcal{U}) = & \sum_{i \in I_2 \cup I_3 \cup I_4} \log c_{1,2}(u_1^i, u_2^i; \boldsymbol{\theta}_{1,2}) \\
 & + \sum_{i \in I_3 \cup I_4} \left[ \log c_{2,3}(u_2^i, u_3^i; \boldsymbol{\theta}_{2,3}) + \log c_{1,3;2}(C_{1|2}(u_1^i | u_2^i; \boldsymbol{\theta}_{12}), C_{3|2}(u_3^i | u_2^i; \boldsymbol{\theta}_{23}); \boldsymbol{\theta}_{1,3;2}) \right] \\
 & + \sum_{i \in I_4} \left[ \log c_{3,4}(u_3^i, u_4^i; \boldsymbol{\theta}_{3,4}) + \log c_{2,4;3}(C_{2|3}(u_2^i | u_3^i; \boldsymbol{\theta}_{2,3}), C_{4|3}(u_4^i | u_3^i; \boldsymbol{\theta}_{4,3}); \boldsymbol{\theta}_{2,4;3}) \right. \\
 & \quad \left. + \log c_{1,4;2,3}(C_{1|3;2}(C_{1|2}(u_1^i | u_2^i; \boldsymbol{\theta}_{1,2}) \mid C_{3|2}(u_3^i | u_2^i; \boldsymbol{\theta}_{2,3}); \boldsymbol{\theta}_{1,3;2}), \right. \\
 & \quad \left. C_{4|2;3}(C_{2|3}(u_2^i | u_3^i; \boldsymbol{\theta}_{2,3}) \mid C_{4|3}(u_4^i | u_3^i; \boldsymbol{\theta}_{4,3}); \boldsymbol{\theta}_{2,4;3}); \boldsymbol{\theta}_{1,4;2,3}) \right]
 \end{aligned} \tag{5.5}$$

For the general case of Model A (Equation 5.2) we saw that the pair-copulas and parameters corresponding to group  $j$  can be estimated independently from those of the remaining groups and only depend on the data contained in  $\mathcal{U}^j$ . Looking at Equation 5.5 (corresponding to Model B) it immediately becomes clear that assuming the pair-copulas and parameters are the same for all groups has changed this phenomenon. The D-vines describing the densities  $c_{1,2}$  and  $c_{1,3}$  are nested sub-vines of the full model  $c_{1,4}$ , which

can easily be understood from Figure 5.1: The dark nodes (and solid edges) correspond to  $c_{1,2}$ ; adding the medium colored nodes (and dashed edges) results in the model of  $c_{1,3}$ ; incorporating also the light nodes (and dotted edges) yields the full model for  $c_{1,4}$ . Therefore, when it comes to estimation we see for example that not only the observations belonging to  $\mathcal{U}^2$  but also those from  $\mathcal{U}^3$  and  $\mathcal{U}^4$  (i.e. the entire sample  $\mathcal{U}$ ) have an influence on the estimate for  $c_{1,2}$  and  $\theta_{1,2}$ . Thus this increases the accuracy of the estimation compared to the approach from Model A.

The assumption of common pair-copula families and parameters for all groups comes with the advantages of better interpretability, less parameters and higher estimation accuracy.

### Missing values

In practice, unfortunately, data do not always look exactly the way we described it in Section 5.2.1. Sometimes there are missing values in the data. For example, there might be individuals for whom the first, third, fourth and fifth measurement are available but the second one is missing. Such situation can occur for various reasons, e.g. a patient skips a measurement date due to illness, measuring instruments have problems causing a loss of the result or data is simply not reported due to human failure. Moreover, there might be (non-informative) dropouts, i.e. individuals with measurements only up to a certain time, e.g. caused by relocation of a patient to another city.

For many model classes such observations cannot be used at all and have to be removed from the data set for model estimation. This way the sample size is decreased and information is lost. For Model B, however, observations with missing values can still be used (assuming they are missing at random). The information gained from our exemplary individual with measurements 1, 3, 4, 5 still contributes to the estimation of  $c_{3,4}$ ,  $c_{4,5}$  and  $c_{3,5,4}$  (and of course to the estimation of the marginal distributions 1, 3, 4, 5). Since the missing second measurement is needed for the estimation of the remaining pair-copulas, this individual cannot be used in order to estimate them. Nevertheless, we prevent the loss of the individual's entire information. In order to include observations with missing value into our model we simply have to modify the log-likelihood such that the sums of the log-likelihood of each pair-copula includes all observations for whom the necessary measurements are available. For the sake of notation we will stick to the formulation of Model B as above, keeping in mind that missing values can also be handled by our approach.

### Conditional prediction

Further, we can use our repeated measurement data model for prediction. In many applications it can be interesting to have a prediction for the size of an upcoming measurement.

For instance, having proper estimates for future claims can be a competitive advantage for the risk management department of an insurance company.

For a  $d$ -dimensional model, consider an individual  $i$  that has had  $d_i < d$  measurements so far, i.e.  $\mathbf{y}^i = (y_1^i, \dots, y_{d_i}^i)^\top$ . We are now interested in the distribution of the next measurement  $d_i + 1$ . Since  $d_i + 1 \leq d$ , the sub-vine describing the dependence of events 1 to  $d_i + 1$  can be extracted from the full model. We consider the conditional distribution function  $F_{d_i+1|1:d_i}^i(\cdot | y_1^i, \dots, y_{d_i}^i)$ . Joe (1996) was the first to show that there exists a recursive representation for such conditional distribution functions. This way one obtains a closed-form expression of the conditional distribution function solely based on the pair-copulas specified in the D-vine (and the univariate marginals, of course) if the variable to be predicted is a leaf in the first tree. In our case,  $d_i + 1$  is in fact a leaf in the first tree of the D-vine on nodes 1 to  $d_i + 1$ . Thus, we know  $F_{d_i+1|1:d_i}^i(\cdot | y_1^i, \dots, y_{d_i}^i)$  analytically and can further simulate from it, which can be very useful for practical application. As an example, we can express the univariate conditional distribution function  $F_{4|1,2,3}^i$  in the following way:

$$F_{4|1,2,3}^i(y_4^i | y_1^i, y_2^i, y_3^i) = C_{4|1,2,3} \left( C_{4|2,3} \left( C_{4|3} (F_4^i(y_4^i) | F_3^i(y_3^i)) | C_{2|3} (F_2^i(y_2^i) | F_3^i(y_3^i)) \right) | C_{1|3,2} \left( C_{1|2} (F_1^i(y_1^i) | F_2^i(y_2^i)) | C_{3|2} (F_3^i(y_3^i) | F_2^i(y_2^i)) \right) \right).$$

Further, being the inverse of the conditional distribution function, the conditional quantile function can be expressed in general as

$$\begin{aligned} q_\alpha(y_1^i, \dots, y_{d_i}^i) &= (F_{d_i+1|1:d_i}^i)^{-1}(\alpha | y_1^i, \dots, y_{d_i}^i) \\ &= (F_{d_i+1}^i)^{-1} \left( C_{d_i+1|1:d_i}^{-1}(\alpha | F_1^i(y_1^i), \dots, F_{d_i}^i(y_{d_i}^i)) \right) \end{aligned} \quad (5.6)$$

and is of great interest in order to determine upper and lower bounds of a prediction interval. Kraus and Czado (2017a) show that inversion also yields a closed-form expression for the conditional quantile function solely based on the specified pair-copulas and marginals. Thus, we can determine arbitrary conditional quantiles for the size of measurement  $d_i + 1$ . For example, for financial applications it might be interesting to obtain a conditional 99%-value-at-risk, i.e. the conditional 99%-quantile, for the size of individual  $i$ 's next measurement.

In order to be able to perform statistical inference of any kind with our D-vine based model we first have to estimate the pair-copula families and associated parameters. Section 5.4 will present two estimation approaches. First, however, we will introduce linear mixed models and illustrate how they are connected to our proposed D-vine based model in Section 5.3.

### 5.3. Connection between the D-vine based model and linear mixed models

Probably the most popular models for longitudinal data are linear mixed models. In this section we will give a short introduction to this model class and show how they are connected to our approach from Section 5.2.

#### 5.3.1. Linear mixed models for repeated measurements

Linear mixed models have been discussed in detail by many authors, e.g. in Diggle (2002), Verbeke and Molenberghs (2009) and Fahrmeir et al. (2013). Describing the outcome of repeated measurements  $j$ ,  $j = 1, \dots, d_i$ , for individuals  $i$ ,  $i = 1, \dots, n$ , as responses  $Y_j^i$ , they extend linear models by including random effects  $\boldsymbol{\gamma}_i \in \mathbb{R}^q$  to the fixed (i.e. non-random) effects  $\boldsymbol{\beta} \in \mathbb{R}^p$ ,  $p, q \in \mathbb{N}$ . These random effects, unlike the fixed effects, are different for each individual. The covariate vectors  $\mathbf{x}_{i,j} \in \mathbb{R}^p$  and  $\mathbf{z}_{i,j} \in \mathbb{R}^q$  are associated to the fixed and random effects, respectively.

For  $i = 1, \dots, n$  and  $j = 1, \dots, d_i$ , the  $j$ th measurement for individual  $i$  is assumed to decompose to

$$Y_j^i = \mathbf{x}_{i,j}^\top \boldsymbol{\beta} + \mathbf{z}_{i,j}^\top \boldsymbol{\gamma}_i + \varepsilon_{i,j}, \quad (5.7)$$

where the vector of random effects  $\boldsymbol{\gamma}_i \sim \mathcal{N}_q(\mathbf{0}, D)$  is normally distributed with zero expectation and covariance matrix  $D \in \mathbb{R}^{q \times q}$  and the error vector  $\boldsymbol{\varepsilon}_i = (\varepsilon_{i,1}, \dots, \varepsilon_{i,d_i})^\top \sim \mathcal{N}_{d_i}(\mathbf{0}, \Sigma_i)$  also follows a centered normal distribution with covariance matrix  $\Sigma_i \in \mathbb{R}^{d_i \times d_i}$ . Further,  $\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_n$ ,  $\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_n$  are assumed to be independent. Hence,

$$Y_j^i \sim \mathcal{N}(\mathbf{x}_{i,j}^\top \boldsymbol{\beta}, \phi_{i,j}^2) \quad (5.8)$$

with standard deviation  $\phi_{i,j} := (\mathbf{z}_{i,j}^\top D \mathbf{z}_{i,j} + \sigma_{i,j}^2)^{1/2}$ , where  $\sigma_{i,j}^2 := \text{Var}(\varepsilon_{i,j})$ . Using the notation

$$X_i := \begin{pmatrix} \mathbf{x}_{i,1}^\top \\ \vdots \\ \mathbf{x}_{i,d_i}^\top \end{pmatrix} \in \mathbb{R}^{d_i \times p}, \quad Z_i := \begin{pmatrix} \mathbf{z}_{i,1}^\top \\ \vdots \\ \mathbf{z}_{i,d_i}^\top \end{pmatrix} \in \mathbb{R}^{d_i \times q}, \quad \mathbf{Y}^i := \begin{pmatrix} Y_1^i \\ \vdots \\ Y_{d_i}^i \end{pmatrix} \in \mathbb{R}^{d_i}$$

we can represent the vector of all measurements belonging to individual  $i$  as follows:

$$\mathbf{Y}^i = X_i \boldsymbol{\beta} + Z_i \boldsymbol{\gamma}_i + \boldsymbol{\varepsilon}_i. \quad (5.9)$$

We see that due to the independence assumptions of  $\boldsymbol{\gamma}_i$  and  $\boldsymbol{\varepsilon}_i$ ,  $i = 1, \dots, n$ , there ex-

### 5.3. Connection between the D-vine based model and linear mixed models

ists a correlation between measurements of one individual but measurements of different individuals are independent. Further, the joint distribution of  $\mathbf{Y}^i$  can be determined to be

$$\mathbf{Y}^i \sim \mathcal{N}_{d_i}(X_i\boldsymbol{\beta}, Z_i D Z_i^\top + \Sigma_i) \quad (5.10)$$

and  $\mathbf{Y}^1, \dots, \mathbf{Y}^n$  are independent. The fixed effects  $\boldsymbol{\beta}$  and random effects  $\boldsymbol{\gamma}_i$  as well as the parameters of the covariance matrices  $D$  and  $\Sigma_i, i = 1, \dots, n$ , can be estimated using (restricted) maximum-likelihood estimation as described for example in Diggle (2002) and Fahrmeir et al. (2013).

Linear mixed models are very popular in practice since they are easy to handle and interpret. Further, observations with missing data can also be used for ML estimation as long as the values are missing at random (see e.g. McCulloch et al., 2011; Ibrahim and Molenberghs, 2009).

#### 5.3.2. Aligning linear mixed models and the D-vine based approach

Equation 5.10 implies that all univariate marginal distributions are normal distributions. Further, the dependence structure is Gaussian and can vary from individual to individual since the correlation matrix  $R_i$  of  $\mathbf{Y}^i$  is given by

$$R_i := \text{Cor}(\mathbf{Y}^i) = \text{diag}(\phi_{i,1}^{-1}, \dots, \phi_{i,d_i}^{-1}) (Z_i D Z_i^\top + \Sigma_i) \text{diag}(\phi_{i,1}^{-1}, \dots, \phi_{i,d_i}^{-1}),$$

where  $\phi_{i,j}$  is the standard deviation of  $Y_j^i, j = 1, \dots, d_i, i = 1, \dots, n$ . In practice, however, this would make estimation infeasible since the number of parameters would be too large; in many cases one would even have more parameters than observations. Therefore, structural assumptions are made, especially for  $\Sigma_i \in \mathbb{R}^{d_i \times d_i}$ , in order to reduce the number of parameters to be estimated.

In Section 5.2.2 we assumed that the dependence structure is basically the same for all individuals and only differs due to the number of measurements  $d_i$  that individual  $i$  has had so far. In order to obtain the same for linear mixed models, we simply have to require the following homogeneity condition:

**Homogeneity condition:** We call correlation matrices  $R_i$  *homogeneous* if they are the same for all individuals  $i = 1, \dots, n$  except for the dimension, i.e.  $R_i = (r_{k,\ell})_{k,\ell=1}^{d_i} \in \mathbb{R}^{d_i \times d_i}$  is a  $(d_i \times d_i)$ -submatrix of a correlation matrix  $R = R_d = (r_{k,\ell})_{k,\ell=1}^d \in \mathbb{R}^{d \times d}$ .

This condition is in particular fulfilled if the covariance matrices of the errors  $\Sigma_i \in \mathbb{R}^{d_i \times d_i}$  and the design matrices of the random effects  $Z_i \in \mathbb{R}^{d_i \times q}$  are constant in  $i$  except for the dimension. Despite being a restriction, linear mixed models meeting this requirement still comprise a wide range of models used in practice. The assumption on the covariance

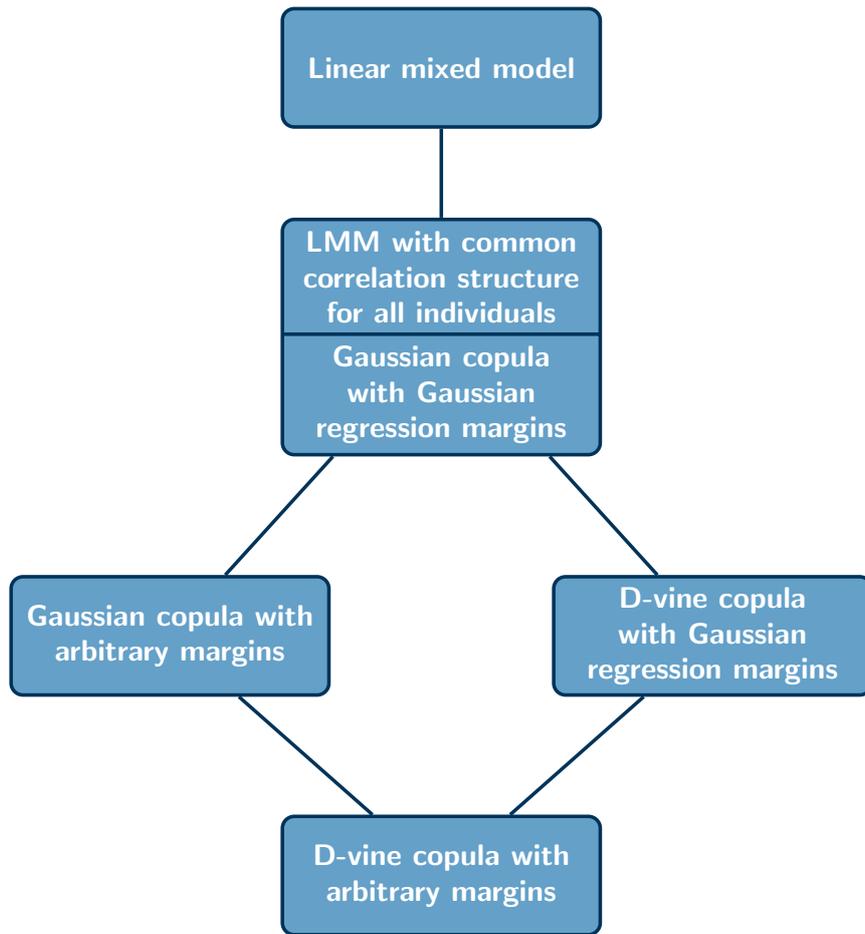
matrices  $\Sigma_i$  is for example fulfilled if errors

- are assumed to be i.i.d., i.e. the  $(k, \ell)$ th entry of  $\Sigma_i$  is given by  $\sigma^2 \mathbf{1}_{\{k=\ell\}}$ , where  $\mathbf{1}_{\{\cdot\}}$  denotes the indicator function;
- exhibit a compound symmetry structure, i.e. the  $(k, \ell)$ th entry of  $\Sigma_i$  is  $\sigma^2 \rho \mathbf{1}_{\{k \neq \ell\}}$  for some  $\rho \in (-1, 1)$ ;
- follow an autoregressive structure of order 1 (AR(1)), i.e. the  $(k, \ell)$ th entry of  $\Sigma_i$  is given by  $\sigma^2 \rho^{|k-\ell|}$  for some  $\rho \in (-1, 1)$ ;
- have an exponential decay structure, i.e. the  $(k, \ell)$ th entry of  $\Sigma_i$  can be written as  $\sigma^2 \exp\{-|k-\ell|/r\}$ , where  $r > 0$  is the constant “range” parameter.

These are typical simplifications that are made anyway for modeling longitudinal data in most applications if the number of individuals is large with respect to the number of measurements. The assumption on the design matrices  $Z_i$  is also often satisfied, e.g. for the popular class of so-called *random intercept models*, where  $Z_i = (1, \dots, 1)^\top \in \mathbb{R}^{d_i \times 1}$  for  $j = 1, \dots, d_i$  and  $i = 1, \dots, n$ . Further, the assumption includes any model where the covariates associated with the random effect only depend on the (common) measurement times  $t_j$ ,  $j = 1, \dots, d$ , i.e. for example  $Z_i = (t_1, \dots, t_{d_i})^\top \in \mathbb{R}^{d_i \times 1}$  or more generally  $Z_i = (h(t_1), \dots, h(t_{d_i}))^\top \in \mathbb{R}^{d_i \times 1}$  for some function  $h: \mathbb{R} \rightarrow \mathbb{R}$ . Thus, assuming that  $Z_i$  only depends on the number of measurements  $d_i$  for individual  $i$  is also not uncommon such that there is in fact a wide class of linear mixed models sharing the property that the correlation matrix  $R_i$  of  $\mathbf{Y}^i$  only depends on the number of measurements.

If  $R_i$  is homogeneous in  $i$ , we have that all individuals  $i$  share the same Gaussian dependence structure, i.e. correlation matrix. This scenario is a special case of the D-vine based model since we can represent any Gaussian correlation matrix using a D-vine with Gaussian pair-copulas and the corresponding (partial) correlations as parameters (see for example Stöber et al., 2013, Theorem 4.1). The univariate margins  $F_j^i$  can be chosen arbitrarily for the copula approach such that we can simply use  $\mathcal{N}(\mathbf{x}_{i,j}^\top \boldsymbol{\beta}, \phi_{i,j}^2)$ -margins (cf. Equation 5.8) to end up with a model describing the same joint distribution of  $\mathbf{Y}^i$  as the corresponding linear mixed model (Equation 5.10). Since we can use arbitrary distributions for the margins and/or any D-vine copula for the dependence structure, our approach can be seen as an extension of linear mixed models with common correlation structure for all individuals. Figure 5.2 illustrates the link between our D-vine based model and linear mixed models.

For the application in Section 5.6 we will compare how well both model classes perform fitting real life data.



**Figure 5.2.:** Flow chart illustrating how the D-vine based model is linked to linear mixed models.

## 5.4. Estimation methods for the D-vine based model

### 5.4.1. Marginal modeling

Although our focus is on dependence modeling, we will briefly discuss how the univariate marginal models for  $Y_j^i$ , i.e.  $F_j^i$ , can be estimated. In general the choice of marginal models is completely arbitrary. They can be parametric or non-parametric. The most common situation for repeated measurements is that in addition to the measurement data  $\mathcal{Y}$  itself further covariates are known for each individual and measurement. Therefore, regression models such as linear (LMs) or generalized linear models (GLMs) can be applied. In this case,  $F_j^i(\cdot) = F_j(\cdot | \mathbf{x}_{i,j})$  depends on the individual's associated covariates  $\mathbf{x}_{i,j} \in \mathbb{R}^p$ , where  $p \in \mathbb{N}$  is the number of covariates used in the model. In our application (Section 5.6) we will simply use the margins implied by the competing linear mixed model making the models better comparable. Our focus, however, is rather to develop a flexible model

describing the dependence structure that is present in the data  $\mathcal{Y}$  such that we will not further elaborate on how to estimate the univariate marginal distributions.

### 5.4.2. Dependence modeling

Assume we have estimated the marginal distributions and obtained (pseudo-)copula data by applying the estimated distribution functions  $\hat{F}_j^i$  to the measurements, i.e.  $\hat{u}_j^i := \hat{F}_j^i(y_j^i)$ . We now use the transformed data as a copula sample to estimate the underlying dependence structure. Section 5.2 has shown that *D*-vine copulas are suited for modeling the dependence structure being present in repeated measurement data. Model B (Equation 5.3) was preferable since it is easier to interpret and estimate. Further, predictions for not yet observed measurements can be made. The aim of the methods presented in Section 5.4 is to find estimates for the set of pair-copula families  $\mathcal{C} = (c_{k,k+\ell;(k+1):(k+\ell-1)} \mid k = 1, \dots, d - \ell \text{ and } \ell = 1, \dots, d - 1)$  and the set of parameters  $\boldsymbol{\theta} = (\boldsymbol{\theta}_{k,k+\ell;(k+1):(k+\ell-1)} \mid k = 1, \dots, d - \ell \text{ and } \ell = 1, \dots, d - 1)$  corresponding to Model B from Section 5.2, where  $d$  is the maximal number of observed events per observation. We will present two approaches: a standard joint maximum-likelihood estimator and a sequential method. Since we want to choose both parameters and families for each pair-copula we will select from a set of  $m$  bivariate candidate family types  $\Gamma = \{\gamma_1, \dots, \gamma_m\}$ , where each member  $\gamma \in \Gamma$  has its own space of admissible parameter values  $\Omega(\gamma)$ .

#### Joint maximum-likelihood approach

The canonical approach in order to find optimal pair-copula families and parameters would be to use maximum-likelihood estimation. In Section 5.2 we have already determined the log-likelihood. Since the families specify which parameter values are admissible, finding the optimal families and parameter estimates is divided into two steps: For each combination of families we have to determine the maximum-likelihood estimate of parameters; then we select the one combination with the overall highest likelihood. This way we find the best *D*-vine model with regards to likelihood optimization. However, since there are  $|\Gamma| = m$  candidates for each of the  $d(d - 1)/2$  pair-copula families, we have to perform  $m^{d(d-1)/2}$  times an at least  $(d(d - 1)/2)$ -dimensional optimization (some families like the *t*-copula may have more than one parameter). It is obvious that this can very quickly become computationally infeasible if the number of candidate families  $m$  is high and—especially—if the dimension  $d$  gets large.

Of course, the possibly large number of parameters to be estimated is a general problem in the statistical analysis of vine copulas. Therefore, Aas et al. (2009) (for *D*-vines) and later Dißmann et al. (2013) (for general vine copulas) developed a sequential tree-by-tree

selection algorithm facilitating vine copula model estimation up to very high dimensions. Dißmann’s algorithm is commonly used to fit the vine’s model structure, pair-copula families and parameters but it can also only be used for the selection of families and parameters only if we have a fixed tree structure (e.g. a D-vine). The difference to the classical situation which we face when we want to estimate a vine copula is that our observations have different lengths.

### Sequential approach

Inspired by Dißmann’s algorithm we want to fit the pair-copula families and the associated parameters of the D-vine to a repeated measurement data set using a sequential approach. Given classical data, Dißmann’s algorithm starts with the estimation of the first tree and estimates the unconditional pair-copulas (and their parameters) via maximum-likelihood estimation. Then the observations are transformed into pseudo-observations needed for the estimation of the second tree using the estimated pair-copulas of tree 1. Continuing this way the vine is built up tree-by-tree.

In the presence of repeated measurement data, however, we can pursue a very similar strategy. The only difference is that we estimate each pair-copula (and its parameter(s)) only based on the available full observation. All pair-copulas to be estimated are of the form  $c_{k,\ell;(k+1):(\ell-1)}$  with parameter  $\boldsymbol{\theta}_{k,\ell;(k+1):(\ell-1)}$ . When all observations are of the form  $(u_1, \dots, u_j)^\top$  for some  $j \in \{1, \dots, d\}$ , i.e. there are no “gaps” between two observed events, we can use the information of observations with a minimum length of  $\ell$ , i.e. all observations in  $\bigcup_{j=\ell}^d \mathcal{U}^j$ , for the estimation of  $c_{k,\ell;(k+1):(\ell-1)}$  and  $\boldsymbol{\theta}_{k,\ell;(k+1):(\ell-1)}$ . Thus, we can maintain the basic scheme known from Dißmann’s algorithm. With a slight modification of the data we are even able to use the function `RVineCopSelect` from the R library `VineCopula` (Schepsmeier et al., 2017) for our purpose, making our approach also very appealing from a practitioner’s point of view.

Of course, this sequential approach can also be applied for data with missing values (Section 5.2.2, page 84). Then, for the estimation of each pair-copula  $c_{k,\ell;(k+1):(\ell-1)}$  with associated parameter  $\boldsymbol{\theta}_{k,\ell;(k+1):(\ell-1)}$  is performed using all observations for whom the necessary measurements  $u_k, u_{k+1}, \dots, u_\ell$  are available. The function `RVineCopSelect` can still be used in the presence of missing values.

The biggest advantage of being able to use sequential estimation approach is that we can estimate models at reasonable computational costs, even in high dimensions. Of course, the approach also works when using non-parametric pair-copulas or even non-simplified vine copulas. For details for estimating non-parametric and non-simplified vines we refer the reader to Nagler and Czado (2016) and Vatter and Nagler (2016), respectively. Yet, as already mentioned at the beginning, we focus on parametric simplified vine copulas here.

### 5.4.3. Model selection

In model selection one often wants to compare different fitted models. For this purpose the log-likelihood and log-likelihood based measures such as AIC (Akaike, 1998) and BIC (Schwarz, 1978), which penalize large numbers of parameters, are frequently applied. Whereas the penalty of the AIC only depends on the number of parameters in the model, that of BIC also depends on the sample size. In our case, however, it is not completely obvious what sample size to use. Therefore, we derive how the BIC for the D-vine based model including margins can be calculated in our situation. Proposition 5.1 shows that each parameter is to be weighted with the (logarithm of) the number of observations that directly contribute to its estimation. A proof can be found in Appendix C.1.

**Proposition 5.1.** Let  $p_j \in \mathbb{N}$  be the number of parameters of the D-vine based model including margins restricted to the measurements 1 to  $j$ ,  $j = 1, \dots, d$ , and define  $\Delta p_j := p_j - p_{j-1}$  for  $j = 2, \dots, d$  and  $\Delta p_1 := p_1$ . Further denote by  $N_j = \sum_{k=j}^d n_k$  the number of individuals with at least  $j$  measurements. The BIC of the D-vine based model including margins is given by

$$\text{BIC} = -2 \log L(\hat{\boldsymbol{\theta}} | \mathcal{Y}) + \sum_{j=1}^d \Delta p_j \log(N_j).$$

Here,  $\log L(\hat{\boldsymbol{\theta}} | \mathcal{Y}) = \log L(\hat{\boldsymbol{\theta}}_M | \mathcal{Y}) + \log L(\hat{\boldsymbol{\theta}}_C | \mathcal{U})$  is the log-likelihood of the fitted model including margins, i.e. the sum of the log-likelihood of the margins  $\log L(\hat{\boldsymbol{\theta}}_M | \mathcal{Y})$  and the one of the copula  $\log L(\hat{\boldsymbol{\theta}}_C | \mathcal{U})$  (which is the one of Model B from Section 5.2.2). Further,  $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\theta}}_M, \hat{\boldsymbol{\theta}}_C)$  is the maximum-likelihood estimate for the set of all model parameters (associated with both the margins  $\hat{\boldsymbol{\theta}}_M$  and the D-vine copula  $\hat{\boldsymbol{\theta}}_C$ ).

**Remark 5.2.** Although this BIC adjustment was developed for the D-vine based model, it can also be used for certain types of linear mixed models due to the connection described in Section 5.3. For LMMs fulfilling the homogeneity condition the BIC can be determined with the formula from Proposition 5.1 if only individuals with  $j$  or more measurements contribute to the estimation of the parameters of the sub-model restricted to the first  $j$  measurements which were not already contained in the sub-model restricted to the first  $j - 1$  measurements. This is for example the case if on the one hand no structural assumptions (besides homogeneity) are imposed on the covariance matrices of the random effects and the errors  $D$  and  $\Sigma_i$  and on the other hand the design matrices  $X_i$  have a form that allows for different marginal regression models for different measurements. For guaranteeing the latter each covariate is only allowed to be incorporated in one of the marginal regressions, i.e. the values of this covariate are zero for all other measurements; if a covariate still is to be included in more than one model, one simply splits up the covariate into several measurement-specific covariates that are non-zero only for one particular measurement.

This way for one covariate a separate coefficient can be estimated for different marginal models (if necessary).

## 5.5. Simulation study

In order to check that the sequential estimation approach from Section 5.4 works reasonably well, we perform a simulation study that is inspired by the data analyzed in Section 5.6.

### Simulation setting

For a maximum number of measurements  $d \in \{5, 10\}$ , we generate  $d$ -dimensional data sets and prune them randomly to obtain an unbalanced setting. In this context pruning means that for each  $d$ -dimensional observation  $i$  we independently draw  $d_i$  from a discrete distribution on  $\{2, \dots, d\}$  and restrict this observation to its first  $d_i$  components. This way we mimic the nature of unbalanced repeated measurement data. In order to assess the implications of having only incomplete data we sequentially fit a D-vine copula to both the full and the pruned data set and compare the estimates.

To obtain data sets we consider randomly generated D-vine copulas with structure 1–2– $\dots$ – $d$ . For this purpose, we rely on the method proposed in Joe (2006) to sample Gaussian correlation matrices that are uniformly distributed over the space of valid correlation matrices. Conveniently, this method is already based on a vine decomposition: For each tree  $i$ ,  $i = 1, \dots, d - 1$ , we generate the corresponding  $d - i$  parameters associated to the Gaussian pair-copulas by drawing from a  $\text{Beta}((d - i + 1)/2, (d - i + 1)/2)$  distribution and transforming the outcome linearly to  $[-1, 1]$ , resulting in a mean and mode of 0 and a variance of  $1/(d - i + 2)$ . However, since we do not only want to consider Gaussian D-vines, we transform the Gaussian parameters to Kendall's  $\tau$  values using the relationship  $\tau = \frac{2}{\pi} \arcsin(\rho)$ . Then, we randomly draw a pair-copula family for each pair-copula to be specified<sup>7</sup> and convert the Kendall's  $\tau$  values to parameters of the respective families. For one-parametric families  $\tau$  can directly be transformed to the parameter space. For two-parametric families there are infinitely many combinations of parameters resulting in the same Kendall's  $\tau$  value. Therefore, we adopt the approach used in Kraus and Czado (2017b): draw the second parameter randomly<sup>8</sup> and determine the first parameter implicitly such that the two parameters imply the required Kendall's  $\tau$ .

With the above procedure we generate  $R = 1000$  D-vine copulas and simulate data sets of size  $n \in \{200, 2000\}$ . Then for each observation  $i$  we randomly determine its length

---

<sup>7</sup>The families are drawn uniformly from the ones available in the library `VineCopula`: Gaussian, t, Clayton, Gumbel, Frank, Joe, BB1, BB6, BB7, BB8 and Tawn as well their rotations (see Schepsmeier et al., 2017, for details).

<sup>8</sup>The specific sampling distributions can be found in Appendix B of Kraus and Czado (2017b).

$d_i \in \{2, \dots, d\}$ . For  $d = 5$ , the underlying distribution mimics the observed measurement rates of the data considered in Section 5.6. The exact proportions of individuals with a least  $j$  measurements would have been 100.0%, 78.5%, 58.5%, 43.9% for  $j = 2, 3, 4, 5$ , respectively. For  $d = 10$ , we extended the scenario of  $d = 5$  accordingly. The distributions are given in Table 5.2.

$j$	2	3	4	5
probability of $d_i = j$	20%	20%	15%	45%
probability of $d_i \geq j$	100%	80%	60%	45%

$j$	2	3	4	5	6	7	8	9	10
probability of $d_i = j$	10%	10%	10%	10%	10%	5%	5%	5%	35%
probability of $d_i \geq j$	100%	90%	80%	70%	60%	50%	45%	40%	35%

**Table 5.2.:** Probability mass function and proportions of individuals with at least  $j$  measurements for the “pruning distribution” (top table:  $d = 5$ ; bottom table:  $d = 10$ ).

For both the full and the pruned data set we use the sequential algorithm implemented in `RVineCopSelect` (from the `VineCopula` library) to fit D-vine copulas. In order to assess how badly the loss of information affects the estimation we compare the resulting D-vines by considering each pair-copula separately. For this purpose, we consider the mean absolute difference between the Kendall’s  $\tau$  values ( $\Delta_\tau := \frac{1}{R} \sum_{r=1}^R |\hat{\tau}_{\text{pruned}}(r) - \hat{\tau}_{\text{full}}(r)|$ ), the lower and the upper tail dependence coefficients ( $\Delta_{\lambda^s} := \frac{1}{R} \sum_{r=1}^R |\hat{\lambda}_{\text{pruned}}^s(r) - \hat{\lambda}_{\text{full}}^s(r)|$  for  $s \in \{\ell, u\}$ ) of the two models. Comparing general strength of dependence and tail behavior enables us to assess how similar the fitted pair-copulas are.<sup>9</sup>

### Results for $d = 5$

For  $d = 5$ , the absolute differences of Kendall’s  $\tau$ , lower and upper tail dependence coefficient (averaged over the  $R = 1000$  data sets) are displayed for each of the 10 pair-copulas in Table 5.3, where the sample sizes are  $n = 200$  and  $n = 2000$ , respectively. For  $n = 200$ , the 10 average absolute estimated Kendall’s  $\tau$  values for the full data set ( $\frac{1}{R} \sum_{r=1}^R |\hat{\tau}_{\text{full}}(r)|$ ) lie between 0.345 and 0.394; the 10 average estimated upper and lower tail dependence coefficients for the 10 pair-copulas are between 0.075 and 0.108 ( $\frac{1}{R} \sum_{r=1}^R \hat{\lambda}_{\text{full}}^\ell(r)$ ) and 0.080 and 0.107 ( $\frac{1}{R} \sum_{r=1}^R \hat{\lambda}_{\text{full}}^u(r)$ ), respectively. For  $n = 2000$ , the three ranges are fairly similar:  $[0.338, 0.422]$ ,  $[0.081, 0.992]$  and  $[0.083, 0.107]$ , respectively.

We can see that even for a sample size of only  $n = 200$  (see upper part of Table 5.3) the differences between the two estimates are relatively small. The largest absolute deviations are 0.058, 0.066 and 0.067 for  $\tau$ ,  $\lambda^\ell$  and  $\lambda^u$ , respectively. The average absolute deviations 0.029 ( $\tau$ ), 0.039 ( $\lambda^\ell$ ) and 0.039 ( $\lambda^u$ ), respectively. Of course,  $c_{1,2}$  is always estimated

<sup>9</sup>Considering the percentage of cases where the same copula family is fitted would not be sensible since the number of candidate families is large and many of them, e.g. a Clayton and a survival Joe copula, can hardly be distinguished.

equally in both cases since all pruned observations have minimum length of 2. We can observe what one would expect given that the number of observations with at least  $j$  measurements descends in  $j$ : Pair-copulas for whose estimation later measurements are needed exhibit larger deviations.

		$c_{1,2}$	$c_{2,3}$	$c_{3,4}$	$c_{4,5}$	$c_{1,3;2}$	$c_{2,4;3}$	$c_{3,5;4}$	$c_{1,4;2,3}$	$c_{2,5;3,4}$	$c_{1,5;2,3,4}$
$n = 200$	$\Delta_\tau$	0.000	0.016	0.026	0.035	0.017	0.027	0.036	0.032	0.043	0.058
	$\Delta_{\lambda^\ell}$	0.000	0.018	0.035	0.050	0.024	0.039	0.053	0.044	0.066	0.065
	$\Delta_{\lambda^u}$	0.000	0.022	0.030	0.041	0.024	0.037	0.058	0.048	0.067	0.065
$n = 2000$	$\Delta_\tau$	0.000	0.005	0.007	0.010	0.005	0.008	0.010	0.008	0.011	0.015
	$\Delta_{\lambda^\ell}$	0.000	0.004	0.008	0.012	0.009	0.011	0.016	0.008	0.021	0.015
	$\Delta_{\lambda^u}$	0.000	0.004	0.010	0.011	0.006	0.008	0.013	0.011	0.021	0.024

**Table 5.3.:** Absolute differences of Kendall's  $\tau$ , lower and upper tail dependence coefficient for each of the 10 pair-copulas, averaged over the  $R = 1000$  data sets of size  $n = 200$  and  $n = 2000$ , respectively.

The results in the lower part of Table 5.3 (corresponding to  $n = 2000$ ) show a similar qualitative behavior. However, the overall level of average absolute deviations is even smaller: Maximum/average values are 0.015/0.008, 0.024/0.011 and 0.021/0.010 for  $\tau$ ,  $\lambda^\ell$  and  $\lambda^u$ , respectively.

### Results for $d = 10$

We performed the same studies as above for  $d = 10$ . Since it does not make sense to display the results for all 45 pair-copulas separately, we only report some summary statistics<sup>10</sup>. For a sample size of  $n = 200$  the maximum/average deviations were 0.091/0.048 ( $\tau$ ), 0.069/0.044 ( $\lambda^\ell$ ) and 0.068/0.043 ( $\lambda^u$ ); for  $n = 2000$  we observed 0.059/0.018 ( $\tau$ ), 0.037/0.017 ( $\lambda^\ell$ ) and 0.037/0.017 ( $\lambda^u$ ). In comparison to the results for  $d = 5$  we detect an increase in deviation, which seems plausible since the dimension of the model increases but the sample sizes are kept constant.

All in all, we see that the sequential fitting of D-vine models to repeated measurement data performs well such that we do not have to hesitate to use it for the real data application in Section 5.6.

## 5.6. Application

In Section 5.5 we have seen that our proposed estimation method performs satisfactory. Now we will apply it to real life data. For this purpose, we consider the aortic valve replacement surgery data set `heart.valve` that is taken from the R library `joiner` (Philipson

<sup>10</sup>The detailed results are of course available on request from the authors.

et al., 2017) and has been analyzed in Lim et al. (2008). For this longitudinal study the regression of the left ventricular mass index (LVMI) of  $n = 256$  individuals was examined in several follow-up appointments after the surgery, where a new heart valve had been implanted. The total number of examinations is 988 such that the average number of measurements per patient is 3.86, where 10 is the maximum. Table 5.4 summarizes the sizes of the groups of individuals with exactly  $j$  and  $j$  or more measurements, respectively,  $j = 1, \dots, 10$ .

$j$	1	2	3	4	5	6	7	8	9	10
patients with $j$ measurements	51	44	41	30	27	15	21	15	6	6
patients with $\geq j$ measurements	256	205	161	120	90	63	48	27	12	6

**Table 5.4.:** Sizes of the groups of individuals with exactly  $j$  and  $j$  or more measurements, respectively,  $j = 1, \dots, 10$ .

Besides the examination results, for every patient and measurement there are also covariates available. We denote them the way they are stored in the data set `heart.valve`. The following list contains the covariates that we used in our final models as well as a short description, which is basically taken from the documentation of the `joiner` library (Philipson et al., 2017):

- **size:** size of the heart valve in millimeters;
- **sex:** gender of the patient
- **bsa:** body surface area (preoperative)
- **time:** date of measurement (with surgery date as time origin)

The quantity we model is the logarithm of the LMVI. We estimate two different models: a linear mixed model and a D-vine copula based model. We focus on the first five measurements since there are rather few observations for the later measurements. This way we use 832 of the 988 available measurements (84.2%).

### Linear mixed model approach

In order to fit a linear mixed model (cf. Section 5.3.1) to the data we use the function `lme` from the R library `nlme` (Pinheiro et al., 2017). Assuming a homogeneous covariance structure for all individuals  $i$ ,  $i = 1, \dots, 256$ , different correlation structures such as i.i.d. errors, compound symmetry or AR(1) can be selected (cf. Section 5.3, page 88). We fit a random intercept model, i.e.  $Z_i = (1, \dots, 1)^\top \in \mathbb{R}^{d_i \times 1}$ , and compare different (homogeneous) correlation structures for the error terms, namely i.i.d., compound symmetry,

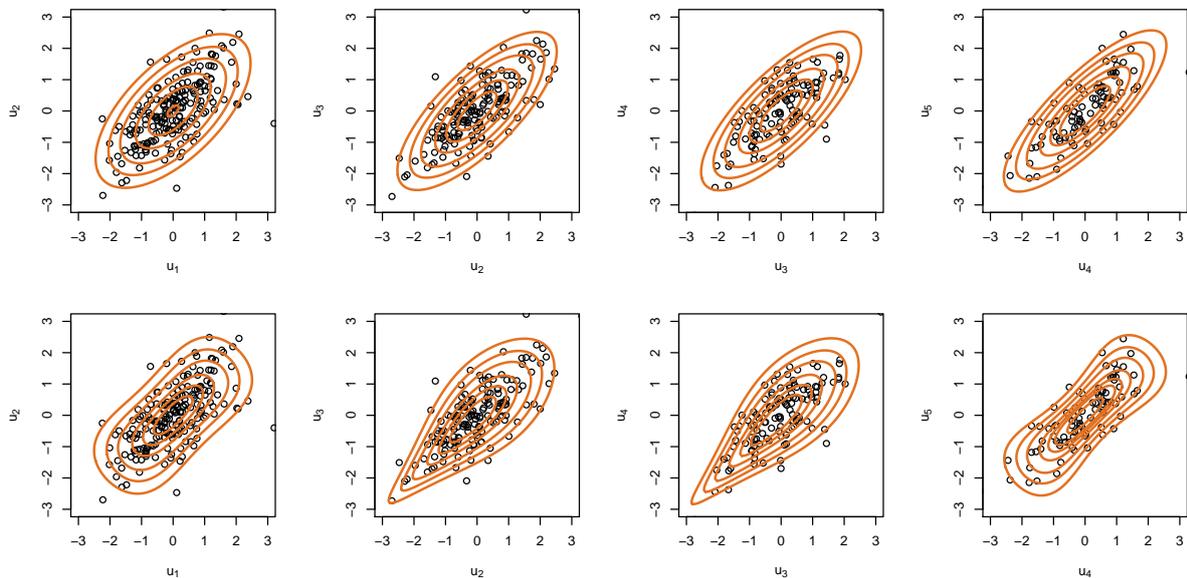
AR(1), exponential decay and a general (i.e. unrestricted) structure. Note that we perform classical maximum-likelihood estimation (instead of restricted maximum-likelihood estimation, which is often used for linear mixed models) since we need to be able to compare the quality of the fit to our D-vine based approach later. The parameter estimates, however, are almost the same for both methods. The best model with respect to log-likelihood and AIC is the one with the general structure; it contains the covariates `size`, `sex` and `bsa` as well as an `intercept` as fixed effects. The AR(1) error structure, where the  $(k, \ell)$ th entry of  $\Sigma_i$  is given by  $\sigma^2 \rho^{|k-\ell|}$ , is more parsimonious than the general structure and exhibits a better BIC although log-likelihood and AIC are worse; it contains the covariates `size`, `sex`, `bsa` and `time` as well as an `intercept`. Note that the use of BIC for linear mixed models is controversial (Hedeker and Gibbons, 2006); it is frequently debated which sample size to use for the calculation of BIC (Jones, 2011; Müller et al., 2013; Delattre et al., 2014). In the penalty of the standard BIC all model parameters are weighted with the logarithm of the total number of observations. Here, we used the adjusted BIC for linear mixed models that was developed by Delattre et al. (2014) and that is better comparable to the one we derived for our approach in Proposition 5.1, where each parameter is weighted with the logarithm of the number of observations that directly contribute to its estimation. In the adjusted penalty term of Delattre et al. (2014) the parameters associated with the fixed effects are weighted with the logarithm of the number of measurements and the parameters associated with the random effects are weighted with the logarithm of the number of individuals.

The log-likelihood, AIC and BIC values of the models with the general and the AR(1) structure can be found in Table 5.5. The remaining structures are not listed there as they performed uniformly worse than the two models.

### D-vine copula based approach

As an alternative we will also fit our D-vine based model. As described in Section 5.2.1 we first deal with the univariate marginal distributions and afterwards estimate the dependence structure. For the marginals we use the univariate marginal regression model that was already estimated for the linear mixed model with AR(1) error correlation structure. Hence, the margins depend on the covariates `size`, `sex`, `bsa` and `time`. In order to transform the measurements to the uniform scale, we apply the estimated normal distribution functions resulting from the regression model (cf. Equation 5.8). Then a D-vine copula with order 1–2–3–4–5 is fitted to the transformed observations according to the sequential approach from Section 5.4.2 (using `RVineCopSelect`). In order to avoid unnecessary parameters we apply a Kendall's  $\tau$  based independence test (significance level  $\alpha = 5\%$ ), which is also implemented in `RVineCopSelect`, to decide for each pair-copula if it is significantly different from an independence copula (for a detailed description see

Hollander et al., 2014; Genest and Favre, 2007). The criterion for the selection of the pair-copula families is standard BIC. We fit both a Gaussian D-vine copula, where all pairs are assumed to be bivariate Gaussian, and a general, unrestricted D-vine copula. The result is in both cases a first-order Markov structure, also known as a 1-truncated vine copula, i.e. all pair-copulas in the second, third and fourth tree are the independence copula. For the Gaussian vine the Kendall's  $\tau$  values of the Gaussian pairs in the first tree are estimated to be  $\hat{\tau}_{1,2} = 0.43$ ,  $\hat{\tau}_{2,3} = 0.54$ ,  $\hat{\tau}_{3,4} = 0.56$  and  $\hat{\tau}_{4,5} = 0.61$ . For the general D-vine copula the pair-copulas in the first tree are estimated to be the following:  $\hat{c}_{1,2} = \text{Frank}$  ( $\hat{\tau}_{1,2} = 0.49$ );  $\hat{c}_{2,3} = \text{Survival Gumbel}$  ( $\hat{\tau}_{2,3} = 0.53$ );  $\hat{c}_{3,4} = \text{Survival Gumbel}$  ( $\hat{\tau}_{3,4} = 0.56$ );  $\hat{c}_{4,5} = \text{Frank}$  ( $\hat{\tau}_{4,5} = 0.65$ ). Figure 5.3 displays pairwise plots of the copula data (transformed to standard normal margins for reasons of comparability) including the contour lines of the corresponding fitted pair-copulas.



**Figure 5.3.:** Pairwise plots of the copula data (transformed to standard normal margins) including the contours lines of the fitted pair-copulas of the Gaussian D-vine copula (upper panel) and the general D-vine copula (lower panel), respectively.

We see that there is a positive medium strength of dependence for all pairs in both models ( $\tau$ -values from 0.43 to 0.61 and 0.49 to 0.65, respectively). The shape of the contours, however, differs considerably between the two models. All four bivariate copulas in the general D-vine model are from different families and non-Gaussian. Whereas  $\hat{c}_{1,2}$  and  $\hat{c}_{4,5}$  show no tail dependence, the survival Gumbel copula modeling the dependence between variables 2 and 3 and 3 and 4, respectively, exhibits moderate lower tail dependence:  $\hat{\lambda}_{2,3}^{\ell} = 0.62$  and  $\hat{\lambda}_{3,4}^{\ell} = 0.64$ . The fact that the dependence between two consecutive measurements is not constant and non-Gaussian is an indicator that the general D-vine approach might be a better choice than a simple Gaussian dependence model.

### Model comparison

In order to see if this is the case we compare the fit of the two D-vine based models (including margins) and the two linear mixed models to the data using the log-likelihood (of the full model including margins), AIC and BIC. Table 5.5 displays all three model selection criteria. Note that the BIC values of the linear mixed models are calculated as proposed by Delattre et al. (2014) and the ones of the D-vine based models are calculated according to Proposition 5.1.

model	log-likelihood	AIC	BIC	# parameters
general LMM	-99.4	230.9	305.3	16
LMM AR(1)	-108.9	233.8	270.4	8
Gaussian D-vine	-102.7	225.3	265.3	10
general D-vine	<b>-85.0</b>	<b>190.1</b>	<b>230.1</b>	10

**Table 5.5.:** Log-likelihood, AIC and BIC values for the fitted linear mixed models with general and AR(1) structure and the Gaussian and the general D-vine based models (including margins). Bold values indicate the best model fit according to the respective model selection criteria.

One can see that the unrestricted D-vine model performs uniformly better than the Gaussian one. This is a clear indicator that the normality assumption for the dependence is not really suited. Nevertheless, due to their flexibility, both D-vine based models yield a considerably better fit than the two linear mixed models with respect to log-likelihood, AIC and BIC; only the log-likelihood of the Gaussian D-vine is slightly worse than the one of the general LMM. We see that the D-vine based approaches are able to capture the structure of the data better since the flexibility of the D-vine helps to fit the dependence structure more appropriately. This is especially important if the deviation from Gaussianity is strong.

### Quantile prediction

As a final application we illustrate how conditional quantiles for the  $(j + 1)$ st measurement of an individual with  $j$  measurements can be determined using the general D-vine copula based approach and the linear mixed model with AR(1) error structure. For this purpose, we select three representative individuals with  $d_i = 4$  measurements from the data set such that they have had rather low ( $\mathbf{y}^1 = (4.63, 4.62, 4.66, 4.91)^\top$ ), medium ( $\mathbf{y}^2 = (5.26, 5.13, 5.00, 5.19)^\top$ ) and high ( $\mathbf{y}^3 = (5.90, 5.80, 5.67, 5.31)^\top$ ) measurement values so far, respectively. We use the corresponding observed covariate values of  $\mathbf{x}^i = (x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4})^\top = (\mathbf{size}_i, \mathbf{sex}_i, \mathbf{bsa}_i, \mathbf{time}_i)^\top$  that are given by  $\mathbf{x}^1 = (29, 0, 1.93, 3.15)^\top$ ,  $\mathbf{x}^2 = (25, 0, 1.65, 5.48)^\top$  and  $\mathbf{x}^3 = (25, 0, 1.71, 3.19)^\top$ , respectively.

We pretend that the three selected individuals have only had three measurements so far. Then we predict the median, i.e. the 50% quantile, and a 90% prediction interval, i.e.

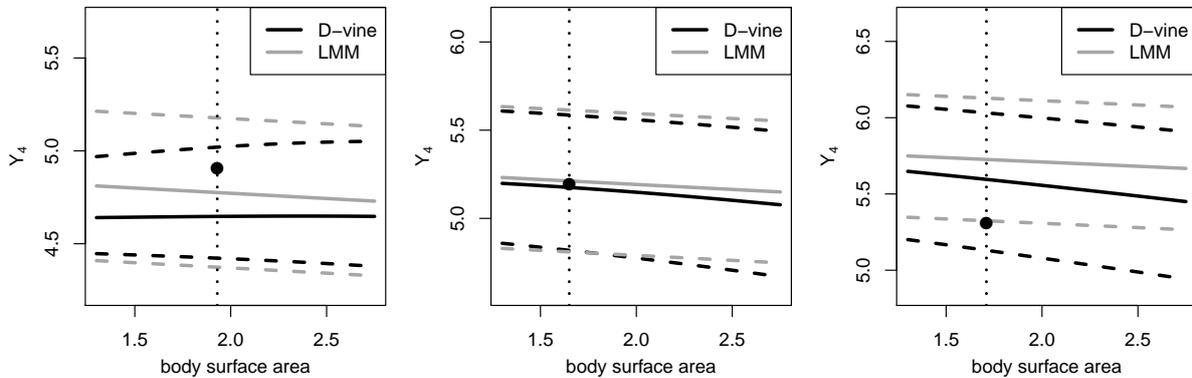
the 5% and the 95% quantile, of the fourth measurement based on the three measurements  $y_1^i, y_2^i, y_3^i$  for both models and compare the results to the true value of the fourth measurement.

For the linear mixed model we know that the joint distribution of the measurements of one individual is a multivariate normal distribution with mean and variance as in Equation 5.10. Having estimated the corresponding parameters we can easily determine the conditional distribution of the fourth measurement given the first three measurements, which is given by a univariate normal distribution (see for example Joe, 2014, Section 2.6). The quantiles of univariate normal distributions are known such that we can easily compute the desired quantities. The estimated median for individual 1 is 4.78, the 90% prediction interval is given by (4.37, 5.18). For individuals 2 and 3 the medians are 5.21 and 5.73 and the prediction intervals are (4.81, 5.61) and (5.32, 6.13), respectively. We see that the observed value of the fourth measurement of individuals 1 and 2 are inside the prediction intervals; for the third individual the prediction interval does not contain the observed measurement value:  $5.31 \notin (5.32, 6.13)$ . Note that due to the normality of the conditional distribution the prediction intervals are symmetric around the mean (which is also the median).

For comparison we determine the conditional quantiles using the general D-vine based model. Since we want to apply Equation 5.6, which uses the inverse conditional distribution function on the copula level  $C_{4|1:3}^{-1}(\cdot | F_1^i(y_1^i), F_2^i(y_2^i), F_3^i(y_3^i))$  and the inverse of the marginal distribution function of the fourth measurement  $(F_4^i)^{-1}(\cdot)$ , we first of all transform the measurements 1, 2 and 3 to the copula scale by  $\hat{u}_j^i := \hat{F}_j^i(y_j^i)$ , where  $\hat{F}_j^i$  are the marginal linear regression estimates obtained from the linear mixed model. For each individual  $i = 1, 2, 3$  and  $\alpha \in \{0.05, 0.50, 0.95\}$  we use the estimated general D-vine copula to calculate  $\hat{q}_u^i(\alpha) := \hat{C}_{4|1:3}^{-1}(\alpha | \hat{u}_1^i, \hat{u}_2^i, \hat{u}_3^i)$ . Since  $\hat{F}_4^i(\cdot)$  is again a univariate normal distribution we can easily determine its inverse and apply  $(\hat{F}_4^i)^{-1}(\cdot)$  to  $\hat{q}_u^i(\alpha)$  to obtain the conditional quantiles on the original scale  $\hat{q}_y^i(\alpha) = (\hat{F}_4^i)^{-1}(\hat{q}_u^i(\alpha))$ . The estimated median for individual 1 is 4.65, the estimated 90% interval is given by (4.42, 5.02). For individuals 2 and 3 the medians are 5.18 and 5.60 and the prediction intervals are (4.82, 5.58) and (5.13, 6.03), respectively. Thus all three prediction intervals contain the corresponding observed measurement values. Note that the 5% and 95% quantiles are in general not symmetric around the 50% quantile in this case.

In practice it might be interesting to investigate the influence of covariates on the estimated conditional quantiles. As an example we illustrate how the quantiles depend on the variable `bsa`. In Figure 5.4 we show the resulting estimates for the median (solid lines) and the prediction intervals (dashed lines) of individuals 1 (left), 2 (middle) and 3 (right) depending on the size of the heart valve for the linear mixed model (gray lines) and the D-vine based model (black lines). The estimated quantiles for the actual covariate

specifications of our three selected individuals are given by the intersections dotted vertical lines (indicating the true  $\text{bsa}$  values) and the quantile lines; for example the median is marked at the intersection point of the prediction interval and the corresponding median line. The actual observed measurement value  $y_4^i$  is also added as a circle.



**Figure 5.4.:** Estimated median (solid lines) and prediction intervals (dashed lines) of the fourth measurement for individuals 1 (left column), 2 (middle column) and 3 (right column) depending on the size of the heart valve for the LMM (gray) and the D-vine based model (black). Vertical lines indicate the true  $\text{bsa}$  values. Actual observed values are marked as circles.

First of all, we see that the quantiles estimated from the linear mixed model depend linearly on  $\text{bsa}$ . This is clear since the covariates only influence the estimated mean of the distribution of the fourth measurement given the first three. This influence is the same for all quantiles, i.e. the slope of the all gray lines is the same. Since the standard deviation does not depend on the covariate values all prediction intervals have the same width (even for different individuals). Normality implies that the prediction intervals are symmetric around the mean (median).

The quantiles estimated on the basis of the D-vine based model, however, inherit the flexibility of the D-vine model and do not depend linearly on the  $\text{bsa}$  value. The width of the prediction intervals varies among the three individuals and even for one individual depending on the covariate values. The slope for different quantiles can even be positive and negative for one individual. These phenomena can be seen very clearly in left plot in Figure 5.4 corresponding to the first individual.

This application illustrated how easily such investigations can be performed with both the linear mixed model and the D-vine based model. Analyzing the results, however, has made clear that the flexibility of the D-vine based model is a non-ignorable advantage over the linear mixed model. For both models it is eye-catching that the differences between the three individuals are considerable. This shows how important it can be to use the information available when making predictions for the future.

## 5.7. Conclusion and outlook

This chapter presented an intuitive and easily interpretable D-vine copula based model with arbitrary margins for unbalanced longitudinal data. The model was compared to linear mixed models and proved to be a generalization of this model class under the assumption that the correlation structure was homogeneous over the individuals. Further, we developed a BIC adjustment for our model. Being based on D-vine copulas our proposed model benefited from the possibility to model the underlying dependence structure very flexibly. Since we did not impose any restrictions on the univariate marginal distributions, this adds even more flexibility to the model. As joint estimation of the D-vine copula would become rather slow in high dimensions, we proposed a fast sequential alternative, where even missing data values could be handled without causing problems. Due to the nested nature of D-vine models our approach further easily allowed for predicting future events. In the application to the heart surgery data set the proposed model was able to fit the data considerably better than the linear mixed models. If data exhibited an even more complicated dependence structure than the considered data set (possibly including stronger tail dependence, asymmetries etc.), the Gaussian assumption of linear mixed models would certainly be so strongly violated that changing to a more flexible model would be inevitable.

In an ongoing research project the D-vine based modeling approach is extended to time-to-event data with right-censoring (Barthel et al., 2017).

# Appendix A

---

## Appendix to Chapter 3

---

### A.1. Proof of Proposition 3.6

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

$$\kappa_j(\mathbf{u}_{(j+1):d}) := \text{KL} \left( c_{j|(j+1):d}^f(\cdot | \mathbf{u}_{(j+1):d}), c_{j|(j+1):d}^g(\cdot | \mathbf{u}_{(j+1):d}) \right).$$

Then, by definition

$$\text{aKL}(\mathcal{R}^f, \mathcal{R}^g) = \sum_{j=1}^{d-1} \frac{1}{n^{d-j}} \sum_{\mathbf{u}_{(j+1):d} \in \mathcal{G}_j} \kappa_j(\mathbf{u}_{(j+1):d}) = \sum_{j=1}^{d-1} \frac{1}{n^{d-j}} \sum_{\mathbf{w}_{(j+1):d} \in \mathcal{W}_j} \kappa_j(T_{c_{(j+1):d}^f}(\mathbf{w}_{(j+1):d})).$$

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

$$\frac{1}{n^{d-j}} \sum_{\mathbf{w}_{(j+1):d} \in \mathcal{W}_j} \kappa_j(T_{c_{(j+1):d}^f}(\mathbf{w}_{(j+1):d})) \xrightarrow{n \rightarrow \infty} \int_{[\varepsilon, 1 - \varepsilon]^{d-j}} \kappa_j(T_{c_{(j+1):d}^f}(\mathbf{w}_{(j+1):d})) d\mathbf{w}_{(j+1):d}.$$

Substituting  $\mathbf{w}_{(j+1):d} = T_{c_{(j+1):d}^f}^{-1}(\mathbf{u}_{(j+1):d})$  yields

$$\begin{aligned} & \int_{[\varepsilon, 1 - \varepsilon]^{d-j}} \kappa_j(T_{c_{(j+1):d}^f}(\mathbf{w}_{(j+1):d})) d\mathbf{w}_{(j+1):d} \\ &= \int_{T_{c_{(j+1):d}^f}([\varepsilon, 1 - \varepsilon]^{d-j})} \kappa_j(\mathbf{u}_{(j+1):d}) c_{(j+1):d}^f(\mathbf{u}_{(j+1):d}) d\mathbf{u}_{(j+1):d} \end{aligned}$$

since

$$T_{c_{(j+1):d}^f}^{-1}(\mathbf{u}_{(j+1):d}) = (C_{j+1|(j+2):d}^f(u_{j+1} | \mathbf{u}_{(j+2):d}), \dots, C_{d-1|d}^f(u_{d-1} | \mathbf{u}_d), u_d)^\top$$

with (upper triangular) Jacobian matrix

$$J = J_{T_{c_{(j+1):d}^f}^{-1}(\mathbf{u}_{(j+1):d})} = \begin{pmatrix} c_{j+1|(j+2):d}^f(u_{j+1}|\mathbf{u}_{(j+2):d}) & & & & \\ & \ddots & & & \\ & & 0 & & * \\ & & & c_{d-1|d}^f(u_{d-1}|\mathbf{u}_d) & \\ & & & & 1 \end{pmatrix}$$

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

$$\lim_{\varepsilon \rightarrow 0} T_{c_{(j+1):d}^f}([\varepsilon, 1 - \varepsilon]^{d-j}) = T_{c_{(j+1):d}^f}([0, 1]^{d-j}) = [0, 1]^{d-j},$$

we obtain

$$\begin{aligned} \lim_{\varepsilon \rightarrow 0} \lim_{n \rightarrow \infty} \text{aKL}(\mathcal{R}^f, \mathcal{R}^g) &= \sum_{j=1}^{d-1} \int_{[0,1]^{d-j}} \kappa_j(\mathbf{u}_{(j+1):d}) c_{(j+1):d}^f(\mathbf{u}_{(j+1):d}) d\mathbf{u}_{(j+1):d} \\ &\stackrel{\text{Prop. 3.2}}{=} \text{KL}(c^f, c^g). \end{aligned}$$

## A.2. Regarding Remark 3.11

### A.2.1. Limit of the dKL

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

$$\kappa_j(\mathbf{u}_{(j+1):d}) := \text{KL}\left(c_{j|(j+1):d}^f(\cdot|\mathbf{u}_{(j+1):d}), c_{j|(j+1):d}^g(\cdot|\mathbf{u}_{(j+1):d})\right).$$

The contribution of  $\mathcal{D}_{j,k}^u$ ,  $j = 1, \dots, d-1$ ,  $k = 1, \dots, 2^{d-j-1}$ , to the dKL is given by

$$\frac{1}{n} \sum_{\mathbf{u}_{(j+1):d} \in \mathcal{D}_{j,k}^u} \kappa_j(\mathbf{u}_{(j+1):d}) = \frac{1}{n} \sum_{\mathbf{w}_{(j+1):d} \in \mathcal{D}_{j,k}^w} \kappa_j(T_{c_{(j+1):d}^f}(\mathbf{w}_{(j+1):d})) = \frac{1}{n} \sum_{i=1}^n \kappa_j(T_{c_{(j+1):d}^f}(\boldsymbol{\omega}(t_i))),$$

where  $\boldsymbol{\omega}(t) = \mathbf{r} + t\mathbf{v}(\mathbf{r})$  with  $\mathbf{v}(\cdot)$  as defined in Definition 3.9,  $\mathbf{r} \in \{0, 1\}^{d-j}$  being a corner point of  $D_{j,k}^w$  and  $t_i = \varepsilon + (i-1)\frac{1-2\varepsilon}{n-1}$  for  $i = 1, \dots, n$ . Letting  $n \rightarrow \infty$  yields

$$\frac{1}{n} \sum_{i=1}^n \kappa_j(T_{c_{(j+1):d}^f}(\boldsymbol{\omega}(t_i))) \xrightarrow{n \rightarrow \infty} \int_{t \in [\varepsilon, 1-\varepsilon]} \kappa_j(T_{c_{(j+1):d}^f}(\boldsymbol{\omega}(t))) dt. \quad (\text{A.1})$$

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

$$\begin{aligned} \int_{t \in [0,1]} \kappa_j(T_{c_{(j+1):d}^f}(\boldsymbol{\omega}(t))) dt &= \frac{1}{\sqrt{d-j}} \int_{t \in [0,1]} \kappa_j(T_{c_{(j+1):d}^f}(\boldsymbol{\omega}(t))) \|\dot{\boldsymbol{\omega}}(t)\| dt \\ &= \frac{1}{\sqrt{d-j}} \int_{\mathbf{w}_{(j+1):d} \in D_{j,k}^w} \kappa_j(T_{c_{(j+1):d}^f}(\mathbf{w}_{(j+1):d})) d\mathbf{w}_{(j+1):d} \\ &= \frac{1}{\sqrt{d-j}} \int_{\mathbf{u}_{(j+1):d} \in D_{j,k}^u} \kappa_j(\mathbf{u}_{(j+1):d}) c_{(j+1):d}^f(\mathbf{u}_{(j+1):d}) d\mathbf{u}_{(j+1):d}, \end{aligned}$$

where we substituted  $\mathbf{u}_{(j+1):d} := T_{c_{(j+1):d}^f}^{-1}(\mathbf{w}_{(j+1):d})$ ,  $d\mathbf{w}_{(j+1):d} = c_{(j+1):d}^f(\mathbf{u}_{(j+1):d}) d\mathbf{u}_{(j+1):d}$  (cf. Appendix A.1) in the last line.

## A.2.2. Tail transformation

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

$$\int_{s \in \Psi^{-1}([\varepsilon, 1-\varepsilon])} \kappa_j(T_{c_{(j+1):d}^f}(\boldsymbol{\omega}(\Psi(s)))) \Psi'(s) ds.$$

We use its discrete pendant

$$\frac{1}{n} \sum_{i=1}^n \kappa_j(T_{c_{(j+1):d}^f}(\boldsymbol{\omega}(\Psi(s_i)))) \Psi'(s_i),$$

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

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

with shape parameter  $a > 0$ , where  $\Phi$  is the standard normal distribution function. Figure A.1 shows the graph of  $\Psi_a$  for different values of  $a$ . We see that larger values of  $a$  imply more points being transformed into the tails. Having tested different values for  $a$ , we found that  $a = 4$  yields the best overall results. Therefore, we consistently use  $a = 4$ .

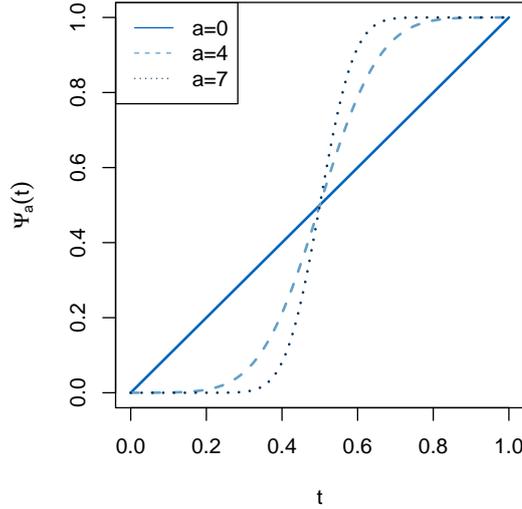


Figure A.1.: Plot of  $\Psi_a$  for  $a = 0, 4, 7$ .

## A.3. Finding the diagonal with the highest weight

### A.3.1. Procedure 1: Finding a starting value

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

1. Take each variable to be a node in an empty graph.
2. Consider the last row of the structure matrix, encoding the unconditional pair-copulas. Connect two nodes by an edge if the dependence of the corresponding variables is described by one of those copulas.
3. Assign a “+” to node 1.
4. As long as not all nodes have been assigned a sign, repeat for each node that was assigned a sign in the previous step:
  - a) Consider this ‘root’ node and its neighborhood, i.e. all other nodes that share an edge with the root node.
  - b) If the root node has a “+”, then assign to the neighbor node the sign of the Kendall’s  $\tau$  of the pair-copula connecting the root and neighbor node, else the opposite sign.

### A.3. Finding the diagonal with the highest weight

- The resulting direction vector  $\mathbf{v} = (v_1, \dots, v_d)^\top \in \{-1, 1\}^d$  has entries  $v_i$  which are 1 or  $-1$  if node  $i$  is has been assigned a “+” or a “-”, respectively.

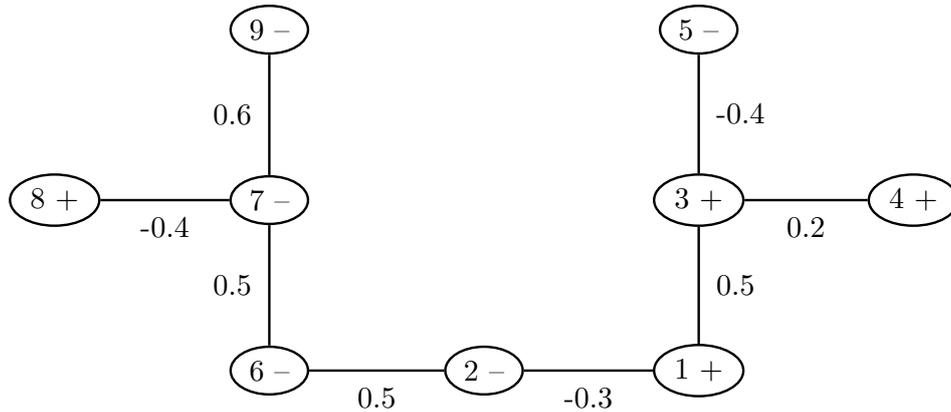
Note that if we had assigned a “-” to node 1 in Step 3, we would have ended up with  $-v$  instead of  $v$ , implying the same diagonal.

To illustrate the procedure from above we consider a nine-dimensional example: Let  $\mathcal{R}$  be a vine copula with density  $c$ , where the following (unconditional) pair-copulas are as specified in Table A.1.

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

**Table A.1.:** Specification of the pair-copulas with empty conditioning set.

Now, we take an empty graph with nodes 1 to 9 and add edges  $(i, j)$  if  $c_{i,j}$  is specified in Table A.1. The result is a tree on the nodes 1 to 9 (see Figure A.2). We assign a “+” to node 1 (implying  $v_1 = 1$ ) and consider its neighborhood  $\{2, 3\}$  as there are still nodes without a sign. Since  $\tau_{1,2} < 0$  and the root node 1 has been assigned a “+”, node 2 gets a “-” and we set  $v_2 = -1$ . Node 3 is assigned a “+” such that  $v_3 = 1$ . Next, we repeat this procedure for the neighborhoods of nodes 2 and 3. Iterating in this way until all nodes have been assigned a “+” or a “-” and all  $v_i$  have been set we obtain what is shown in Figure A.2. The resulting direction vector is given by  $\mathbf{v} = (v_1, \dots, v_9)^\top = (1, -1, 1, 1, -1, -1, -1, 1, -1)^\top$ .



**Figure A.2.:** Example for finding the candidate vector.

#### A.3.2. Procedure 2: Local search for better candidates

Having found a diagonal through Procedure 1 (Appendix A.3.1), we additionally perform the following steps in order to look if there is a diagonal with even higher weight in the “neighborhood” of  $\mathbf{v}$ .

Appendix A. Appendix to Chapter 3

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

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

In Step 2 the weight of numerous diagonals has to be calculated. For a fast determination of these weights it is reasonable to approximate the integral in Equation 3.14 by

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

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

# Appendix B

---

## Appendix to Chapter 4

---

### B.1. Proof of Theorem 4.1

In order to prove Theorem 4.1 we first prove an auxiliary lemma from which Theorem 4.1 follows as a corollary.

**Lemma B.1.** For  $k \in \{1, \dots, d\}$  and  $u_j \in [0, 1]$ ,  $j = 1, \dots, d$ , we have

$$\begin{aligned} \frac{\partial^k}{\partial u_1 \dots \partial u_k} \left[ C(u_1^{1/n}, \dots, u_d^{1/n})^n \right] &= \frac{1}{n^k} \left( \prod_{j=1}^k u_j \right)^{\frac{1}{n}-1} \\ &\times \sum_{j=1}^{k \wedge n} \left\{ \frac{n!}{(n-j)!} C(u_1^{1/n}, \dots, u_d^{1/n})^{n-j} \sum_{\mathcal{P} \in \mathcal{S}_{k,j}} \prod_{M \in \mathcal{P}} \partial_M C(u_1^{1/n}, \dots, u_d^{1/n}) \right\}. \end{aligned}$$

*Proof.* We will prove this statement using induction. For  $k = 1$  we have

$$\begin{aligned} \frac{\partial}{\partial u_1} \left[ C(u_1^{1/n}, \dots, u_d^{1/n})^n \right] &= n C(u_1^{1/n}, \dots, u_d^{1/n})^{n-1} \partial_1 C(u_1^{1/n}, \dots, u_d^{1/n}) \frac{1}{n} u_1^{\frac{1}{n}-1} = \\ &\frac{1}{n^1} \left( \prod_{j=1}^1 u_j \right)^{\frac{1}{n}-1} \sum_{j=1}^{1 \wedge n} \left\{ \frac{n!}{(n-j)!} C(u_1^{1/n}, \dots, u_d^{1/n})^{n-j} \sum_{\mathcal{P} \in \mathcal{S}_{1,j}} \prod_{M \in \mathcal{P}} \partial_M C(u_1^{1/n}, \dots, u_d^{1/n}) \right\}. \end{aligned}$$

The inductive step ( $k \rightarrow k+1$ ) proceeds as follows

$$\frac{\partial^{k+1}}{\partial u_1 \dots \partial u_{k+1}} \left[ C(u_1^{1/n}, \dots, u_d^{1/n})^n \right] = \frac{\partial}{\partial u_{k+1}} \left\{ \frac{\partial^k}{\partial u_1 \dots \partial u_k} \left[ C(u_1^{1/n}, \dots, u_d^{1/n})^n \right] \right\} =: (*)_1.$$

Applying the inductive assumption yields

Appendix B. Appendix to Chapter 4

$$\begin{aligned}
 (*)_1 &= \frac{\partial}{\partial u_{k+1}} \left\{ \frac{1}{n^k} \left( \prod_{j=1}^k u_j \right)^{\frac{1}{n}-1} \sum_{j=1}^{k \wedge n} \left\{ \frac{n!}{(n-j)!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \right. \right. \\
 &\quad \left. \left. \times \sum_{\mathcal{P} \in \mathcal{S}_{k,j}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \right\} \right\} =: (*)_2.
 \end{aligned}$$

We will consider the cases “ $n > k$ ” and “ $n \leq k$ ” separately. We begin with Case 1 ( $n > k$ ): We have  $k \wedge n = k$  and hence

$$\begin{aligned}
 (*)_2 &= \frac{1}{n^k} \left( \prod_{j=1}^k u_j \right)^{\frac{1}{n}-1} \sum_{j=1}^k \left\{ \frac{n!}{(n-j)!} \cdot \left\{ \frac{\partial}{\partial u_{k+1}} \left[ C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \right] \right. \right. \\
 &\quad \times \sum_{\mathcal{P} \in \mathcal{S}_{k,j}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) + C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \\
 &\quad \left. \left. \times \frac{\partial}{\partial u_{k+1}} \left[ \sum_{\mathcal{P} \in \mathcal{S}_{k,j}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \right] \right\} \right\} = (*)_3.
 \end{aligned}$$

Now we use that fact that for  $k \in \{1, \dots, d-1\}$  and  $j \in \{1, \dots, k \wedge n\}$  we have

$$\frac{\partial}{\partial u_{k+1}} \left[ \sum_{\mathcal{P} \in \mathcal{S}_{k,j}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \right] = \frac{u_{k+1}^{\frac{1}{n}-1}}{n} \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1,j} \\ \{k+1\} \notin \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right). \tag{B.1}$$

Applying Equation B.1 yields

$$\begin{aligned}
 (*)_3 &= \frac{1}{n^k} \left( \prod_{j=1}^k u_j \right)^{\frac{1}{n}-1} \left\{ \sum_{j=1}^k \frac{n!}{(n-j)!} \cdot (n-j) \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j-1} \right. \\
 &\quad \times \partial_{k+1} C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \frac{1}{n} u_{k+1}^{\frac{1}{n}-1} \cdot \sum_{\mathcal{P} \in \mathcal{S}_{k,j}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \\
 &\quad \left. + \sum_{j=1}^k \frac{n!}{(n-j)!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \cdot \frac{1}{n} u_{k+1}^{\frac{1}{n}-1} \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1,j} \\ \{k+1\} \notin \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \right\} \\
 &= \frac{1}{n^{k+1}} \left( \prod_{j=1}^{k+1} u_j \right)^{\frac{1}{n}-1} \left\{ \sum_{j=1}^k \frac{n!}{(n-(j+1))!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-(j+1)} \right. \\
 &\quad \times \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1,j+1} \\ \{k+1\} \in \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) +
 \end{aligned}$$

$$+ \left. \sum_{j=1}^k \frac{n!}{(n-j)!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \cdot \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1,j} \\ \{k+1\} \notin \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \right\} = (*)_4.$$

We perform an index shift in the first sum such that  $j+1$  is replaced by  $j$  and make use of the following two properties:

(A) For all  $\mathcal{P} \in \mathcal{S}_{l,1} = \{\{\{1, \dots, l\}\}\}$  holds that  $\{l\} \notin \mathcal{P}$ .

(B) For all  $\mathcal{P} \in \mathcal{S}_{l,l} = \{\{\{1\}, \dots, \{l\}\}\}$  holds that  $\{l\} \in \mathcal{P}$ .

This results in

$$\begin{aligned} (*)_4 &= \frac{1}{n^{k+1}} \left( \prod_{j=1}^{k+1} u_j \right)^{\frac{1}{n}-1} \left\{ \sum_{j=1}^{k+1} \frac{n!}{(n-j)!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \right. \\ &\quad \times \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1,j} \\ \{k+1\} \in \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \\ &\quad \left. + \sum_{j=1}^{k+1} \frac{n!}{(n-j)!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \cdot \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1,j} \\ \{k+1\} \notin \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \right\} \\ &= \frac{1}{n^{k+1}} \left( \prod_{j=1}^{k+1} u_j \right)^{\frac{1}{n}-1} \left\{ \sum_{j=1}^{(k+1) \wedge n} \frac{n!}{(n-j)!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \right. \\ &\quad \left. \times \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1,j} \\ M \in \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \right\}, \end{aligned}$$

where we used the fact that  $k+1 = (k+1) \wedge n$  since  $n > k$ . This concludes the first case. Case 2 ( $n \leq k$ ) is similar to the first one. The main difference is that for  $j = n$  we have  $C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} = 1$ , which was not possible before since  $j \leq k < n$ . Now  $k \wedge n = n$  and therefore we obtain

$$\begin{aligned} (*)_2 &= \frac{1}{n^k} \left( \prod_{j=1}^k u_j \right)^{\frac{1}{n}-1} \left\{ \sum_{j=1}^{n-1} \frac{n!}{(n-j)!} \cdot (n-j) \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j-1} \right. \\ &\quad \times \partial_{k+1} C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \frac{1}{n} u_{k+1}^{\frac{1}{n}-1} \cdot \sum_{\mathcal{P} \in \mathcal{S}_{k,j}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \\ &\quad \left. + \sum_{j=1}^n \frac{n!}{(n-j)!} C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \frac{\partial}{\partial u_{k+1}} \left[ \sum_{\mathcal{P} \in \mathcal{S}_{k,j}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \right] \right\} \\ &= \frac{1}{n^{k+1}} \left( \prod_{j=1}^{k+1} u_j \right)^{\frac{1}{n}-1} \left\{ \sum_{j=1}^{n-1} \frac{n!}{(n-(j+1))!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-(j+1)} \times \right. \end{aligned}$$

$$\begin{aligned}
& \times \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1, j+1} \\ \{k+1\} \in \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \\
& + \sum_{j=1}^n \frac{n!}{(n-j)!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \cdot \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1, j} \\ \{k+1\} \notin \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \Big\} \\
& = \frac{1}{n^{k+1}} \left( \prod_{j=1}^{k+1} u_j \right)^{\frac{1}{n}-1} \left\{ \sum_{j=1}^n \frac{n!}{(n-j)!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \right. \\
& \quad \times \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1, j} \\ \{k+1\} \in \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) + \sum_{j=1}^n \frac{n!}{(n-j)!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \\
& \quad \left. \times \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1, j} \\ \{k+1\} \notin \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \right\} \\
& = \frac{1}{n^{k+1}} \left( \prod_{j=1}^{k+1} u_j \right)^{\frac{1}{n}-1} \left\{ \sum_{j=1}^{(k+1) \wedge n} \frac{n!}{(n-j)!} \cdot C \left( u_1^{1/n}, \dots, u_d^{1/n} \right)^{n-j} \right. \\
& \quad \left. \times \sum_{\substack{\mathcal{P} \in \mathcal{S}_{k+1, j} \\ M \in \mathcal{P}}} \prod_{M \in \mathcal{P}} \partial_M C \left( u_1^{1/n}, \dots, u_d^{1/n} \right) \right\},
\end{aligned}$$

where we applied Equation B.1 in the second equality. Further, in the third equality we performed an index shift in the first sum and used Property (A) from above. Since  $n \leq k$  we have that  $n = (k+1) \wedge n$ . This concludes the second case and hence the proof of Lemma B.1.  $\square$

Having proved the auxiliary lemma we can now easily prove the statement from Theorem 4.1.

*Proof of Theorem 4.1.* Using Equation 4.3 we obtain

$$c_{\mathbf{M}^{(n)}}(u_1, \dots, u_d) = \frac{\partial^d}{\partial u_1 \dots \partial u_d} C_{\mathbf{M}^{(n)}}(u_1, \dots, u_d) = \frac{\partial^d}{\partial u_1 \dots \partial u_d} \left[ C(u_1^{1/n}, \dots, u_d^{1/n})^n \right].$$

As a final step, Theorem 4.1 follows directly as a corollary from Lemma B.1 by plugging in  $k = d$ .  $\square$

## B.2. Proof of Proposition 4.4

*Proof.* The expressions in Proposition 4.4 can be obtained by straight-forward calculations. We will start from the end: Expression 4 is simply the vine copula decomposition of  $c(u_1, u_2, u_3)$  from Equation 4.8. Using the vine copula decomposition or expression 4,

respectively, for expression 3c) we can write

$$\begin{aligned}
 \partial_{23}C(u_1, u_2, u_3) &= c_{2,3}(u_2, u_3) \int_0^{u_1} c_{1,2}(v_1, u_2) c_{1,3;2}(C_{1|2}(v_1|u_2), C_{3|2}(u_3|u_2); u_2) dv_1 \\
 &= c_{2,3}(u_2, u_3) \int_0^{u_1} \partial_{12}C_{1,2}(v_1, u_2) \partial_{13}C_{1,3;2}(C_{1|2}(v_1|u_2), C_{3|2}(u_3|u_2); u_2) dv_1 \\
 &= c_{2,3}(u_2, u_3) \int_0^{u_1} \left\{ \frac{\partial}{\partial v_1} \left[ \underbrace{\partial_2 C_{1,2}(v_1, u_2)}_{C_{1|2}(v_1|u_2)=w_1} \right] \right. \\
 &\quad \left. \times \partial_{13}C_{1,3;2}(C_{1|2}(v_1|u_2), C_{3|2}(u_3|u_2); u_2) \right\} dv_1 \\
 &= c_{2,3}(u_2, u_3) \int_0^{u_1} \frac{\partial w_1}{\partial v_1} \frac{\partial}{\partial w_1} \partial_3 C_{1,3;2}(w_1, C_{3|2}(u_3|u_2); u_2) \Big|_{w_1=C_{1|2}(v_1|u_2)} dv_1 \\
 &= c_{2,3}(u_2, u_3) C_{1|3;2}(C_{1|2}(u_1|u_2), C_{3|2}(u_3|u_2); u_2).
 \end{aligned}$$

Similar calculations yield expression 3a). Expression 2c) is obtained through the following calculations:

$$\begin{aligned}
 \partial_3 C(u_1, u_2, u_3) &= \int_0^{u_1} \int_0^{u_2} c(v_1, v_2, u_3) dv_1 dv_2 \\
 &= \int_0^{u_2} c_{2,3}(v_2, u_3) C_{1|2,3}(u_1|v_2, u_3) dv_2 \\
 &= \int_0^{u_2} c_{2,3}(v_2, u_3) C_{1|3;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) dv_2.
 \end{aligned}$$

Similarly, expression 2a) can be derived. In order to calculate  $C(u_1, u_2, u_3)$ , we derive another representation for  $\partial_3 C(u_1, u_2, u_3)$ :

$$\begin{aligned}
 \partial_3 C(u_1, u_2, u_3) &= \int_0^{u_1} \int_0^{u_2} c(v_1, v_2, u_3) dv_1 dv_2 \\
 &= \int_0^{u_2} \frac{\partial^2}{\partial v_2 \partial u_3} C_{2,3}(v_2, u_3) \frac{\partial}{\partial w_2} C_{1,3;2}(C_{1|2}(u_1|v_2), w_2; v_2) \Big|_{w_2=C_{3|2}(u_3|v_2)} dv_2 \\
 &= \int_0^{u_2} \frac{\partial w_2}{\partial u_3} \frac{\partial}{\partial w_2} C_{1,3;2}(C_{1|2}(u_1|v_2), w_2; v_2) \Big|_{w_2=C_{3|2}(u_3|v_2)} dv_2 \\
 &= \int_0^{u_2} \frac{\partial}{\partial u_3} C_{1,3;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) dv_2 \\
 &= \frac{\partial}{\partial u_3} \left[ \int_0^{u_2} C_{1,3;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) dv_2 \right].
 \end{aligned}$$

For the copula function  $C$  (expression 1) it follows

$$\begin{aligned} C(u_1, u_2, u_3) &= \int_0^{u_3} \frac{\partial}{\partial v_3} \left[ \int_0^{u_2} C_{1,3;2}(C_{1|2}(u_1|v_2), C_{3|2}(v_3|v_2); v_2) dv_2 \right] dv_3 \\ &= \int_0^{u_2} C_{1,3;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) dv_2. \end{aligned}$$

In order to obtain expression 2b) we can simply differentiate the above expression for the copula function  $C$  with respect to  $u_2$ . Finally, we differentiate expression 2c) with respect to  $u_1$  to end up with expression 3b):

$$\begin{aligned} \partial_{13}C(u_1, u_2, u_3) &= \frac{\partial}{\partial u_1} \left[ \int_0^{u_2} C_{1,3;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) c_{2,3}(v_2, u_3) dv_2 \right] \\ &= \int_0^{u_2} \partial_{13}C_{1,3;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) c_{1,2}(u_1, v_2) c_{2,3}(v_2, u_3) dv_2 \\ &= \int_0^{u_2} c_{1,3;2}(C_{1|2}(u_1|v_2), C_{3|2}(u_3|v_2); v_2) c_{1,2}(u_1, v_2) c_{2,3}(v_2, u_3) dv_2. \end{aligned}$$

□

# Appendix C

---

## *Appendix to Chapter 5*

---

### C.1. Proof of Proposition 5.1

We will prove the statement of Proposition 5.1 for  $d = 2$  in order to present the basic idea. The extension to higher dimensions works similarly but involves more tedious calculations. In our proof we adapt the derivation from Neath and Cavanaugh (2012). Since our proof is very similar up to the last step, we refer the reader to their paper for a more detailed argumentation.

The BIC is used for model selection when different parametric models  $M_1, \dots, M_K$  are available as candidates to describe a data set  $\mathbf{Y} = \{\mathbf{y}^1, \dots, \mathbf{y}^n\}$ . Further, let  $L(\boldsymbol{\theta}_k|\mathbf{Y})$  be the likelihood corresponding to model  $M_k$ , depending on the parameters  $\boldsymbol{\theta}_k \in \Omega_k$ , where  $\Omega_k \subseteq \mathbb{R}^{p_k}$  is the space of admissible parameter values. Let  $\pi(k)$  be the prior probability corresponding to model  $M_k$  and  $g(\boldsymbol{\theta}_k|k)$  denote a prior on  $\boldsymbol{\theta}_k$  given the model  $M_k$ . Using Bayes' Theorem we obtain the joint posterior of  $M_k$  and  $\boldsymbol{\theta}_k$ :

$$h(k, \boldsymbol{\theta}_k|\mathcal{Y}) = \frac{\pi(k)g(\boldsymbol{\theta}_k|k)L(\boldsymbol{\theta}_k|\mathcal{Y})}{m(\mathcal{Y})},$$

where  $m(\mathcal{Y})$  denotes the marginal distribution of  $\mathbf{Y}$ . We are interested in finding the highest posterior probability of  $M_k$  given  $\mathcal{Y}$ , which can be expressed as

$$P(k|\mathcal{Y}) = \frac{\pi(k)}{m(\mathcal{Y})} \int_{\Omega_k} L(\boldsymbol{\theta}_k|\mathcal{Y})g(\boldsymbol{\theta}_k|k)d\boldsymbol{\theta}_k.$$

Since maximizing  $P(k|\mathcal{Y})$  is equivalent to minimizing  $-2 \log P(k|\mathcal{Y})$  with respect to  $k$  and

$m(\mathcal{Y})$  does not depend on  $k$ , we will from now on consider

$$S(k|\mathcal{Y}) := -2 \log \pi(k) - 2 \log \int_{\Omega_k} L(\boldsymbol{\theta}_k|\mathcal{Y})g(\boldsymbol{\theta}_k|k)d\boldsymbol{\theta}_k. \quad (\text{C.1})$$

In order to be able to approximate the integrand from Equation C.1 we perform a second-order Taylor series expansion of the log-likelihood  $\log L(\boldsymbol{\theta}_k|\mathcal{Y})$  around the maximum-likelihood parameter estimate  $\hat{\boldsymbol{\theta}}_k = \arg \max_{\boldsymbol{\theta}_k \in \Omega_k} L(\boldsymbol{\theta}_k|\mathcal{Y})$ :

$$\begin{aligned} \log L(\boldsymbol{\theta}_k|\mathcal{Y}) \approx & \log L(\hat{\boldsymbol{\theta}}_k|\mathcal{Y}) + (\boldsymbol{\theta}_k - \hat{\boldsymbol{\theta}}_k)^\top \left. \frac{\partial \log L(\boldsymbol{\theta}_k|\mathcal{Y})}{\partial \boldsymbol{\theta}_k} \right|_{\boldsymbol{\theta}_k = \hat{\boldsymbol{\theta}}_k} \\ & + \frac{1}{2} (\boldsymbol{\theta}_k - \hat{\boldsymbol{\theta}}_k)^\top \left[ \left. \frac{\partial^2 \log L(\boldsymbol{\theta}_k|\mathcal{Y})}{\partial \boldsymbol{\theta}_k \partial \boldsymbol{\theta}_k^\top} \right|_{\boldsymbol{\theta}_k = \hat{\boldsymbol{\theta}}_k} \right] (\boldsymbol{\theta}_k - \hat{\boldsymbol{\theta}}_k). \end{aligned}$$

Since  $\hat{\boldsymbol{\theta}}_k$  maximizes  $L(\boldsymbol{\theta}_k|\mathcal{Y})$ , and hence also  $\log L(\boldsymbol{\theta}_k|\mathcal{Y})$ , we obtain

$$L(\boldsymbol{\theta}_k|\mathcal{Y}) \approx L(\hat{\boldsymbol{\theta}}_k|\mathcal{Y}) \exp \left\{ -\frac{1}{2} (\boldsymbol{\theta}_k - \hat{\boldsymbol{\theta}}_k)^\top H(\hat{\boldsymbol{\theta}}_k|\mathcal{Y})(\boldsymbol{\theta}_k - \hat{\boldsymbol{\theta}}_k) \right\},$$

where we denote the negative Hessian matrix of the log-likelihood by

$$H(\boldsymbol{\theta}_k|\mathcal{Y}) := -\frac{\partial^2 \log L(\boldsymbol{\theta}_k|\mathcal{Y})}{\partial \boldsymbol{\theta}_k \partial \boldsymbol{\theta}_k^\top}.$$

Neath and Cavanaugh (2012) and Cavanaugh and Neath (1999) argue that the above approximations hold for large samples  $\mathcal{Y}$  and further justify the use of a non-informative prior  $g(\boldsymbol{\theta}_k|k) = 1$  for any  $\boldsymbol{\theta}_k \in \Omega_k$ . Thus,

$$\int_{\Omega_k} L(\boldsymbol{\theta}_k|\mathcal{Y})d\boldsymbol{\theta}_k \approx L(\hat{\boldsymbol{\theta}}_k|\mathcal{Y})(2\pi)^{p_k/2} \left| H(\hat{\boldsymbol{\theta}}_k|\mathcal{Y}) \right|^{-1/2}. \quad (\text{C.2})$$

Plugging Equation C.2 into Equation C.1 yields

$$S(k|\mathcal{Y}) \approx -2 \log \pi(k) - 2 \log L(\hat{\boldsymbol{\theta}}_k|\mathcal{Y}) - p_k \log \pi + \log \left| H(\hat{\boldsymbol{\theta}}_k|\mathcal{Y}) \right|. \quad (\text{C.3})$$

In order to compute the determinant of  $H(\hat{\boldsymbol{\theta}}_k|\mathcal{Y})$  we consider the  $(\ell, m)$ th entry  $H_{\ell, m}$  of  $H(\boldsymbol{\theta}_k|\mathcal{Y})$ . Since  $d = 2$  the parameter vector  $\boldsymbol{\theta}_k = (\boldsymbol{\theta}_k^1, \boldsymbol{\theta}_k^2, \boldsymbol{\theta}_k^3)^\top$  can be split up such that  $\boldsymbol{\theta}_k^j \in \mathbb{R}^{q_j}$  parametrize the marginal distributions  $F_j$  of the  $j$ th measurement,  $j = 1, 2$  and  $\boldsymbol{\theta}_k^3 \in \mathbb{R}^{q_3}$  is the parameter vector of the copula  $c_{1,2}$  with  $p_k = q_1 + q_2 + q_3$ . For the sake of notation we assume that  $\mathcal{Y}$  is ordered such that  $\mathcal{Y}^2 = \{\mathbf{y}^1, \dots, \mathbf{y}^{n_2}\}$  and

$\mathcal{Y}^1 = \{\mathbf{y}^{n_2+1}, \dots, \mathbf{y}^n\}$  and further recall that  $N_1 = n_1 + n_2 = n$  and  $N_2 = n_2$ . We have

$$\begin{aligned}
 H_{\ell,m} &= -\frac{\partial^2}{\partial\theta_\ell\partial\theta_m} \sum_{i=1}^n \log L(\boldsymbol{\theta}_k|\mathbf{y}^i) \\
 &= -\sum_{i=1}^{N_1} \frac{\partial^2}{\partial\theta_\ell\partial\theta_m} \log f_1(y_1^i|\boldsymbol{\theta}_k^1) - \sum_{i=1}^{N_2} \frac{\partial^2}{\partial\theta_\ell\partial\theta_m} \log f_2(y_2^i|\boldsymbol{\theta}_k^2) \\
 &\quad - \sum_{i=1}^{N_2} \frac{\partial^2}{\partial\theta_\ell\partial\theta_m} \log c_{1,2}(F_1(y_1^i|\boldsymbol{\theta}_k^1), F_2(y_2^i|\boldsymbol{\theta}_k^2)|\boldsymbol{\theta}_k^3) \\
 &= N_1 \left[ -\frac{1}{N_1} \sum_{i=1}^{N_1} \frac{\partial^2}{\partial\theta_\ell\partial\theta_m} \log f_1(y_1^i|\boldsymbol{\theta}_k^1) \right] + N_2 \left[ -\frac{1}{N_2} \sum_{i=1}^{N_2} \frac{\partial^2}{\partial\theta_\ell\partial\theta_m} \log f_2(y_2^i|\boldsymbol{\theta}_k^2) \right] \\
 &\quad + N_2 \left[ -\frac{1}{N_2} \sum_{i=1}^{N_2} \frac{\partial^2}{\partial\theta_\ell\partial\theta_m} \log c_{1,2}(F_1(y_1^i|\boldsymbol{\theta}_k^1), F_2(y_2^i|\boldsymbol{\theta}_k^2)|\boldsymbol{\theta}_k^3) \right].
 \end{aligned}$$

Assuming that the data set is large, i.e.  $N_1$  and  $N_2$  are large, the expressions in the brackets (approximately) represent entries of the Fisher information matrices

$$\begin{aligned}
 I_1 &= I_1(\boldsymbol{\theta}_k^1|\mathcal{Y}) = -\mathbb{E} \left[ \frac{\partial^2}{\partial\boldsymbol{\theta}_k^1\partial(\boldsymbol{\theta}_k^1)^\top} \log f_1(Y_1|\boldsymbol{\theta}_k^1) \right] \in \mathbb{R}^{q_1 \times q_1}, \\
 I_2 &= I_2(\boldsymbol{\theta}_k^2|\mathcal{Y}^2) = -\mathbb{E} \left[ \frac{\partial^2}{\partial\boldsymbol{\theta}_k^2\partial(\boldsymbol{\theta}_k^2)^\top} \log f_2(Y_2|\boldsymbol{\theta}_k^2) \right] \in \mathbb{R}^{q_2 \times q_2}
 \end{aligned}$$

and

$$I_3 = \begin{pmatrix} I_3^{1,1} & I_3^{1,2} & I_3^{1,3} \\ I_3^{2,1} & I_3^{2,2} & I_3^{2,3} \\ I_3^{3,1} & I_3^{3,2} & I_3^{3,3} \end{pmatrix} = I_3((\boldsymbol{\theta}_k^1, \boldsymbol{\theta}_k^2, \boldsymbol{\theta}_k^3)|\mathcal{Y}^2) \in \mathbb{R}^{(q_1+q_2+q_3) \times (q_1+q_2+q_3)},$$

where

$$I_3^{\ell,m} = -\mathbb{E} \left[ \frac{\partial^2}{\partial\boldsymbol{\theta}_k^\ell\partial(\boldsymbol{\theta}_k^m)^\top} \log c_{1,2}(F_1(Y_1|\boldsymbol{\theta}_k^1), F_2(Y_2|\boldsymbol{\theta}_k^2)|\boldsymbol{\theta}_k^3) \right] \in \mathbb{R}^{q_\ell \times q_m}.$$

Thus,  $H(\hat{\boldsymbol{\theta}}_k|\mathcal{Y})$  can be written as

$$H(\hat{\boldsymbol{\theta}}_k|\mathcal{Y}) = \begin{pmatrix} N_1 I_1 + N_2 I_3^{1,1} & N_2 I_3^{1,2} & N_2 I_3^{1,3} \\ N_2 I_3^{2,1} & N_2 I_2 + N_2 I_3^{2,2} & N_2 I_3^{2,3} \\ N_2 I_3^{3,1} & N_2 I_3^{3,2} & N_2 I_3^{3,3} \end{pmatrix}.$$

Appendix C. Appendix to Chapter 5

Using the formula for the determinant of block-matrices (Silvester, 2000) we obtain

$$\begin{aligned}
\left| H(\hat{\boldsymbol{\theta}}_k | \mathcal{Y}) \right| &= N_1^{q_1} N_2^{q_2+q_3} \left| I_1 + \frac{N_2}{N_1} I_3^{1,2} - I_3^{1,3} (I_3^{3,3})^{-1} I_3^{3,1} + \frac{N_2}{N_1} [I_3^{1,2} - I_3^{1,3} (I_3^{3,3})^{-1} I_3^{3,2}] \right. \\
&\quad \times [I_2 + I_3^{2,2} - I_3^{2,3} (I_3^{3,3})^{-1} I_3^{3,2}]^{-1} [I_3^{2,1} - I_3^{2,3} (I_3^{3,3})^{-1} I_3^{3,1}] \left. \right| \\
&\quad \times \left| I_2 + I_3^{2,2} - I_3^{2,3} (I_3^{3,3})^{-1} I_3^{3,2} \right| \left| I_3^{3,3} \right| \\
&=: N_1^{q_1} N_2^{q_2+q_3} a(N_1, N_2).
\end{aligned}$$

Note that since  $N_2/N_1$  is bounded between 0 and 1,  $a(N_1, N_2)$  is also bounded. Plugging the expression for  $\left| H(\hat{\boldsymbol{\theta}}_k | \mathcal{Y}) \right|$  into Equation C.3 we obtain

$$S(k | \mathcal{Y}) \approx -2 \log \pi(k) - 2 \log L(\hat{\boldsymbol{\theta}}_k | \mathcal{Y}) - p_k \log \pi + q_1 \log N_1 + (q_2 + q_3) \log N_2 + \log a(N_1, N_2).$$

Discarding the terms that are bounded as the sample size goes to infinity yields

$$S(k | \mathcal{Y}) \approx -2 \log L(\hat{\boldsymbol{\theta}}_k | \mathcal{Y}) + \Delta p_1 \log N_1 + \Delta p_2 \log N_2$$

since  $\Delta p_1 = q_1$  and  $\Delta p_2 = q_2 + q_3$ . This proves the statement for  $d = 2$ . The proof of Proposition 5.1 in higher dimensions only differs from the above in that the calculations necessary to compute the determinant of  $H(\hat{\boldsymbol{\theta}}_k | \mathcal{Y})$  are much more involved since one has to compute the determinant of a  $(d(d+1)/2) \times (d(d+1)/2)$  block matrix.

---

## *Bibliography*

---

- Aas, K., Czado, C., Frigessi, A., and Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance, Mathematics and Economics*, 44:182–198.
- Acar, E. F., Genest, C., and Nešlehová, J. (2012). Beyond simplified pair-copula constructions. *Journal of Multivariate Analysis*, 110:74–90.
- Airy, G. (1861). *On the algebraic and numerical theory of errors of observations and the combination of observations*. London: Macmillan.
- Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. New York, NY: Springer.
- Barthel, N., Geerdens, C., Killiches, M., Janssen, P., and Czado, C. (2016). Vine copula based inference of multivariate event time data. *arXiv preprint arXiv:1603.01476*.
- Barthel, N., Geerdens, C., Killiches, M., Janssen, P., and Czado, C. (2017). Vine models for recurrent event times subject to right censoring. *Unpublished working paper*.
- Bedford, T. and Cooke, R. M. (2002). Vines: A new graphical model for dependent random variables. *Annals of Statistics*, 30(4):1031–1068.
- Brechmann, E. C. and Czado, C. (2013). Risk management with high-dimensional vine copulas: An analysis of the Euro Stoxx 50. *Statistics & Risk Modeling*, 30(4):307–342.
- Brechmann, E. C., Czado, C., and Aas, K. (2012). Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics*, 40(1):68–85.
- Brechmann, E. C. and Joe, H. (2015). Truncation of vine copulas using fit indices. *Journal of Multivariate Analysis*, 138:19–33.
- Bücher, A. and Segers, J. (2014). Extreme value copula estimation based on block maxima of a multivariate stationary time series. *Extremes*, 17(3):495–528.

## Bibliography

- Caffisch, R. E. (1998). Monte Carlo and quasi-Monte Carlo methods. *Acta numerica*, 7:1–49.
- Casella, G. and Berger, R. L. (2002). *Statistical inference*, volume 2. Pacific Grove, CA: Duxbury.
- Cavanaugh, J. E. and Neath, A. A. (1999). Generalizing the derivation of the Schwarz information criterion. *Communications in Statistics-Theory and Methods*, 28(1):49–66.
- Chen, X. and Fan, Y. (2005). Pseudo-likelihood ratio tests for semiparametric multivariate copula model selection. *Canadian Journal of Statistics*, 33(3):389–414.
- Chen, X. and Fan, Y. (2006). Estimation and model selection of semiparametric copula-based multivariate dynamic models under copula misspecification. *Journal of Econometrics*, 135(1):125–154.
- Cooke, R. M., Joe, H., and Chang, B. (2015). Vine regression. *Resources for the Future Discussion Paper*, pages 15–52.
- Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. Hoboken, NJ: John Wiley & Sons.
- Czado, C. (2010). Pair-copula constructions of multivariate copulas. In Jaworski, P., Durante, F., Härdle, W. K., and Rychlik, T., editors, *Copula Theory and Its Applications: Proceedings of the Workshop Held in Warsaw, 25-26 September 2009*, pages 93–109. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Czado, C. and Min, A. (2011). Bayesian inference for D-vines: Estimation and model selection. *Dependence Modeling: Vine Copula Handbook*, World Scientific Publishing Co., Singapore, pages 249–264.
- Davis, C. S. (2002). *Statistical methods for the analysis of repeated measurements*. Springer Science & Business Media.
- Delattre, M., Lavielle, M., and Poursat, M.-A. (2014). A note on BIC in mixed-effects models. *Electronic Journal of Statistics*, 8(1):456–475.
- Diggle, P. (2002). *Analysis of longitudinal data*. Oxford University Press.
- Diggle, P. J. and Donnelly, J. B. (1989). A selected bibliography on the analysis of repeated measurements and related areas. *Australian Journal of Statistics*, 31(1):183–193.
- Diks, C., Panchenko, V., and Van Dijk, D. (2010). Out-of-sample comparison of copula specifications in multivariate density forecasts. *Journal of Economic Dynamics and Control*, 34(9):1596–1609.

- Dißmann, J., Brechmann, E. C., Czado, C., and Kurowicka, D. (2013). Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59:52–69.
- Do, M. N. (2003). Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models. *Signal Processing Letters, IEEE*, 10(4):115–118.
- Dombry, C. (2015). Existence and consistency of the maximum likelihood estimators for the extreme value index within the block maxima framework. *Bernoulli*, 21(1):420–436.
- Efron, B. and Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. CRC press.
- Embrechts, P., McNeil, A., and Straumann, D. (2002). Correlation and dependence in risk management: properties and pitfalls. *Risk management: value at risk and beyond*, pages 176–223.
- Embrechts, P., Mcneil, E., and Straumann, D. (1999). Correlation: pitfalls and alternatives. In *Risk Magazine*. Citeseer.
- Erhardt, T. M., Czado, C., and Schepsmeier, U. (2015). R-vine models for spatial time series with an application to daily mean temperature. *Biometrics*, 71(2):323–332.
- Fahrmeir, L., Kneib, T., Lang, S., and Marx, B. (2013). *Regression: models, methods and applications*. Springer Science & Business Media.
- Faranda, D., Lucarini, V., Turchetti, G., and Vaianti, S. (2011). Numerical convergence of the block-maxima approach to the Generalized Extreme Value distribution. *Journal of Statistical Physics*, 145(5):1156–1180.
- Ferreira, A. and de Haan, L. (2014). On the block maxima method in extreme value theory: PWM estimators. *The Annals of Statistics*, 43(1):276–298.
- Fitzmaurice, G., Davidian, M., Verbeke, G., and Molenberghs, G. (2008). *Longitudinal data analysis*. CRC Press.
- Genest, C. and Favre, A.-C. (2007). Everything you always wanted to know about copula modeling but were afraid to ask. *Journal of Hydrologic Engineering*, 12(4):347–368.
- Genest, C. and Nešlehová, J. (2012). *Copula Modeling for Extremes*, pages 530–541. John Wiley & Sons, Ltd.
- Genest, C. and Rémillard, B. (2008). Validity of the parametric bootstrap for goodness-of-fit testing in semiparametric models. *Annales de l’IHP Probabilités et Statistiques*, 44(6):1096–1127.

## Bibliography

- Gräler, B. (2014). Modelling skewed spatial random fields through the spatial vine copula. *Spatial Statistics*, 10:87–102.
- Gruber, L. and Czado, C. (2015). Sequential bayesian model selection of regular vine copulas. *Bayesian Analysis*, 10(4):937–963.
- Gudendorf, G. and Segers, J. (2010). Extreme-value copulas. In *Copula theory and its applications*, pages 127–145. Springer.
- Hedeker, D. and Gibbons, R. D. (2006). *Longitudinal data analysis*, volume 451. John Wiley & Sons.
- Hershey, J. R. and Olsen, P. A. (2007). Approximating the Kullback Leibler divergence between Gaussian mixture models. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–317. IEEE.
- Hobæk Haff, I., Aas, K., and Frigessi, A. (2010). On the simplified pair-copula construction — Simply useful or too simplistic? *Journal of Multivariate Analysis*, 101(5):1296–1310.
- Hofert, M., Mächler, M., and McNeil, A. J. (2012). Archimedean copulas in high dimensions: Estimators and numerical challenges motivated by financial applications. *Journal de la Société Française de Statistique*, 154(1):25–63.
- Hollander, M., Wolfe, D. A., and Chicken, E. (2014). *Nonparametric statistical methods*. John Wiley & Sons.
- Hüsler, J. and Reiss, R.-D. (1989). Maxima of normal random vectors: between independence and complete dependence. *Statistics & Probability Letters*, 7(4):283–286.
- Ibrahim, J. G. and Molenberghs, G. (2009). Missing data methods in longitudinal studies: a review. *Test*, 18(1):1–43.
- Jarušková, D. and Hanek, M. (2006). Peaks over threshold method in comparison with block-maxima method for estimating high return levels of several northern moravia precipitation and discharges series. *Journal of Hydrology and Hydromechanics*, 54(4):309–319.
- Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 186, pages 453–461. The Royal Society.
- Joe, H. (1996). Families of  $m$ -variate distributions with given margins and  $m(m - 1)/2$  bivariate dependence parameters. *Lecture Notes-Monograph Series*, 28:120–141.

- Joe, H. (1997). *Multivariate models and multivariate dependence concepts*. CRC Press.
- Joe, H. (2006). Generating random correlation matrices based on partial correlations. *Journal of Multivariate Analysis*, 97(10):2177–2189.
- Joe, H. (2014). *Dependence modeling with copulas*. CRC Press.
- Jones, R. H. (2011). Bayesian information criterion for longitudinal and clustered data. *Statistics in Medicine*, 30(25):3050–3056.
- Killiches, M. and Czado, C. (2015). Block-Maxima of Vines. In Dey, D. and Yan, J., editors, *Extreme Value Modelling and Risk Analysis: Methods and Applications*, pages 109–130. Boca Raton, FL: Chapman & Hall/CRC Press.
- Killiches, M. and Czado, C. (2017). A D-vine copula based model for repeated measurements extending linear mixed models with homogeneous correlation structure. *arXiv preprint arXiv:1705.06261*.
- Killiches, M., Kraus, D., and Czado, C. (2017a). Examination and visualisation of the simplifying assumption for vine copulas in three dimensions. *Australian & New Zealand Journal of Statistics*, 59(1):95–117.
- Killiches, M., Kraus, D., and Czado, C. (2017b). Model distances for vine copulas in high dimensions. *Statistics and Computing*, doi:10.1007/s11222-017-9733-y.
- Killiches, M., Kraus, D., and Czado, C. (2017c). Using model distances to investigate the simplifying assumption, model selection and truncation levels for vine copulas. *arXiv preprint arXiv:1610.08795v3*.
- Kraus, D. and Czado, C. (2017a). D-vine copula based quantile regression. *Computational Statistics & Data Analysis*, 110C:1–18.
- Kraus, D. and Czado, C. (2017b). Growing simplified vine copula trees: improving Dißmann’s algorithm. *arXiv preprint arXiv:1703.05203*.
- Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.
- Lambert, P. and Vandenhende, F. (2002). A copula-based model for multivariate non-normal longitudinal data: analysis of a dose titration safety study on a new antidepressant. *Statistics in Medicine*, 21(21):3197–3217.

## Bibliography

- Lim, E., Ali, A., Theodorou, P., Sousa, I., Ashrafiyan, H., Chamageorgakis, T., Duncan, A., Henein, M., Diggle, P., and Pepper, J. (2008). Longitudinal study of the profile and predictors of left ventricular mass regression after stentless aortic valve replacement. *The Annals of Thoracic Surgery*, 85(6):2026–2029.
- Lindsey, J. and Lindsey, P. (2006). Multivariate distributions with correlation matrices for nonlinear repeated measurements. *Computational Statistics & Data Analysis*, 50(3):720–732.
- Lindsey, J. K. (1999). *Models for repeated measurements*. Oxford University Press.
- Lorch, R. F. and Myers, J. L. (1990). Regression analyses of repeated measures data in cognitive research. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16(1):149.
- Ludbrook, J. (1994). Repeated measurements and multiple comparisons in cardiovascular research. *Cardiovascular Research*, 28(3):303–311.
- Malin, A. and Linnakylä, P. (2001). Multilevel modelling in repeated measures of the quality of finnish school life. *Scandinavian Journal of Educational Research*, 45(2):145–166.
- Marty, C. and Blanchet, J. (2012). Long-term changes in annual maximum snow depth and snowfall in Switzerland based on extreme value statistics. *Climatic Change*, 111(3-4):705–721.
- Maya, L., Albeiro, R., Gomez-Gonzalez, J. E., and Melo Velandia, L. F. (2015). Latin American exchange rate dependencies: A regular vine copula approach. *Contemporary Economic Policy*, 33(3):535–549.
- McCulloch, C., Searle, S., and Neuhaus, J. (2011). *Generalized, Linear, and Mixed Models*. Wiley Series in Probability and Statistics. Wiley.
- McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245.
- McNeil, A. J., Frey, R., and Embrechts, P. (2010). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton University Press.
- Meester, S. G. and MacKay, J. (1994). A parametric model for cluster correlated categorical data. *Biometrics*, 50(4):954–963.

- Min, A. and Czado, C. (2010). Bayesian inference for multivariate copulas using pair-copula constructions. *Journal of Financial Econometrics*, 8(4):511–546.
- Morales-Nápoles, O. (2011). Counting vines. In D. Kurowicka and H. Joe (Eds.), *Dependence Modeling: Vine Copula Handbook*. World Scientific Publishing Co.
- Müller, D. and Czado, C. (2016). Representing sparse Gaussian DAGs as sparse R-vines allowing for non-Gaussian dependence. *arXiv preprint arXiv:1604.04202*.
- Müller, S., Scealy, J. L., and Welsh, A. H. (2013). Model selection in linear mixed models. *Statistical Science*, 28(2):135–167.
- Nagler, T. (2017). *kdevine: Multivariate Kernel Density Estimation with Vine Copulas*. R package version 0.3.0.
- Nagler, T. and Czado, C. (2016). Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas. *Journal of Multivariate Analysis*, 151:69–89.
- Nai Ruscone, M. and Osmetti, S. A. (2017). Modelling the dependence in multivariate longitudinal data by pair copula decomposition. In Ferraro, M. B., Giordani, P., Vantaggi, B., Gagolewski, M., Ángeles Gil, M., Grzegorzewski, P., and Hryniewicz, O., editors, *Soft Methods for Data Science*, pages 373–380. Springer International Publishing, Cham.
- Naveau, P., Guillou, A., Cooley, D., and Diebolt, J. (2009). Modelling pairwise dependence of maxima in space. *Biometrika*, 96(1):1–17.
- Neath, A. A. and Cavanaugh, J. E. (2012). The Bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203.
- Nelsen, R. (2006). An Introduction to Copulas, 2nd. *New York: Springer Science Business Media*.
- Newbold, E. M. (1927). Practical applications of the statistics of repeated events’ particularly to industrial accidents. *Journal of the Royal Statistical Society*, 90(3):487–547.
- Nikulin, G., Kjellström, E., Hansson, U., Strandberg, G., and Ullerstig, A. (2011). Evaluation and future projections of temperature, precipitation and wind extremes over europe in an ensemble of regional climate simulations. *Tellus A*, 63(1):41–55.
- Panagiotelis, A., Czado, C., and Joe, H. (2012). Pair copula constructions for multivariate discrete data. *Journal of the American Statistical Association*, 107(499):1063–1072.

## Bibliography

- Panagiotelis, A., Czado, C., Joe, H., and Stöber, J. (2017). Model selection for discrete regular vine copulas. *Computational Statistics & Data Analysis*, 106:138–152.
- Philipson, P., Sousa, I., Diggle, P. J., Williamson, P., Kolamunnage-Dona, R., Henderson, R., and Hickey, G. L. (2017). *joiner: Joint Modelling of Repeated Measurements and Time-to-Event Data*. R package version 1.1.0.
- Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., and R Core Team (2017). *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-131.
- Potvin, C., Lechowicz, M. J., and Tardif, S. (1990). The statistical analysis of ecophysiological response curves obtained from experiments involving repeated measures. *Ecology*, 71(4):1389–1400.
- Rocco, M. (2014). Extreme value theory in finance: A survey. *Journal of Economic Surveys*, 28(1):82–108.
- Rosenblatt, M. (1952). Remarks on a Multivariate Transformation. *Ann. Math. Statist.*, 23(3):470–472.
- Salmon, F. (2009). Recipe for Disaster: The Formula That Killed Wall Street. *Wired*, 17(3):74–79.
- Schepsmeier, U. (2015). Efficient information based goodness-of-fit tests for vine copula models with fixed margins. *Journal of Multivariate Analysis*, 138:34–52.
- Schepsmeier, U., Stöber, J., Brechmann, E. C., Gräler, B., Nagler, T., and Erhardt, T. (2017). *VineCopula: Statistical Inference of Vine Copulas*. R package version 2.1.1.
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.
- Shen, C. and Weissfeld, L. (2006). A copula model for repeated measurements with non-ignorable non-monotone missing outcome. *Statistics in Medicine*, 25(14):2427–2440.
- Shi, P., Feng, X., and Boucher, J.-P. (2016). Multilevel modeling of insurance claims using copulas. *The Annals of Applied Statistics*, 10(2):834–863.
- Shi, P. and Yang, L. (2016). Pair copula constructions for insurance experience rating. *Working paper*.
- Sibuya, M. (1960). Bivariate extreme statistics, I. *Annals of the Institute of Statistical Mathematics*, 11(2):195–210.

- Silvester, J. R. (2000). Determinants of block matrices. *The Mathematical Gazette*, 84(501):460–467.
- Sklar, A. (1959). Fonctions de Répartition à  $n$  Dimensions et leurs Marges. *Publ. Inst. Stat. Univ. Paris*, 8:229–231.
- Smith, M., Min, A., Almeida, C., and Czado, C. (2010). Modeling longitudinal data using a pair-copula decomposition of serial dependence. *Journal of the American Statistical Association*, 105(492):1467–1479.
- Smith, M. S. (2015). Copula modelling of dependence in multivariate time series. *International Journal of Forecasting*, 31(3):815–833.
- Spanhel, F. and Kurz, M. S. (2015). Simplified vine copula models: Approximations based on the simplifying assumption. *arXiv preprint arXiv:1510.06971*.
- Stöber, J. and Czado, C. (2012). Pair Copula Constructions. In J.-F. Mai and M. Scherer (2012), *Simulating Copulas: Stochastic Models, Sampling Algorithms, and Applications*. World Scientific.
- Stöber, J., Joe, H., and Czado, C. (2013). Simplified pair copula constructions—limitations and extensions. *Journal of Multivariate Analysis*, 119:101–118.
- Sun, J., Frees, E. W., and Rosenberg, M. A. (2008). Heavy-tailed longitudinal data modeling using copulas. *Insurance: Mathematics and Economics*, 42(2):817–830.
- Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78.
- Vatter, T. and Nagler, T. (2016). Generalized additive models for pair-copula constructions. *Available at SSRN 2817949*.
- Verbeke, G. and Molenberghs, G. (2009). *Linear mixed models for longitudinal data*. Springer Science & Business Media.
- Yeung, K. Y., Medvedovic, M., and Bumgarner, R. E. (2003). Clustering gene-expression data with repeated measurements. *Genome Biology*, 4(5):R34.