Technische Universität München
Fakultät für Mathematik
Lehrstuhl für Mathematische Statistik

D-vine copula based quantile regression and the simplifying assumption for vine copulas

Daniel Victor Kraus

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.

Vorsitzender:		Prof. Dr. Matthias Scherer
Prüfende/-r der Dissertation:	1.	Prof. Claudia Czado, Ph.D.
	2.	Prof. Dr. Matthias Fischer, Friedrich-Alexander-Universität Erlangen-Nürnberg
	3.	Prof. Dr. Roger Cooke, Technische Universität Delft, Niederlande (schriftliche Beurteilung)

Die Dissertation wurde am 29.05.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 04.10.2017 angenommen.

Abstract

Over the last decade, vine copulas have evolved to represent one of the standard tools for dependence modeling in the statistical community. As so-called pair-copula constructions they are flexible models decomposing multivariate copulas into bivariate building blocks. Each of these can be modeled separately by a bivariate parametric or nonparametric copula, resulting in a huge class of models.

In this thesis we consider several aspects of vine copulas. First, we use a subclass of vine copulas called D-vine copulas for quantile regression, which is the prediction of the quantiles of a response variable conditioned on several covariates assuming certain values. We develop an algorithm that sequentially constructs the D-vine, adding one covariate after another ordered decreasingly with regards to explanatory power, as long as the model's fit is significantly improved. Thus, an automatic covariate selection and ranking is facilitated. The resulting D-vine model admits an analytic extraction of the conditional quantiles guaranteeing fast and precise calculations. In contrast to traditional linear quantile regression, no distributional assumptions are made and quantiles for different quantile levels cannot cross each other. With the application of stress testing, a task with increasing relevance in the financial world, we show how D-vine copula based quantile regression can be used in practice. Further, we describe how D-vine quantile regression can be generalized to account for mixed discrete and continuous data sets and present another application explaining and predicting bike rental counts for a bike sharing system in Washington, D.C.

The second big topic of this thesis is the simplifying assumption that is usually made for vine copulas in order to make inference tractable, especially in higher dimensions. It assumes that the conditional copulas of a vine copula decomposition do not vary with the values of the conditioning vector. We investigate the implications of the simplifying assumption for three-dimensional vine copulas by plotting and comparing contour surfaces of vine copula densities in simplified and non-simplified scenarios. We find that non-simplified vine copulas exhibit much more irregular shapes than simplified vines. By a comparison of fitted simplified and non-simplified vine copula densities with nonparametric density fits, we describe a visual test for the choice between a specified more complicated non-simplified and the more parsimonious simplified models.

Further, we develop a statistical procedure testing for the simplifying assumption for a given data set. After fitting both simplified and non-simplified vine copulas, we test whether the difference between both models is significantly different from zero. As a distance measure we use a modified version of the Kullback-Leibler distance, which is specifically designed to be fast and still accurate, even in higher dimensions. We show that the test has a high power and demonstrate its usefulness in two real data applications.

Finally, we propose two new algorithms that sequentially estimate the tree structure of a vine copula model with the focus on producing models for which the simplifying assumption is violated as little as possible. By using a recently developed test for constant conditional correlations we use information on how the chosen tree structure affects the validity of the simplifying assumption. In a simulation study as well as several real data examples we show that our algorithms are able to outperform the benchmark structure selection method given by Dißmann's algorithm in many cases.

Zusammenfassung

Im Laufe des letzten Jahrzehnts sind Vine Copulas zu einem der Standardwerkzeuge für Abhängigkeitsmodellierung in der Statistik geworden. Als sogenannte Paar-Copula Konstruktionen sind sie flexible Modelle, die multivariate Copulas in bivariate Bausteine zerlegen. Diese können unabhängig voneinander durch bivariate parametrische oder nichtparametrische Copulas modelliert werden, was zu einer riesigen Klasse von Modellen führt. In dieser Arbeit betrachten wir verschiedene Aspekte von Vine Copulas. Zuerst verwenden wir eine Unterklasse von Vine Copulas, die D-Vine Copulas, für Quantilsregression, also die Vorhersage der Quantile einer Variable, die auf mehrere Kovariablen bedingt ist. Wir entwickeln einen Algorithmus, der den D-Vine sequentiell konstruiert, indem eine Kovariable nach der anderen, geordnet im Bezug auf ihre Erklärungskraft, hinzugefügt wird, solange dadurch die Anpassung des Modells signifikant verbessert wird. So wird eine automatische Kovariablen-Auswahl und Ordnung nach Wichtigkeit ermöglicht. Das resultierende D-Vine Modell lässt eine analytische Bestimmung der bedingten Quantile zu, was schnelle und genaue Berechnungen zur Folge hat. Im Gegensatz zur herkömmlichen linearen Quantilsregression werden keine Annahmen über die zugrunde liegenden Verteilungen getroffen und Quantile für unterschiedliche Quantilniveaus können sich nicht schneiden. Mit der Anwendung von Stress-Tests, einer Aufgabe mit zunehmender Relevanz in der Finanzwelt, zeigen wir, wie D-Vine Copula basierte Quantilsregression in der Praxis genutzt werden kann. Weiterhin beschreiben wir, wie die D-Vine Quantilsregression verallgemeinert werden kann, um gemischte diskrete und kontinuierliche Datensätze zu berücksichtigen und präsentieren eine weitere Anwendung zur Vorhersage der Anzahl von geliehenen Fahrrädern in Washington, D.C.

Das zweite große Thema dieser Arbeit ist die simplifying assumption, die gewöhnlich für Vine Copulas gemacht wird, um Inferenz, insbesondere in höheren Dimensionen, möglich zu machen. Hierbei nimmt man an, dass die bedingten Copulas einer Vine Copula Zerlegung nicht mit den Werten des Bedingungsvektors variieren. Wir untersuchen die Implikationen der simplifying assumption für dreidimensionale Vine Copulas, indem wir die Konturoberflächen der Vine Copula Dichten in Szenarien mit und ohne simplifying assumption plotten und vergleichen. Wir sehen, dass nonsimplified Vine Copulas eine viel unregelmäßigere Form aufweisen als simplified Vine Copulas. Durch einen Vergleich mit nichtparametrisch geschätzten Dichten beschreiben wir einen visuellen Test für die Wahl zwischen einem komplizierteren nonsimplified und dem einfacheren simplified Modell. Außerdem entwickeln wir einen statistischen Test für die simplifying assumption. Nach der Anpassung von simplified und nonsimplified Vine Copulas wird getestet, ob die Distanz zwischen beiden Modellen signifikant von Null verschieden ist. Als Distanzmaß verwenden wir eine modifizierte Version des Kullback-Leibler-Abstandes, die dafür entwickelt wurde, auch in hohen Dimensionen schnell und noch genau zu sein. Wir zeigen, dass der Test eine hohe Güte hat und demonstrieren seine Nützlichkeit in zwei Datenanwendungen.

Schließlich stellen wir zwei neue Algorithmen vor, die die Baumstruktur eines Vine Copula Modells sequentiell schätzen, mit dem Fokus auf Modellen, für die die simplifying assumption so wenig wie möglich verletzt wird. Durch die Verwendung eines kürzlich entwickelten Tests auf konstante bedingte Korrelationen verwenden wir Informationen darüber, wie die gewählte Baumstruktur die Gültigkeit dieser Annahme beeinflusst. In einer Simulationsstudie sowie in mehreren realen Datenbeispielen zeigen wir, dass unsere Algorithmen in der Lage sind, in vielen Fällen bessere Ergebnisse zu liefern als die gängige Strukturauswahlmethode von Dißmann.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to Prof. *Claudia Czado* who supervised and guided me during my doctoral candidacy. You always had an open door for me and could apply your great expertise about statistics and vine copulas to help me even with the toughest problems. It was a great pleasure working with and learning from you.

I would not be where I am today if it was not for *Matthias Killiches*. For over two decades now, our joint paths led us from the first day at school to a successful Master's degree in mathematics, culminating in door-to-door offices during our PhD with many collaborations and two published journal papers. Thank you for these fantastic years and I hope that our paths will not separate for many years to come.

Thanks must also go to all of my other fellow PhD students for the fun we had in the office as well as at the conferences we went to. To only mention a few, I want to thank *Nicole Barthel* for her always positive attitude and her incredible birthday cakes, *Thomas Nagler* for his constant interest in and valuable feedback on my research projects, *Dominik Müller* for his helpfulness and great sense of humor and *Sven Buhl* simply for being Sven.

Further appreciation is directed towards the anonymous referees and the editors of my publications and also to the referees of this thesis for their time and effort.

Last but not least, I would like to thank my friends and family for the sheer endless support and encouragement they gave me throughout the past years: My parents *Victor* and *Ursula Kraus* have always been there for me, be it financially, emotionally or simply by supporting my decisions. My brother *Felix Kraus* and his partner *Bianca Kennedy* were great roommates at the beginning of my PhD and always surprised me with their infinite amount of creativity. My aunt *Ursula Bauckholt* helped me whenever she could and supported me in the most selfless ways. Thanks also to all the friends who distracted me from the sometimes exhausting daily routine of a PhD candidate, be it with climbing, badminton, tennis, playing Schafkopf or just by having a friendly chat. My final and most important thanks go to you, *Katharina Liebel*. Our love has been growing since the day we met just a few months before the start of my PhD and I will always be grateful for the support, love and kindness you offered me even in the more difficult times of my PhD. I dedicate this thesis to you.

List of contributed articles

This thesis is a publication-based dissertation, based on the following articles:

I) Kraus, D. and Czado, C. (2017)
D-vine copula based quantile regression
Computational Statistics & Data Analysis, 110, 1–18

II) Killiches, M., Kraus, D. and Czado, C. (2017)
Examination and visualisation of the simplifying assumption for vine copulas in three dimensions
Australian & New Zealand Journal of Statistics, 59, 95–117

III) Kraus, D. and Czado, C. (2017)
Growing simplified vine copula trees: improving Dißmann's algorithm
in revision at Computational Statistics & Data Analysis

IV) Killiches, M., Kraus, D. and Czado, C. (2017)
Model distances for vine copulas in high dimensions
Statistics and Computing, doi:10.1007/s11222-017-9733-y

V) Killiches, M., Kraus, D. and Czado, C. (2017)
Using model distances to investigate the simplifying assumption, model selection and truncation levels for vine copulas
submitted for publication in the Australian & New Zealand Journal of Statistics

VI) Schallhorn, N., Kraus, D., Nagler, T. and Czado, C. (2017)
D-vine quantile regression with discrete variables
submitted for publication in Computational Statistics & Data Analysis

VII) Fischer, M., Kraus, D., Pfeuffer, M. and Czado C. (2017)
Stress Testing German Industry Sectors – Results from a Vine Copula Based Quantile Regression
Risks, 5, 38

Due to the publication-based character of this thesis, large parts of it are very similar to the above articles.

Contents

1 Introduction and outline

"***The most important questions of life are indeed, for the most part, only problems of probability.***" Pierre-Simon Laplace (1825)[1]

We live in a world that is more and more driven by data. The Internet, smartphones, cars, hospitals or the financial markets are only a few examples of sources generating an exploding amount of data. In the media, often the analogy of a second gold rush is made[2]. However, instead of gold, data miners, as they are often called, dig for answers by developing statistical tools to make sense of these mountains of numbers. In this context, the need for adequately describing dependencies between various random quantities is ever present. Since the seminal paper of Sklar (1959) has been published featuring the famous Sklar's Theorem, the modeling of any multivariate distribution function can be separated into a consideration of the random vector's marginal distributions and their collective dependence function, a so-called *copula*.

Despite having already been introduced in the 1950s, the transition from simple dependence concepts such as correlations to the more sophisticated concept of copulas has been inspired by papers and books published around the turn of the millennium (Embrechts et al., 1999; Joe, 1997; Nelsen, 2007). Since then, copulas have been an active field of research with applications to finance (Cherubini et al., 2004; Genest et al., 2009a,b), hydrology (Salvadori and De Michele, 2007; Favre et al., 2004; Renard and Lang, 2007) and biology (Kim et al., 2008; Nikoloulopoulos and Karlis, 2008b), just to name a few. While there exist many parametric copula families, e.g. elliptical copulas (Frahm et al., 2003; Demarta and McNeil, 2005) or Archimedean copulas (McNeil and Nešlehová, 2009), many of them lack the flexibility to adequately model high dimensional data exhibiting asymmetries and/or tail dependencies with varying characteristics. By only modeling the dependence of pairs of variables, so called pair-copula constructions (PCCs) are able to overcome these shortcomings, providing flexible copula models for arbitrary dimensions. Having been first discovered in Joe (1996) and further developed in Bedford and Cooke (2002), *vine copulas* have become the most important and frequently used PCC, especially due to the seminal work of Aas et al. (2009), in which inferential methods for vine copulas were developed.

Since then, the theoretical foundations of vines have been further explored. Panagiotelis et al. (2012) considered discrete vine copulas, Min and Czado (2010) and Gruber and Czado (2015) performed a Bayesian estimation of vine copulas, Almeida et al. (2016) discussed time-varying vine copula models and Erhardt et al. (2015) developed spatial vine copulas. Regarding the computation and implementation of vine copulas, Stöber and Czado (2012) provided a sampling algorithm, Dißmann et al. (2013) developed an algorithm for the sequential estimation of vine copulas, Stöber and Schepsmeier (2013) gave estimates of the standard errors in vine copula models and Schepsmeier (2016) discussed goodness-of-fit tests for vines. Further, Brechmann et al. (2012) considered truncated vines for parameter reduction, Krupskii and Joe (2015) combined vine copulas with factor copulas and Nagler et al. (2017) and Schellhase and Spanhel (2017) used nonparametric pair-copulas for the construction of vine copulas. In his latest book about dependence modeling with copulas, Joe (2014) dedicates an entire section to vine copulas and their properties. Additionally, an R package named `VineCopula` has been developed by Schepsmeier et al. (2017), containing implementations of many of the above methods.

[1] http://www-groups.dcs.st-and.ac.uk/history/Extras/Laplace_Probabilities.html

[2] https://www.forbes.com/sites/bradpeters/2012/06/21/the-big-data-gold-rush/#7be218afb247

Apart from the theoretical contributions, vine copulas have also found applications in numerous different fields, such as finance (Maya et al., 2015; Almeida et al., 2016; Brechmann et al., 2014; Cooke et al., 2015a), insurance (Krämer et al., 2013; Erhardt and Czado, 2012), biology (Barthel et al., 2016; Schellhase and Spanhel, 2017), energy (Czado et al., 2011), sociology (Cooke et al., 2015b), meteorology (Kauermann and Schellhase, 2014) and hydrology (Erhardt and Czado, 2015; Hobæk Haff et al., 2015; Killiches and Czado, 2015; Pereira et al., 2016; Hobæk Haff and Segers, 2015; Nikoloulopoulos et al., 2012). In the survey of Aas (2016) many more applications of vine copulas in the financial sector are presented, which is without doubt the most prominent application area of vines.

Being a publication based dissertation, the present thesis is based on several published papers and submitted manuscripts. All of them deal with the general topic of vine copula models, illuminating various aspects. Nevertheless, two main sub-topics stand out dividing the seven contributions into two groups: *D-vine copula based quantile regression* and the *simplifying assumption for vine copulas*.

The first three articles (Kraus and Czado, 2017a; Fischer, Kraus, Pfeuffer, and Czado, 2017; Schallhorn, Kraus, Nagler, and Czado, 2017) are centered around the so-called D-vine copula based quantile regression, which is a newly developed semiparametric method to estimate conditional quantiles of a response given covariates taking on certain values. The main motivation for developing this method was that the prevalent quantile regression approach, which is the linear quantile regression of Koenker and Bassett (1978), has many drawbacks. For example, as discussed in Bernard and Czado (2015), the assumption underlying the linear quantile regression method, namely that the dependence between normally distributed response and covariates is a Gaussian copula, is very restrictive and in practice almost never fulfilled. Further, quantile crossing may occur and the usual issues of linear regression appear, such as collinearity and the questions of including interactions and transformations of variables. Our approach, which is developed in Kraus and Czado (2017a), breaks away from the strong linearity assumption and uses the advantage of copulas, namely the possibility of separately modeling marginal distributions and the variables' dependence function, to flexibly model the quantiles of the response given the covariates. By estimating the marginals nonparametrically and the copula as a D-vine copula with the response as the first node, the conditional quantiles of this semiparametric model can be calculated analytically, guaranteeing fast and precise results. Further, the D-vine is estimated sequentially, adding one covariate, which increases the models conditional log-likelihood the most, after another until none of the remaining variables increases the model fit. This way, an automatic covariate selection is ensured which simultaneously ranks the covariates by importance.

In the context of *stress testing* we demonstrate the usefulness of D-vine copula based quantile regression. Stress tests have become frequent practice in the financial world and they are usually performed to assess a company's sensitivity to negative exogenous influences. Using data on credit default swaps we investigate several stress scenarios in the international banking and insurance market and find out that spillover effects are mainly induced by geographical proximity. In Fischer, Kraus, Pfeuffer, and Czado (2017) we use probabilities of default arising from a standard Merton model to investigate the interdependencies between German industry sectors. Among other results, surprisingly, stressing the financial sector does not seem to have a strong impact on the remaining industry sectors. The most severe stress scenarios were the ones originating from the Basic Materials and Cyclical Consumer Goods sectors.

The methodology introduced in Kraus and Czado (2017a) is restricted to continuous data sets, i.e. the response as well as all covariates are assumed to be continuous random variables. The extension of D-vine based quantile regression to *mixed discrete and continuous data*, such that some or all of the variables are allowed to take on countably many values, is described in Schallhorn, Kraus, Nagler, and Czado (2017). All ingredients needed for D-vine quantile regression (conditional distributions, conditional quantiles, conditional log-likelihoods), which can be expressed by bivariate building blocks themselves, are generalized by discriminating the four cases, i.e. whether the building block consists of two discrete variables, two continuous variables or one discrete and one continuous variable and vice versa. Using this general version of D-vine quantile regression we investigate a data set containing continuous as well as discrete variables. The conditional quantiles of the response *number of bike rentals* conditioned on various variables containing seasonal and climate information are estimated. We find out that the number of bike rentals is highly correlated with temperature, as long as it does not get too hot. Further, the number of bike rentals decreases with high humidity and strong wind.

The second big topic of this thesis is the simplifying assumption. It assumes that the copulas associated with conditional distributions do not depend on the specific values of the conditioning vector. It is often made to enable fast and robust inference. Nevertheless, many researchers have considered non-simplified vine copulas and the question when the simplifying assumption is valid. For example, Stöber et al. (2013) determined the decomposition of several known multivariate copulas into simplified vine copulas. Further, Hobæk Haff et al. (2010) stated that simplified vine copulas are "a rather good solution, even when the simplifying assumption is far from being fulfilled by the actual model". However, this statement was criticized to be too optimistic in Acar et al. (2012), who present cases where simplified vine copulas are insufficient. The importance of this contribution is also stressed in Oh and Patton (2017). Another critical discussion of the simplifying assumption was given in Spanhel and Kurz (2015) who focused on possible misspecifications of simplified vine copulas when the true distribution is non-simplified.
As a first approach to the topic in Killiches, Kraus, and Czado (2017a) we visually determine the differences between simplified and non-simplified vines in three dimensions. Staying in three dimensions has the advantage that the three-dimensional densities can still be visualized by plotting their contour surfaces. Furthermore, in a three-dimensional vine copula there is only one conditional pair-copula with a single conditioning variable, making it easy to isolate and interpret the effect of the simplifying assumption. We consider several scenarios of three-dimensional simplified and non-simplified vine copulas and display their contour surfaces from three different angles as well as the two-dimensional contour lines of the three bivariate marginal densities. Our analyses admit several conclusions: First, for simplified vines the shapes of the three-dimensional contours are smooth extensions of the bivariate marginals, while for non-simplified vines this is not necessarily the case. Further, the density contours of non-simplified vine copulas often exhibit twists, bumps and changing dependencies, whereas simplified vines appear to be smooth and rather convex with a more homogeneous dependence. We also see that the three-dimensional Frank copula, which cannot be decomposed as a simplified vine copula, does not severely violate the simplifying assumption, exhibiting rather constant conditional dependence. In an application we use these findings to compare the contour surfaces of the density of a three-dimensional subset of the often utilized `uranium` data set, fitted as a simplified vine, a non-simplified vine and a nonparametric kernel density fit. Since we see that the non-simplified fit is much closer to the kernel density fit than the simplified fit, we reason that this data set is best fitted by a non-simplified vine copula. This conclusion has also been drawn in Acar et al.

(2012).

While this method gives a good idea about some of the implications of the simplifying assumption, it is not a test in a strict statistical sense and only works in three dimensions. Therefore, in Killiches, Kraus, and Czado (2017c) we develop a *statistical test for the simplifying assumption* that is valid in any dimension. It is based on modified version of the Kullback-Leibler distance developed in Killiches, Kraus, and Czado (2017b), enabling fast computations even in high dimensions. Given copula data that is to be tested concerning the validity of the simplifying assumption, a simplified as well as a non-simplified vine copula model are fitted. In order to test the null hypothesis that the simplifying assumptions is fulfilled, i.e. the simplified vine copula suffices to model the data, we investigate whether the distance (measured in terms of a modified Kullback-Leibler distance) between the two fitted models is significantly different from zero. If it is, then we reject the null hypothesis in favor of the more complicated non-simplified vine copula model. However, if the distance is not significantly different from zero, we conclude that the simpler simplified vine is sufficient to represent the dependencies exhibited by the data. Since the theoretical distribution of the test statistic given by the modified KL distance between the two models cannot be derived analytically, we use a parametric bootstrapping scheme in order to determine approximate confidence intervals. In a simulation study we show that the test has a very high power. Further, we revisit the three-dimensional `uranium` data set and, using our test, once again come to the conclusion that a non-simplified vine copula is necessary to adequately model this data set. Finally, we apply our test to a four-dimensional financial data set containing the European national indices DAX, MIB, AEX and IBEX and find out that in this case a simplified vine copula suffices.

Lastly, in Kraus and Czado (2017b) we present alternatives to the widely used Dißmann algorithm, which have the goal to find vine tree structures that produce models for which the simplifying assumption is violated as little as possible. This research project was motivated by the insight that the tree structure of a vine copula (determining which pair-copulas are to be modeled) has a severe impact on whether the simplifying assumption is violated or not. Until now, the prevalent method of selecting a vine's tree structure is given by Dißmann's algorithm, a heuristic that, starting with the lowest tree, sequentially fits a vine maximizing the overall dependence of the modeled pair-copulas measured in terms of Kendall's τ. We show that this selection method might yield structures that would theoretically result in non-simplified vine copulas even though the true underlying vine is of the simplified form. This results in insufficient model fits (measured in terms of the log-likelihood), which are considerably worse than models fitted using the true tree structure. Therefore, we try to tackle this issue proposing structure selection methods that take into account whether the pair-copulas modeled violate the simplifying assumption or not. For this, we make use of a statistical test recently developed by Kurz and Spanhel (2017) testing whether the conditional correlation of a pair-copula is constant or not (implemented in the R package `pacotest`, Kurz, 2017). The latter would be a strong indicator for a violation of the simplifying assumption for the tree structure containing this pair-copula. By using the results of these tests we try to find tree structures with a desirably low number of pair-copulas violating the simplifying assumption. The first algorithm we propose selects the first tree similar to Dißmann's algorithm. Afterwards, it sequentially constructs maximum spanning trees with a mixture of Kendall's τ values and p-values from the test for constant conditional correlation as weights. The second algorithm finds a C-vine tree structure where the root nodes are selected maximizing the sum of p-values and Kendall's τ values that are allowed by the proximity condition in the next tree. We conduct an extensive simulation study to compare the performances of our newly introduced algorithms to the benchmark given by Dißmann's algorithm. We find that both our algorithms manage to outperform in high percentages of the times in different scenarios. Especially the second algorithm fitting a C-vine excels with significant comparative advantages in high dimensions. Also in real data applications we beat the performance of Dißmann's algorithm with the exception of financial data sets which seem to be adequately modeled by simplified vine copulas and therefore are not as sensitive to the chosen tree structure.

In the following sections, the methods introduced in the above mentioned papers are presented: Section 2 gives a general introduction to copulas (Section 2.1) and vine copulas (Section 2.2 and Section 2.3). Further, Section 3 deals with D-vine copula based quantile regression. To be precise, after motivating quantile regression in Section 3.1, in Section 3.2 the theory and methods of D-vine copula based quantile regression are introduced. Applications of D-vine regression to the CDS data and the German industry default probability data are presented in Section 3.3. The extension to discrete data sets are discussed in Section 3.4, together with an application to a bike sharing data set. Section 4 deals with all the aspects of the simplifying assumption we considered. The visual examination of the simplifying assumption can be found in Section 4.1. After the introduction in Section 4.1.1, several simplified (Section 4.1.2) and non-simplified (Section 4.1.3) scenarios are visualized and interpreted. An application to simulated and real data is given in Section 4.1.4 and Section 4.1.5 concludes the section. Our test for the simplifying assumption using model distances is introduced in Section 4.2. The topic is motivated in Section 4.2.1 and our modifications of the Kullback-Leibler distance are summarized in Section 4.2.2. Next, we describe the test of the simplifying assumption (Section 4.2.3) and apply it to two data sets (Section 4.2.4). Finally, Section 4.3 contains the discussion of the two new algorithms for structure selection of simplified vine copula models. After a motivation (Section 4.3.1), Dißmann's algorithm (Section 4.3.2) and the test for constant conditional correlations (Section 4.3.3) are presented. In Section 4.3.4 the two new tree selection algorithms are introduced and explained. An extensive simulation study is performed in Section 4.3.5 and several real data examples are provided in Section 4.3.6. The section is concluded in Section 4.3.7. Ultimately, Section 5 concludes this thesis with a summary and an outlook to future research.

2 Introduction to vine copulas

Parts of Section 2 are very similar to the publications Kraus and Czado (2017a), Kraus and Czado (2017b) and Killiches, Kraus, and Czado (2017a).

2.1 Copulas

A d-dimensional *copula* C is a d-variate distribution function on the unit hypercube $[0,1]^d$ with uniform marginal distribution functions. Sklar's Theorem (Sklar, 1959) provides a link between multivariate distributions and their associated copulas. It states that for every multivariate random vector $\mathbf{X} = (X_1, \ldots, X_d)' \sim F$ with marginal distribution functions $F_1, \ldots, F_d$ there exists a copula C associated with $\mathbf{X}$ such that

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

This decomposition of the multivariate distribution into its margins and its associated copula is unique when $\mathbf{X}$ is absolutely continuous (which we will generally assume in this thesis unless otherwise noted). In that case, the density of $\mathbf{X}$ can be decomposed similarly:

$$f(x_1, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)) f_1(x_1) \cdot \ldots \cdot f_d(x_d),$$

where $c(u_1, \ldots, u_d) := \frac{\partial^d}{\partial_1 \cdots \partial_d} C(u_1, \ldots, u_d)$ is the copula density and $f_1, \ldots, f_d$ are the marginal densities.

If we are interested solely in the dependence structure of $\mathbf{X}$, we consider it on the so-called *u-scale* (or copula scale) by applying the *probability integral transform* (PIT) to its marginals: $U_j := F_j(X_j)$, $j = 1, \ldots, d$. The U_j are then uniformly distributed and their joint distribution function is the copula C associated with $\mathbf{X}$. Refer to Joe (1997) and Nelsen (2007) for a detailed examination of copulas including many examples of parametric copulas, especially bivariate copulas. Those are of special interest to us since they are the building blocks used for the pair-copula construction of regular vine copulas.

2.2 D-vine copulas

Before introducing regular vine copulas, we define an important subclass known as D-vine copulas, which are of special importance for the D-vine copula based quantile regression discussed in Section 3.

For a random vector $\mathbf{X}$, a set $D \subset \{1, \ldots, d\}$ and $i, j \in \{1, \ldots, d\} \setminus D$ we use the following notation:

(a) $C_{X_i, X_j; \mathbf{X}_D}(\cdot, \cdot; \mathbf{x}_D)$ denotes the copula associated with the conditional distribution of $(X_i, X_j)'$ given $\mathbf{X}_D = \mathbf{x}_D$. We abbreviate this by $C_{ij;D}(\cdot, \cdot; \mathbf{x}_D)$. Further, $c_{ij;D}(\cdot, \cdot; \mathbf{x}_D)$ is the copula density corresponding to $C_{ij;D}(\cdot, \cdot; \mathbf{x}_D)$.

(b) By $F_{X_i|\mathbf{X}_D}(\cdot|\mathbf{x}_D)$ we denote the conditional distribution of the random variable X_i given $\mathbf{X}_D = \mathbf{x}_D$. We use $F_{i|D}(\cdot|\mathbf{x}_D)$ as an abbreviation.

(c) $C_{U_i|\mathbf{U}_D}(\cdot|\mathbf{u}_D)$ denotes the conditional distribution of the PIT random variable U_i given $\mathbf{U}_D = \mathbf{u}_D$. We abbreviate this by $C_{i|D}(\cdot|\mathbf{u}_D)$.

Following Czado (2010), the joint density f of the continuously distributed random vector $\mathbf{X}$ can be written in terms of (conditional) bivariate copula densities and its marginal densities as

$$f(x_1,\ldots,x_d) = \prod_{k=1}^{d} f_k(x_k) \prod_{i=1}^{d-1} \prod_{j=i+1}^{d} c_{ij;i+1,\ldots,j-1}\big(F_{i|i+1,\ldots,j-1}\left(x_i|x_{i+1},\ldots,x_{j-1}\right), \\ F_{j|i+1,\ldots,j-1}\left(x_j|x_{i+1},\ldots,x_{j-1}\right); x_{i+1},\ldots,x_{j-1}\big). \qquad (2.1)$$

We call this pair-copula construction (PCC) a *D-vine density* with order X_1–X_2–...–X_d. If all margins are uniform, we speak of a *D-vine copula*. As introduced by Bedford and Cooke (2002) we present a graph theoretic representation of the D-vine, where each edge of the graph corresponds to a pair-copula.

Example 2.1. Figure 1 shows an exemplary 5-dimensional D-vine corresponding to

$$\begin{aligned} f(x_1,x_2,x_3,x_4,x_5) =& f_1(x_1)f_2(x_2)f_3(x_3)f_4(x_4)f_5(x_5) \\ & \cdot c_{12}\cdot c_{23}\cdot c_{34}\cdot c_{45} && (T_1)\\ & \cdot c_{13;2}\cdot c_{24;3}\cdot c_{35;4} && (T_2)\\ & \cdot c_{14;23}\cdot c_{25;34} && (T_3)\\ & \cdot c_{15;234}, && (T_4)\end{aligned}$$

where for brevity we omitted the arguments of the pair-copulas.

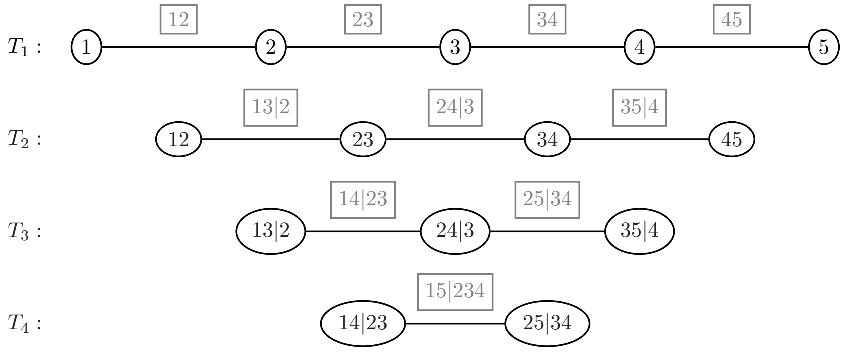

Figure 1: Graph theoretic representation of a D-vine with order X_1–X_2–X_3–X_4–X_5. The nodes of the trees are plotted in black circles and the corresponding indices of the pair-copulas in gray squares.

We see that all pair-copulas used in the decomposition appear as edges in the corresponding nested set of trees displayed in Figure 1.

In order to fit a D-vine copula with a fixed order to given data, all pair-copulas appearing in Equation (2.1) are estimated as parametric bivariate copulas. A common assumption when working with vine copulas is to assume that the copulas associated with conditional distributions $c_{i,j;D}$ do not depend on the specific values of the conditioning vector $\mathbf{x}_D$, i.e. $c_{i,j;D}(\cdot,\cdot;\mathbf{x}_D) \equiv c_{i,j;D}(\cdot,\cdot)$.
The conditional distributions $F_{i|D}\left(x_i|\mathbf{x}_D\right)$ appearing in the PCC can be evaluated using only the pair-copulas specified for the D-vine from lower trees by applying the following recursion, which was first stated by Joe (1997): Let $l \in D$ and $D_{-l} := D\setminus\{l\}$. Then,

$$F_{i|D}\left(x_i|\mathbf{x}_D\right) = h_{i|l;D_{-l}}\left(F_{i|D_{-l}}\left(x_i|\mathbf{x}_{D_{-l}}\right)|F_{l|D_{-l}}\left(x_l|\mathbf{x}_{D_{-l}}\right)\right), \qquad (2.2)$$

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

In Example 2.1 the first argument of $c_{14;23}$ from Tree 3, namely $F_{1|23}(x_1|x_2,x_3)$, can be evaluated using the h-functions associated with $C_{13;2}$, C_{12} and C_{23} from the first two trees:

$$\begin{aligned} F_{1|23}(x_1|x_2,x_3) =& h_{1|3;2}(F_{1|2}(x_1|x_2)|F_{3|2}(x_3|x_2)) \\ =& h_{1|3;2}(h_{1|2}(F_1(x_1)|F_2(x_2))|h_{3|2}(F_3(x_3)|F_2(x_2))). \end{aligned}$$

2.3 Regular vine copulas

We briefly recall the most important definitions needed for the construction of vine copulas. Details can be found in Bedford and Cooke (2002) or Aas et al. (2009). We restrict all our analyses to variables and data on the uniform copula scale $[0,1]^d$.
A d-dimensional vine copula is a pair-copula construction consisting of $d(d-1)/2$ unconditional and conditional bivariate copulas, whose structure is organized by a set of linked trees $\mathcal{V} = (T_1, \ldots, T_{d-1})$ satisfying

(i) $T_1 = (V_1, E_1)$ is a tree with nodes $V_1 = \{1, \ldots, d\}$ and edges E_1. A tree is understood as a graph where any two nodes are connected by a unique path (refer to Diestel, 2005, for an introduction to graph theory).

(ii) For $m = 2, \ldots, d-1$, the tree T_m consists of nodes $V_m = E_{m-1}$ and edges E_m.

(iii) For $m = 2, \ldots, d-1$, two nodes of T_m can only be connected by an edge if the corresponding edges of T_{m-1} have a common node.

Each edge e of the vine copula model's $d-1$ trees is associated with a bivariate pair-copula $c_{j_e,k_e;D_e}$, 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. Thus, $c_{j_e,k_e;D_e}$ is the density of the copula between random variables $U_{j_e|D_e}$ and $U_{k_e|D_e}$, where $U_{i|D} := C_{i|D}(U_i|U_D)$. The vine density can then be written as

$$c(u_1, \ldots, u_d) = \prod_{m=1}^{d-1} \prod_{e \in E_m} c_{j_e k_e;D_e}\left(C_{j_e|D_e}(u_{j_e}|\mathbf{u}_{D_e}), C_{k_e|D_e}(u_{k_e}|\mathbf{u}_{D_e}); \mathbf{u}_{D_e}\right). \tag{2.3}$$

Condition (iii) for the set of trees $\mathcal{V}$ is also known as the *proximity condition*. When one constructs the trees sequentially, starting with T_1, then it induces constraints on the nodes in the next tree which can be connected by an edge. Two types of trees play a special role regarding the proximity condition. For $m = 1, \ldots, d-2$, if tree T_m has a star like structure, i.e. there is one node j connected to all the other nodes, then the proximity condition imposes no restriction for the construction of tree T_{m+1} since all nodes share the common node j from T_m. Thus, all nodes of tree T_{m+1} are allowed to be connected. On the contrary, if tree T_m has a path like structure, i.e. each node has at most two neighbors, then all higher trees $T_{m+1}, \ldots, T_{d-1}$ are already determined by the proximity condition since for every node in T_{m+1} there exist at most two nodes that can be connected to it, resulting in a path like structure for tree T_{m+1} as well. Note that a vine structure where all trees consist of paths is nothing else than a *D-vine*. A vine with only star like trees is known as a *C-vine*.

All numerical calculations in this thesis are done using the programing language R (R Core Team, 2017), mainly using the package `VineCopula` (Schepsmeier et al., 2017). In

the following sections (unless otherwise noted) we will use the parametric bivariate copula families implemented in `VineCopula` as building blocks. This group of copulas includes Gaussian ($\mathcal{N}$), Clayton ($\mathcal{C}$), Gumbel ($\mathcal{G}$), Frank ($\mathcal{F}$), Joe ($\mathcal{J}$), Clayton-Gumbel (BB1), Joe-Frank (BB8), Tawn type 1 ($\mathcal{T}_{(1)}$) and Tawn type 2 ($\mathcal{T}_{(2)}$) copulas[3] as well as their survival versions and rotations by 90 degrees and 270 degrees (indicated by the superscripts 180, 90 or 270, respectively). The densities of the survival and rotated versions of a bivariate copula density c are given by $c^{90}(u_1, u_2) = c(1 - u_2, u_1)$, $c^{180}(u_1, u_2) = c(1 - u_1, 1 - u_2)$ and $c^{270}(u_1, u_2) = c(u_2, 1 - u_1)$. When we specify a pair-copula, we state both the family and the corresponding parameters. For example, a Gaussian copula with correlation $\rho = 0.5$ is denoted by $\mathcal{N}(0.5)$ and $\mathcal{T}_{(2)}^{270}(-3, 0.6)$ stands for a Tawn type 2 copula rotated by 270 degrees with first parameter -3 and second parameter 0.6.
The space of admissible parameters depends on the copula family. For example, whereas the parameter space of a Tawn type 1 copula is $(1, \infty) \times (0, 1)$, that of a Frank copula is given by $\mathbb{R} \setminus \{0\}$. Since we still want to compare different copula families we often transform the parameters to the same scale using Kendall's τ as a measure for the strength of dependence. See for example (Nelsen, 2007, Ch. 5.1.1) for a discussion of Kendall's τ in the context of copulas.

[3] See Appendix A.1 for a definition of the Tawn copula and its two-parametric versions used in `VineCopula`.

3 D-vine copula based quantile regression

Parts of Section 3 are very similar to the publication Kraus and Czado (2017a).

3.1 Introduction

Predicting quantiles (e.g. median or quartiles) of a random variable conditioned on other variables taking on fixed values, has continually attracted interest and found applications in various fields, especially in finance. It has become a standard tool for risk managers working on portfolio optimization, asset pricing and the evaluation of systemic risk. For example, Adrian and Brunnermeier (2016) introduce the $CoVaR$, a measure for systemic risk calculating conditional quantiles of a financial institution's loss distribution conditional on other institutions being in distress, and use it to evaluate the institution's contribution to systemic risk. A similar approach to measure systemic risk is found in Brownlees and Engle (2016). Further applications of quantile regression in the financial sector include measuring dependence in the FX markets (Bouyé and Salmon, 2009), developing pricing models for real estates (Li et al., 2013) and predicting volatilities in the stock market (Noh et al., 2015).

The literature is quite rich in methods to predict conditional quantiles. The most famous and therefore frequently used method is linear quantile regression (Koenker and Bassett, 1978) which can be seen as the expansion of the well known ordinary least squares estimation used to predict conditional means. These simple linear models have been refined to account for nonparametric effects via additive models (Koenker, 2011; Fenske et al., 2012). Further methods include local quantile regression (Spokoiny et al., 2013), single-index quantile regression (Wu et al., 2010), semiparametric quantile regression (Noh et al., 2015), nonparametric quantile regression (Li et al., 2013) and quantile regression for time series (e.g. Chen et al., 2009; Xiao and Koenker, 2009). In the machine learning context, Hwang and Shim (2005) use support vector machines for conditional quantile estimation while random forests are utilized in Meinshausen (2006). Moreover, Bouyé and Salmon (2009) propose a general approach to nonlinear quantile regression with one predictor based on a copula function.

The linear quantile regression method by Koenker and Bassett (1978) has been criticized by Bernard and Czado (2015) for imposing too restrictive assumptions on the shape of the regression quantiles. They show that for normally distributed marginals the model is misspecified as soon as the underlying dependence structure between response and covariates deviates from a Gaussian copula. Further, the method suffers from issues like quantile crossing and from the typical pitfalls of linear models such as multicollinearity, selection and significance of covariates and the inclusion of interactions or transformed variables.

In contrast, the methodology proposed in Section 3 makes no assumptions about the shape of the conditional quantiles. The dependence relationship between response and covariates is modeled flexibly using a parametric D-vine copula. Then the model's conditional quantiles can be extracted analytically without approximations or excessive computational effort. As is usual when working with copulas, we further gain from the added flexibility of separating marginal and dependence modeling.

One of the main contributions of this section is a new algorithm that sequentially fits a regression D-vine copula to given copula data, exhibiting many desirable features. On the one hand, step by step, the algorithm adds covariates to the regression model with the objective of maximizing a conditional likelihood, i.e. the likelihood of the predictive model of the response given the covariates. On the other hand, an automatic variable selection is incorporated, meaning that the algorithm will stop adding covariates to the model as soon as none of the remaining covariates is able to significantly increase the model's conditional likelihood. This results in parsimonious and at the same time flexible models whose conditional quantiles may strongly deviate from linearity. Due to the model construction, quantile crossings do not occur. Thus, the resulting D-vine quantile regression is able to overcome all the shortcomings of classical linear quantile regression mentioned above and therefore adds a new approach to the existing research on quantile regression.

3.2 Theory and methods

The main purpose of D-vine copula based quantile regression is to predict the quantile of a response variable Y given the outcome of some predictor variables $X_1, \ldots, X_d$, $d \geq 1$, where $Y \sim F_Y$ and $X_j \sim F_j$, $j = 1, \ldots, d$. Hence, the focus of interest lies on the joint modeling of Y and $\mathbf{X}$ and in particular on the *conditional quantile function* for $\alpha \in (0,1)$:

$$q_\alpha(x_1, \ldots, x_d) := F^{-1}_{Y|X_1,\ldots,X_d}(\alpha|x_1, \ldots, x_d). \tag{3.4}$$

Using the probability integral transforms (PIT) $V := F_Y(Y)$ and $U_j := F_j(X_j)$ with corresponding PIT values $v := F_Y(y)$ and $u_j := F_j(x_j)$, it follows that

$$\begin{aligned} F_{Y|X_1,\ldots,X_d}(y|x_1, \ldots, x_d) &= P(Y \leq y|X_1 = x_1, \ldots, X_d = x_d) \\ &= P(F_Y(Y) \leq v|F_1(X_1) = u_1, \ldots, F_d(X_d) = u_d) \\ &= C_{V|U_1,\ldots,U_d}(v|u_1, \ldots, u_d). \end{aligned}$$

Therefore, inversion yields

$$F^{-1}_{Y|X_1,\ldots,X_d}(\alpha|x_1, \ldots, x_d) = F_Y^{-1}\left(C^{-1}_{V|U_1,\ldots,U_d}(\alpha|u_1, \ldots, u_d)\right). \tag{3.5}$$

Hence, the conditional quantile function can be expressed in terms of the inverse marginal distribution function F_Y^{-1} of the response Y and the conditional copula quantile function $C^{-1}_{V|U_1,\ldots,U_d}$ conditioned on the PIT values of $\mathbf{x}$.
Now, we can obtain an estimate of the conditional quantile function by estimating the marginals F_Y and F_j, $j = 1, \ldots, d$, as well as the copula $C_{V,U_1,\ldots,U_d}$ and plugging them into Equation (3.5):

$$\hat{q}_\alpha(x_1, \ldots, x_d) := \hat{F}_Y^{-1}\left(\hat{C}^{-1}_{V|U_1,\ldots,U_d}(\alpha|\hat{u}_1, \ldots, \hat{u}_d)\right), \tag{3.6}$$

where $\hat{u}_j := \hat{F}_j(x_j)$ is the estimated PIT of x_j, $j = 1, \ldots, d$.
While regarding the $\hat{F}_j$ there is a vast literature about the estimation of a univariate distribution function, the question arises how to estimate the multivariate copula $C_{V,U_1,\ldots,U_d}$, such that on the one hand it facilitates a flexible model that is able to capture asymmetric dependencies, heavy tails and tail dependencies between the variables, and on the other hand the estimated conditional quantile function $\hat{C}^{-1}_{V|U_1,\ldots,U_d}(\alpha|\hat{u}_1, \ldots, \hat{u}_d)$ is easily calculable. As an answer we suggest to fit a D-vine copula to $(V, U_1, \ldots, U_d)'$, such that V is the first node in the first tree (i.e. a D-vine with order V–U_{l_1}–…–U_{l_d}, where $(l_1, \ldots, l_d)'$ is allowed to be an arbitrary permutation of $(1, \ldots, d)'$). This results in a flexible class of copulas since each bivariate copula of the pair-copula construction can be modeled separately and the order of the U_j is a parameter that can be chosen such that the conditional likelihood is maximized as will be explained in detail in the next section. Finally, the recursion given in Equation (2.2) allows us to express $C_{V|U_1,\ldots,U_d}(v|u_1, \ldots, u_d)$ in terms of nested h-functions and consequently, $C^{-1}_{V|U_1,\ldots,U_d}(\alpha|u_1, \ldots, u_d)$ in terms of inverse h-functions.

Note that $C^{-1}_{V|U_1,\dots,U_d}(\alpha|u_1,\dots,u_d)$ is monotonically increasing in α. Therefore, a crossing of quantile functions corresponding to different quantile levels is not possible. This issue of quantile crossing often arises in linear and non-linear quantile regression (e.g. see the application section of Fenske et al., 2012). Bernard and Czado (2015) show that in linear regression quantile functions may cross if non-Gaussian data is modeled. In addition, in (non-)linear quantile regression a substantial amount of effort has to be put into dealing with issues such as transforming response and covariates, including interactions among covariates and avoiding collinearity between covariates. Our approach solves these issues automatically since the distribution class given by the D-vines is much more flexible and makes less restrictive model assumptions how the covariates influence the response. This is also noted for regular vine regression by Cooke et al. (2015b).

Let in the following $\mathbf{y} := \left(y^{(i)}\right)_{i=1,\dots,n}$, $\mathcal{X} := \left(x_j^{(i)}\right)_{j=1,\dots,d,\ i=1,\dots,n}$ be n independent and identically distributed observations of the random vector $(Y, X_1, X_2, \dots, X_d)'$. The representation of $\hat{q}_\alpha(\mathbf{x})$ in Equation (3.6) allows us to divide the estimation process into two steps. In the first step we estimate the marginal distribution functions F_Y and F_j of Y and X_j, $j = 1, \dots, d$, respectively, and in the second step the D-vine that specifies the pair copulas needed to evaluate $\hat{C}^{-1}_{V|U_1,\dots,U_d}(\alpha|\hat{u}_1,\dots,\hat{u}_d)$ is estimated.

3.2.1 Estimation of the marginals

In general, we have two choices of how to fit the marginal distributions, either parametrically or nonparametrically. Since we will fit the copula in the second step parametrically, this choice will either result in a fully parametric or semiparametric estimate of $q_\alpha(\mathbf{x})$. Noh et al. (2013) point out that modeling the marginals as well as the copula parametrically might cause the resulting fully parametric estimator to be biased and inconsistent if one of the parametric models is misspecified. Therefore we prefer the semiparametric approach and estimate the marginals nonparametrically. Since we later need the inverse of the estimated marginals for the quantile prediction (c.f. Equation (3.6)) we do not want to use the discrete valued empirical distribution function for the estimation. Thus we choose the continuous kernel smoothing estimator (Parzen, 1962), which is, given a sample $\left(x^{(i)}\right)_{i=1,\dots,n}$, defined as

$$\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x^{(i)}}{h}\right), \quad x \in \mathbb{R}. \tag{3.7}$$

Here $K(x) := \int_{-\infty}^{x} k(t)dt$ with $k(\cdot)$ being a symmetric probability density function and $h > 0$ a bandwidth parameter. Usually, we choose $k = \varphi$, i.e. a Gaussian kernel, and the plug-in bandwidth developed in Duong (2016b, Equation (4)), which minimizes the asymptotic mean integrated squared error. This is implemented in the function `kcde` of the package `ks` (Duong, 2016a).

Hence, we obtain $\hat{F}_Y$ and $\hat{F}_j$ as estimates for the marginal distribution functions. We use these to transform the observed data to pseudo copula data $\hat{v}^{(i)} := \hat{F}_Y\left(y^{(i)}\right)$ and $\hat{u}_j^{(i)} := \hat{F}_j\left(x_j^{(i)}\right)$, $j = 1, \dots, d$, $i = 1, \dots, n$. The pseudo copula data $\hat{\mathbf{v}} = \left(\hat{v}^{(i)}\right)_{i=1,\dots,n}$, $\hat{\mathcal{U}} = \left(\hat{u}_j^{(i)}\right)_{j=1,\dots,d,\ i=1,\dots,n}$ is then an approximately i.i.d. sample from the PIT random vector $(V, U_1, \dots, U_d)'$ and will be used to estimate the D-vine copula in the second step.

3.2.2 Estimation of the D-vine

As motivated by Equation (3.6), we fit a D-vine with order V–U_{l_1}–…–U_{l_d} to the pseudo copula data since then the evaluation of $\hat{C}^{-1}_{V|U_1,\ldots,U_d}(\alpha|\hat{u}_1,\ldots,\hat{u}_d)$, which is needed to calculate the conditional quantile is easily feasible. For this to work, the ordering $\boldsymbol{l}=(l_1,\ldots,l_d)'$ can generally be chosen arbitrarily. However, since the explanatory power of the resulting model does depend on the particular ordering, we want to choose it such that the resulting model for the prediction of the conditional quantile has the highest explanatory power. Since it would be infeasible to compare all $d!$ possible orderings, we propose a new algorithm that automatically constructs the D-vine sequentially choosing only the most influential covariates. Similar to the `step` function for sequential estimation of linear models (cf. Venables and Ripley, 2002), starting with zero covariates, in each step we add the covariate to the model that improves the model's fit the most. As a measure for the model's fit we define the conditional log-likelihood (cll) of an estimated D-vine copula with ordering $\boldsymbol{l}$, estimated parametric pair-copula families $\hat{\boldsymbol{\mathcal{F}}}$ and corresponding copula parameters $\hat{\boldsymbol{\theta}}$ given pseudo copula data $(\hat{\mathbf{v}},\hat{\mathcal{U}})$ as

$$\operatorname{cll}\left(\boldsymbol{l},\hat{\boldsymbol{\mathcal{F}}},\hat{\boldsymbol{\theta}};\hat{\mathbf{v}},\hat{\mathcal{U}}\right):=\sum_{i=1}^{n}\log c_{V|\mathbf{U}}\left(\hat{v}^{(i)}|\hat{\mathbf{u}}^{(i)};\boldsymbol{l},\hat{\boldsymbol{\mathcal{F}}},\hat{\boldsymbol{\theta}}\right). \tag{3.8}$$

The conditional copula density $c_{V|\mathbf{U}}$ can be expressed as the product over all pair-copulas of the D-vine that contain V (see Killiches, Kraus, and Czado, 2017b):

$$\begin{aligned}c_{V|\mathbf{U}}\left(\hat{v}^{(i)}|\hat{\mathbf{u}}^{(i)};\boldsymbol{l},\hat{\boldsymbol{\mathcal{F}}},\hat{\boldsymbol{\theta}}\right)&=c_{VU_{l_1}}(\hat{v}^{(i)},\hat{u}^{(i)}_{l_1};\hat{\mathcal{F}}_{VU_{l_1}},\hat{\theta}_{VU_{l_1}})\times\\ \prod_{j=2}^{d}c_{VU_{l_j};U_{l_1},\ldots,U_{l_{j-1}}}\Big(\hat{C}_{V|U_{l_1},\ldots,U_{l_{j-1}}}&(\hat{v}^{(i)}|\hat{u}^{(i)}_{l_1},\ldots,\hat{u}^{(i)}_{l_{j-1}}),\hat{C}_{U_{l_j}|U_{l_1},\ldots,U_{l_{j-1}}}(\hat{u}^{(i)}_{l_j}|\hat{u}^{(i)}_{l_1},\ldots,\hat{u}^{(i)}_{l_{j-1}});\\ &\hat{\mathcal{F}}_{VU_{l_j};U_{l_1},\ldots,U_{l_{j-1}}},\hat{\theta}_{VU_{l_j};U_{l_1},\ldots,U_{l_{j-1}}}\Big),\end{aligned}$$

where $\hat{\mathcal{F}}_I$ and $\hat{\theta}_I$ denote the estimated family and parameter(s) of pair-copula c_I. Note that for the evaluation of the conditional copula density a specification of all pair-copulas of the D-vine is necessary since the pair-copulas not containing V are needed for the evaluation of the conditional distribution functions appearing in the arguments of the pair-copulas.

We now describe the D-vine regression algorithm, which sequentially constructs a D-vine while maximizing the model's conditional log-likelihood in each step (for a detailed code see Appendix A in Kraus and Czado, 2017a). Assume that at the beginning of the kth step of the algorithm the current optimal D-vine contains $k-1$ predictors (for illustration, see the black D-vine in Figure 2). For each of the remaining variables U_j that have not been chosen yet, we fit the pair-copulas that are needed to extend the current model to a D-vine with order V–U_{l_1}–…–$U_{l_{k-1}}$–U_j (see the gray circles).

To be precise, the pair-copula $C_{U_{l_{k-1}},U_j}$ is fitted by maximum-likelihood estimation (implemented in `VineCopula` as `BiCopSelect`) based on the data $\left(\hat{u}^{(i)}_{l_{k-1}},\hat{u}^{(i)}_j\right)_{i=1,\ldots,n}$. Subsequently, the pair-copula $C_{U_{l_{k-2}},U_j|U_{l_{k-1}}}$ is estimated. For this, we need the pseudo copula data $\left(\hat{u}^{(i)}_{l_{k-2}}|\hat{u}^{(i)}_{l_{k-1}},\hat{u}^{(i)}_j|\hat{u}^{(i)}_{l_{k-1}}\right)_{i=1,\ldots,n}$, defined as $\hat{u}^{(i)}_{l_{k-2}}|\hat{u}^{(i)}_{l_{k-1}}=\hat{C}_{U_{l_{k-2}}|U_{l_{k-1}}}(\hat{u}^{(i)}_{l_{k-2}}|\hat{u}^{(i)}_{l_{k-1}})$ and $\hat{u}^{(i)}_j|\hat{u}^{(i)}_{l_{k-1}}=\hat{C}_{U_j|U_{l_{k-1}}}(\hat{u}^{(i)}_j|\hat{u}^{(i)}_{l_{k-1}})$. Again the copula is fitted via maximum-likelihood based on this data. This is repeated until finally the pair-copula $C_{V,U_j|U_{l_1},\ldots,U_{l_{k-1}}}$ is estimated based on the pseudo copula data $\left(\hat{v}^{(i)}|\hat{\mathbf{u}}^{(i)}_{l_1,\ldots,l_{k-1}}, \hat{u}_j^{(i)}|\hat{\mathbf{u}}^{(i)}_{l_1,\ldots,l_{k-1}}\right)_{i=1,\ldots,n}$, defined as $\hat{v}^{(i)}|\hat{\mathbf{u}}^{(i)}_{l_1,\ldots,l_{k-1}} = \hat{C}_{V|U_{l_1},\ldots,U_{l_{k-1}}}(\hat{v}^{(i)}|\hat{u}^{(i)}_{l_1},\ldots,\hat{u}^{(i)}_{l_{k-1}})$ and $\hat{u}_j^{(i)}|\hat{\mathbf{u}}^{(i)}_{l_1,\ldots,l_{k-1}} = \hat{C}_{U_j|U_{l_1},\ldots,U_{l_{k-1}}}(\hat{u}_j^{(i)}|\hat{u}^{(i)}_{l_1},\ldots,\hat{u}^{(i)}_{l_{k-1}})$. Once all new pair-copulas are estimated, we can compute the resulting model's conditional log-likelihood. Finally, the current model is updated by adding the variable corresponding to the highest cll, concluding step k. That way, step by step, the covariates are ordered regarding their power to predict the response.

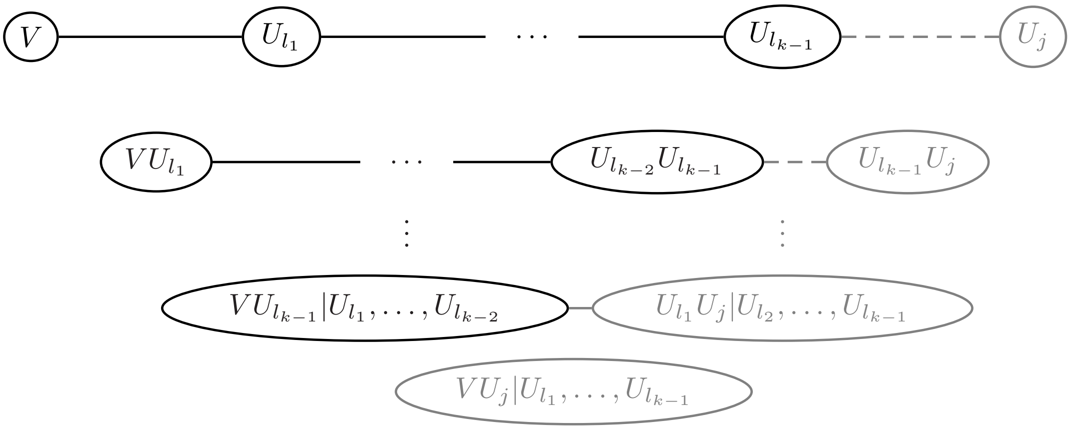
V
U_{l_1}
$\cdots$
$U_{l_{k-1}}$
U_j
VU_{l_1}
$\cdots$
$U_{l_{k-2}}U_{l_{k-1}}$
$U_{l_{k-1}}U_j$
$VU_{l_{k-1}}|U_{l_1},\ldots,U_{l_{k-2}}$
$U_{l_1}U_j|U_{l_2},\ldots,U_{l_{k-1}}$
$VU_j|U_{l_1},\ldots,U_{l_{k-1}}$

Figure 2: Extending the current D-vine (black) by adding U_j in the k-th step of the algorithm. For this purpose, the gray pair-copulas have to be estimated.

In the case that in the kth step none of the remaining covariates is able to increase the model's cll, the algorithm stops and returns the model only containing the $k-1$ chosen covariates so far. Therefore, an *automatic forward covariate selection* is accomplished resulting in parsimonious models. In order to get even more parsimonious models, we also consider two variants of the cll penalizing the number of parameters $|\hat{\boldsymbol{\theta}}|$ used for the construction of the D-vine: the AIC-corrected conditional log-likelihood $\mathrm{cll}^{\mathrm{AIC}}$, defined as

$$\mathrm{cll}^{\mathrm{AIC}}\left(\boldsymbol{l}, \hat{\boldsymbol{\mathcal{F}}}, \hat{\boldsymbol{\theta}}; \hat{\mathrm{v}}, \hat{\mathcal{U}}\right) := -2\,\mathrm{cll}\left(\boldsymbol{l}, \hat{\boldsymbol{\mathcal{F}}}, \hat{\boldsymbol{\theta}}; \hat{\mathrm{v}}, \hat{\mathcal{U}}\right) + 2|\hat{\boldsymbol{\theta}}|$$

and the BIC-corrected conditional log-likelihood $\mathrm{cll}^{\mathrm{BIC}}$, defined as

$$\mathrm{cll}^{\mathrm{BIC}}\left(\boldsymbol{l}, \hat{\boldsymbol{\mathcal{F}}}, \hat{\boldsymbol{\theta}}; \hat{\mathrm{v}}, \hat{\mathcal{U}}\right) := -2\,\mathrm{cll}\left(\boldsymbol{l}, \hat{\boldsymbol{\mathcal{F}}}, \hat{\boldsymbol{\theta}}; \hat{\mathrm{v}}, \hat{\mathcal{U}}\right) + \log(n)|\hat{\boldsymbol{\theta}}|.$$

Depending on how parsimonious the resulting model is desired to be, one can decide which version of the corrected conditional log-likelihood to use. In our applications in the later sections we always use the AIC-corrected $\mathrm{cll}^{\mathrm{AIC}}$ since in a simulation study it has shown to select the most reasonable models in the sense that unimportant variables are disregarded and influential ones are kept in most of the instances.

Example 3.1. We illustrate how the algorithm works for a four-dimensional data set $(y^{(i)}, x_1^{(i)}, x_2^{(i)}, x_3^{(i)})'$, $i = 1,\ldots,n = 500$, sampled from $(Y, X_1, X_2, X_3)' \sim \mathcal{N}_4(\mathbf{0}, \Sigma)$ with

$$\Sigma = \begin{pmatrix} 1 & 0.4 & 0.8 & 0 \\ 0.4 & 1 & 0.32 & 0 \\ 0.8 & 0.32 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

First, the data is transformed to pseudo copula data $(\hat{v}^{(i)}, \hat{u}_1^{(i)}, \hat{u}_2^{(i)}, \hat{u}_3^{(i)})'$, $i = 1, \ldots, n$, using the kernel smoothing estimators introduced in Equation (3.7).
In the **first step** of the algorithm, for each of the pairs $(V, U_j)'$, $j = 1, 2, 3$, the AIC-optimal pair-copula is chosen using the function `BiCopSelect` of the R package `VineCopula` with an independence test at level 0.05 (as described in Genest and Favre, 2007). Further, the conditional log-likelihood is calculated for each of the pairs (we omit the AIC- and BIC-corrected cll-values here since the fitted models have the same number of parameters and therefore these statistics would imply the same conclusions). The results are shown in the Table 1. Implying the largest cll, U_2 is chosen as the first variable to construct the D-vine.

Pair-copula	$\hat{C}_{V,U_1}$	$\hat{C}_{V,U_2}$	$\hat{C}_{V,U_3}$
Family	Gauss	Gauss	Indep
Parameter	0.34	0.79	0
cll	33.0	249.4	0

Table 1: Candidate models with corresponding cll after the algorithm's first step.

In the **second step**, we investigate whether the addition of either of the remaining variables U_1 or U_3 to the current D-vine can improve the conditional log-likelihood of the model. Adding U_1 would update the D-vine to order V–U_2–U_1 with newly estimated pair-copulas $\hat{C}_{V,U_1;U_2}$ (Gaussian with $\rho = 0.27$) and $\hat{C}_{U_1,U_2}$ (Gaussian with $\rho = 0.23$). The log-likelihood of the resulting conditional copula $\hat{c}_{V|U_2,U_1}$ is 269.8. The addition of U_3 would result in both new pair-copulas to be estimated as independence copulas. Consequently, the conditional log-likelihood would not improve compared to the model without U_3. Since $269.8 > 249.4$, we update the vine to order V–U_2–U_1.
In the **third step**, we check whether the addition of the remaining variable U_3 to the D-vine improves the conditional log-likelihood of the model. Not surprisingly, as in the second step the new pair copulas $\hat{C}_{V,U_3;U_2,U_1}$, $\hat{C}_{U_2,U_3;U_1}$ and $\hat{C}_{U_1,U_3}$ are all estimated to be the independence copula. Hence, the conditional log-likelihood of the full model with order V–U_2–U_1–U_3 is equal to the one with order V–U_2–U_1. Consequently, the algorithm stops and returns the D-vine with order V–U_2–U_1.

This example demonstrates the main advantages of the proposed algorithm: It automatically selects the influential covariates, ranks them by their strength of predicting the response, disregards any superfluous variables and finally flexibly models the dependence between the response and the chosen covariates. Thus, the typical issues of regression such as collinearity, transformation and inclusion/exclusion of covariates are solved without any additional effort.

3.3 Stress testing applications

3.3.1 Stress testing using CDS spreads

As an application of D-vine quantile regression to real data we want to exploit interdependencies in the financial market in order to construct global stress tests. For this purpose we consider a data set containing 1371 daily observations (01/04/2006 – 10/25/2011) of log-returns of credit default swap (CDS) spreads with 5 year maturity of 38 European, US American and Asian-Pacific financial institutions in the banking and insurance sectors. This data set has already been analyzed by Brechmann et al. (2013) who argue that CDS spreads are a viable and accurate measure of a company's creditworthiness. After applying an appropriate GARCH model to each of the univariate time series in order to get approximately i.i.d. residuals, Brechmann et al. (2013) perform stress tests. By sampling from a conditional C-vine the authors stress one company at a time (i.e. setting it to its 90%/95%/99%-quantile) and examine the joint impact on the other institutions conditioned on this stress event. With our method we can go even further and consider scenarios where more than one company is in distress. Moreover, our results are based on exact calculations of the conditional quantiles, in contrast to the Monte Carlo simulation approach used in Brechmann et al. (2013). Another difference between the approaches is that we apply a separate regression for every response while the approach of Brechmann et al. (2013) allows for a joint quantification of the single effects. In the following, we want to investigate the spillover effects of a financial crisis in a certain region or branch to other regions and branches.

The financial institutions considered in the stress test are 18 banks and 20 (re-)insurers from the regions USA, Europe and Asia-Pacific:

- **Banks**: *USA* (Goldman Sachs (GS), JP Morgan Chase (JPM), Citigroup), *Europe* (Banco Bilbao Vizcaya Argentaria (BBVA), Banco Santander (BS), Barclays, BNP Paribas, Deutsche Bank (DB), Intesa Sanpaolo, Royal Bank of Scotland (RBS), Société Générale (SG), Standard Chartered (StanCha), UBS, Unicredit), *Asia-Pacific* (Bank of China (BoC), Kookmin Bank, Sumitomo Mitsui, Westpac Banking)

- **(Re-)Insurers**: *USA* (ACE, Allstate, American International Group (AIG), Chubb, Hartford Financial Services, XL Group), *Europe* (Aegon, Allianz, Assicurazioni Generali, Aviva, AXA, Hannover Rück (HR), Legal & General (LG), Munich Re (MR), Prudential, SCOR, Swiss Re (SR), Zurich Insurance), *Asia-Pacific* (Tokio Marine (TM),QBE Insurance)

We consider three stress scenarios, corresponding to crises originating in different sectors, and investigate the resulting spillover effects. For this, we proceed as in Brechmann et al. (2013) and remove serial dependencies from each of the 38 univariate time series by fitting adequate GARCH models. The resulting residuals r_j, $j = 1, \ldots, 38$, which are approximately independent and distributed according to their model's estimated innovations distribution $\hat{F}_j$, carry the information about the dependence structure between the institutions. We consider company j to be stressed at level $\kappa \in (0,1)$, if its residual r_j takes on the 100κ%-quantile of its innovation distribution $\hat{F}_j$, i.e. $r_j = \hat{F}_j^{-1}(\kappa)$. This is equivalent to the PIT transformed variable $u_j := \hat{F}_j(r_j)$ taking on value κ. Likewise, we are interested in the resulting predicted quantile levels of the non-stressed companies. This allows us to directly work on the u-scale and consider the PIT transformed variables $u_j := \hat{F}_j(r_j)$, $j = 1, \ldots, 38$, and their dependencies.

In Scenario 1, we analyze the effect of stressing the European *systemic* banks as specified by International Monetary Fund (2009) at different stress levels. Therefore, we stress the banks Banco Santander, Barclays, BNP Paribas, Deutsche Bank, Royal Bank of Scotland, Société Générale, UBS and Unicredit at level $\kappa \in \{0.9, 0.95, 0.99\}$ (corresponding to moderate, severe and extreme stress scenarios) and use the D-vine quantile regression to estimate the conditional medians of the remaining institutions conditioned on this stress event. This way, we can assess the spillover effect to other sectors and regions. The left panel of Figure 3 shows the results of the stress test of Scenario 1. For each institution the predicted median values for the three stress levels are coded by circles (moderate stress with $\kappa = 0.9$), diamonds (severe stress with $\kappa = 0.95$) and triangles (extreme stress with $\kappa = 0.99$). For visualization, the currently stressed institutions' names are printed in ***bold and italics***. Further, solid lines separate the geographical regions (Europe in the upper, USA in the middle and Asia-Pacific in the lower panel), while dashed lines separate banks (upper) from insurance companies (lower).

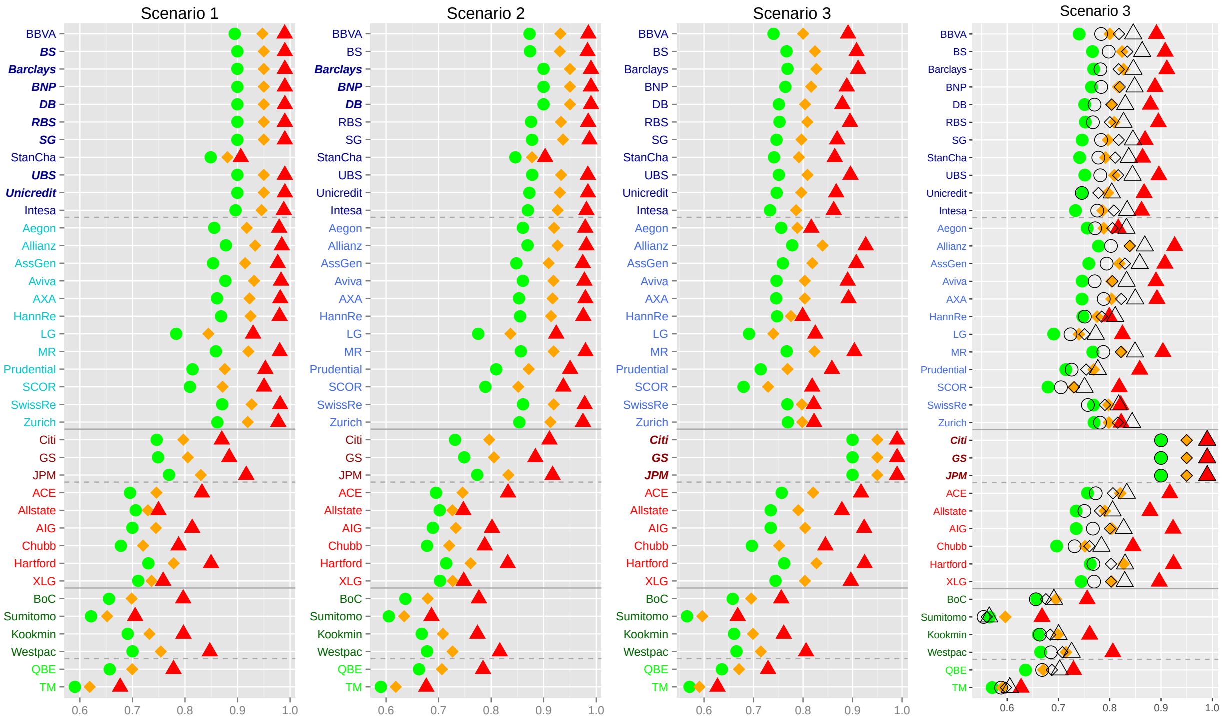

Figure 3: Stress tests stressing European systemic banks (left panel), major European banks (second panel) and US banks (third and fourth panel). For each institution the predicted median values for the three stress levels are represented by circles (moderate stress with $\kappa = 0.9$), diamonds (severe stress with $\kappa = 0.95$) and triangles (extreme stress with $\kappa = 0.99$). In the right panel the non-filled symbols represent the results of the linear quantile regression method.

We observe that the spillover effect is strongest for European insurances with predicted median values of up to 0.98 (Allianz and Aviva) for the extreme stress scenario. The comparably small values of the British bank Standard Chartered are explained by the fact that according to their annual report of 2014 (https://www.sc.com/annual-report/2014/documents/SCB_ARA_2014_full_report.pdf) 90% of the bank's income and profits are earned in Asia, Africa and the Middle East. Similar arguments holds for the British insurance company L&G with operations in Asia and the United States (http://www.legalandgeneralgroup.com/all-our-sites/). Another group that is affected quite strongly by this stress scenario are the US banks with predicted median values exceeding 0.9 in the extreme stress case. However, we observe that the geographic spillover effect is stronger than the institutional one, because the effect on European insurance companies is stronger than the effect on US banks. US insurance companies as well as the Asian-Pacific market are also affected by the stress scenario, but not as severe as the other groups.

It is interesting to see that in Scenario 2, where we only stress the three major European banks Barclays, BNP Paribas and Deutsche Bank, the results of the stress test are very similar to those of Scenario 1. We can conclude that for a crisis to evolve it suffices that only few but important banks default. In the last scenario, we analyze the spillover effect of a default of the US American banking system (see the third panel of Figure 3). Therefore, we stress the banks Citigroup, Goldman Sachs and JP Morgan Chase at level $\kappa \in \{0.9, 0.95, 0.99\}$ and estimate the conditional medians of the remaining institutions conditioned on this stress event. Again, we see the quite strong interconnectedness between US banks and insurance companies, as well as between US banks and the European market, and observe rather weak spillover effects on the Asian-Pacific sector.

Finally, in the right panel of Figure 3, we compare the stress testing results of D-vine quantile regression to those of linear quantile regression (Koenker, 2005) for Scenario 3. Additional to the filled symbols indicating the results of D-vine quantile regression, we also added the predicted medians of linear quantile regression with non-filled symbols, where circles again denote moderate stress ($\kappa = 0.9$), diamonds severe stress ($\kappa = 0.95$) and triangles extreme stress ($\kappa = 0.99$). We see that for almost all companies linear regression overestimates the moderate stress results and underestimates the extreme ones. This reflects the fact that Gaussian dependence structure implied by linear quantile regression fails to imitate the tail dependence that is typically exhibited by financial data such as these CDS log-returns. Another flaw of linear quantile regression observable from the plot is that due to the linearity of the model the median predictions for the three different stress levels seem to always have a similar distance to each other. The predictions of the D-vine based quantile regression appear as much more flexible with some narrower, wider and skewed sets of predictions for the three stress levels. The linear quantile regression results of Scenarios 1 and 2 allow for similar conclusions and are therefore omitted here.

All in all, we see that with D-vine copula based quantile regression we can extend the analysis of Brechmann et al. (2013). While they analyze the spillover effects stressing only one institution by simulating from a conditioned C-vine, with our new method we are able to perform stress tests that are conditioned on multiple banks and insurances being in distress. Our analysis admits the conclusion that the spillover effect is mainly driven by geography, so that European banks have a greater influence on European insurances than on US American banks. Further, the claim of Brechmann et al. (2013) that US banks have a stronger influence on the international financial market than European banks is not supported by our analysis. This may be due to the fact they only stress one institution at a time. We come to the conclusion that stressing the major European banks has a greater overall impact on the financial system than stressing the US American banks.

3.3.2 Stress testing German industry sector PDs

Parts of Section 3.3.2 are very similar to the publication Fischer, Kraus, Pfeuffer, and Czado (2017).

In this section we examine one-year probabilities of default (PDs) of German exchange traded corporates, observed monthly between May 2007 and September 2016 (112 observations). In a first step, the PDs are averaged over 9 industry sectors (Basic Materials, Communications, Cyclical Consumer Goods & Services, Non-cyclical Consumer Goods &

Services, Energy, Financials, Industrials, Technology, Utilities). We consider the monthly differences of the aggregated sector PDs. The differenced data series are stationary and do not exhibit any autocorrelation or volatility clustering. Therefore, no ARMA-GARCH models are necessary to account for time dependencies.
Next, we transform the differenced data to the copula scale by applying the probability integral transform using the kernel density estimators as marginal distribution functions (as described in detail in Kraus and Czado, 2017a). The corresponding contour plots with standard normal margins and Kendall's τ values are displayed in Figure 4.

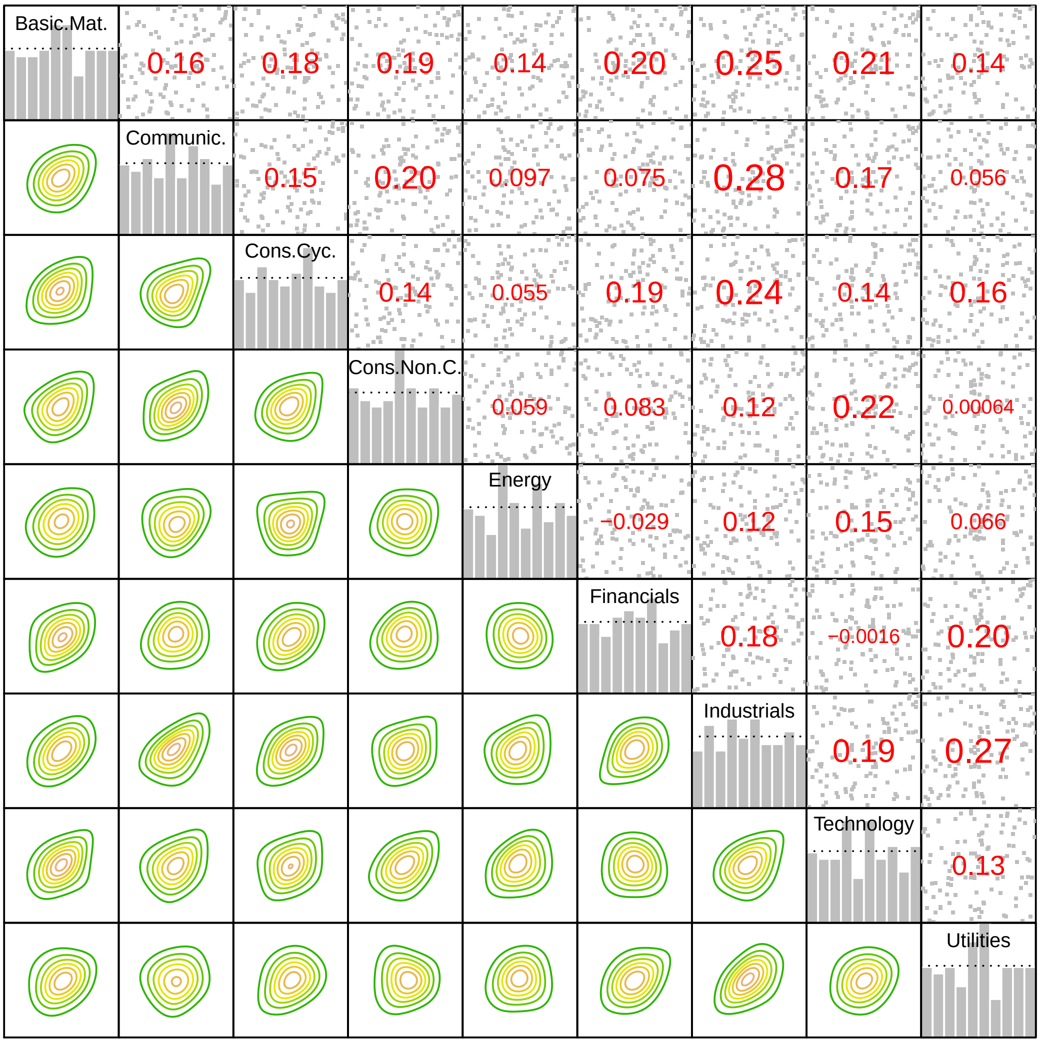

Figure 4: Upper triangular matrix: scatter plots and Kendall's τ values between pairs of aggregated sectors.
Lower triangular matrix: contour plots of densities of pairs of aggregated sectors based on empirical copulas and standard normal margins.
Diagonal: histograms of marginals after transformation to the copula scale.

The dependencies are weak to medium and mostly positive, at first glance. The Industrials sector seems to have the strongest interdependencies with the other ones. The empirical copula density contours with standard normal margins suggest that the dependencies are quite asymmetric, such that Gaussian copulas or elliptical copulas in general would not provide reasonable fits. Some pairs seem to exhibit tail dependence (e.g. Industrials and Technology). The histograms of the transformed marginals displayed on the diagonal are reasonably flat for a sample of this size.

Selected stress testing results

We perform stress tests similar to the ones described in Section 3.3.1. Large values (i.e. close to 1) of the variables on the copula scale correspond to large differences in the sector PDs. Therefore, inducing stress on an industry sector will be treated as setting the value of the respective industry sector covariate to a predetermined quantile level $\kappa \in (0, 1)$, usually $\kappa \in \{0.95, 0.99\}$. Then we use D-vine quantile regression to examine the effect of the stressed sectors (covariates) on the other sectors (responses). The predicted quantile will give information on how strongly the response sectors are affected by the stress scenario. For example, large deviations of the conditional predicted mean from the unconditional median of 0.5 imply strong effects of the stress scenario.

We present selected scenarios stressing one sector at stress levels $\kappa = 0.95$ (black) and $\kappa = 0.99$ (gray). For reasons of brevity, our focus lies on the sectors Basic Materials (upper left panel of Figure 5), Cyclical Consumer Goods (upper right panel of Figure 5), Financials (lower left panel of Figure 5) and Industrials (lower right panel of Figure 5).

As expected, the effect on the conditional quantile functions strongly depends on the specific (stressed) sector and - to some minor extent - on the concrete stress level. Across all sectors under consideration, Energy seems to be quite resistant against local sector crises. The same holds for the Utilities sector if we restrict ourselves to crises arising from Basic Materials and Cyclical Consumer Goods. On the other hand, sector crises arising from Basic Materials and Cyclical Consumer Goods spread over to most of the other sectors beside the Utilities sector. This does not hold for Financials and Industrials. In particular, stressing the sector Financials mainly affects the sector Cyclical Consumer Goods and Utilities. Above that, a simulated crisis in the Industrial sector has a significant impact on the segments Basic Materials, Communications, Cyclical Consumer Goods and Technology.

The copula families chosen for the D-vines by the algorithm were mainly ones exhibiting upper tail dependence, such as Gumbel and Joe copulas. This is in line with what we expected from the contour plots of Figure 4.

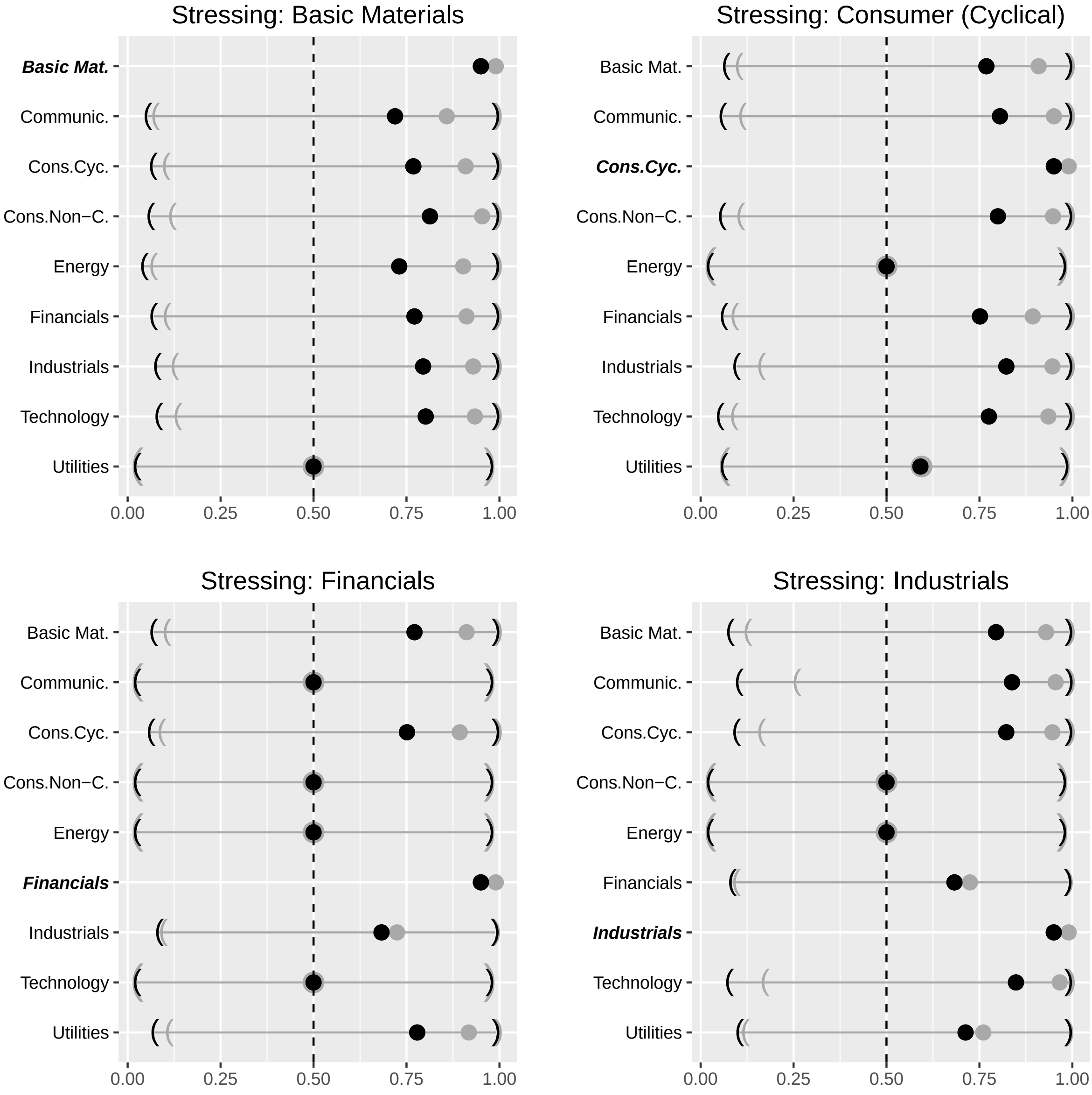

Figure 5: Stress testing results for selected industry sectors. In each plot, the sector written in bold italics is stressed at levels 95% (black) and 99% (gray). The brackets indicate the 95% prediction interval.

3.4 Extension to mixed discrete/continuous data

Parts of Section 3.4 are very similar to the publication Schallhorn, Kraus, Nagler, and Czado (2017).

3.4.1 Theory

Until now, for the method of D-vine copula based quantile regression to work, we assumed the response and all covariates to be continuous. As described in Schallhorn, Kraus, Nagler, and Czado (2017) this assumption can be relaxed, admitting that some or all of the variables are discrete. Two approaches dealing with discrete variables are discussed. The main ingredient for the first method, parametric D-vine quantile regression (PDVQR), are discrete modifications of the h-functions, introduced as

$$\tilde{h}_{i|j;D}(u|v_1, v_2) = \frac{C_{i,j;D}(u, v_1) - C_{i,j;D}(u, v_2)}{v_1 - v_2}.$$

With the help of these $\tilde{h}$-functions, the conditional distribution functions needed for D-vine quantile regression can be evaluated for all combinations of discrete and continuous variables. In a similar manner, the conditional log-likelihood used to determine the best D-vine for quantile prediction can be adapted to account for discreteness in the data. The second method introduced in Schallhorn, Kraus, Nagler, and Czado (2017) is a nonparametric D-vine quantile regression (NPDVQR), which uses the continuous convolution methods described in Nagler (2017) to make the discrete variables continuous. Afterwards, the vine copula is estimated using nonparametric estimators for the pair-copulas (Nagler et al., 2017) and the conditional quantiles are extracted as before. For details on both approaches, please refer to Schallhorn, Kraus, Nagler, and Czado (2017).

3.4.2 Application to bike sharing data

We investigate the bike sharing data set from the UCI machine learning repository (Lichman, 2013), first analyzed in Fanaee-T and Gama (2013). It contains information on rental counts from the bicycle sharing system *Capital Bikeshare* offered in Washington, D.C., together with weather and seasonal information. As a response for the quantile regression we choose the daily count of bike rentals, observed in the years 2011-2012 (731 observations). They are displayed in the left panel of Figure 6.

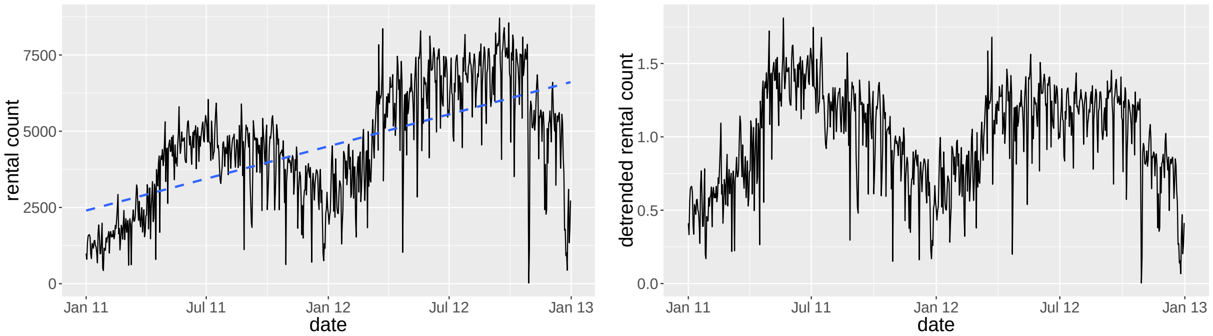

Figure 6: Observed (left) and detrended (right) bike rental counts in the years 2011-2012.

There is an obvious seasonal pattern and a linear trend reflecting a growth of the bike share system (visualized by the dashed line which is the least square linear line). While the seasonal pattern will be handled by the covariates, we cannot account for the linear trend. Therefore we remove the linear trend by dividing each observation by the least squares estimate of the linear trend. We use the division rather than the subtraction of the trend since the trend is a measure for the overall members of the bike sharing community and we are interested in the proportion of members renting bikes. The resulting detrended response is plotted in the right panel of Figure 6.

For each day we have continuous covariates *temperature* (apparent temperature in Celsius), *wind speed* (in mph) and *humidity* (relative in %). Additionally, there is the discrete variable *weather situation* giving information about the overall weather with values 1 (clear to partly cloudy), 2 (misty and cloudy) and 3 (rain, snow, thunderstorm). Further, we have information about the *season* (spring, summer, fall and winter), *month* and *weekday* of the observed day and an indicator whether the day is a *working day*.

We applied all quantile regression methods discussed in Schallhorn, Kraus, Nagler, and Czado (2017) to the bike sharing data set for the quantile levels 0.1, 0.5 and 0.9 and use 10-fold cross-validation to evaluate their out-of-sample performance. Table 2 displays the corresponding averaged cross-validated tick-losses (see e.g. Komunjer, 2013), given by $\frac{1}{731}\sum_{i=1}^{731}\rho_\alpha(y^{(i)}-\hat{q}_\alpha^{(i)})$, where $\rho_\alpha(y) = y(\alpha - \mathbb{1}(y<0))$ denotes the check function, $y^{(i)}$ is the i-th observation of the response and $\hat{q}_\alpha^{(i)}$ is the α-quantile prediction. The smallest losses and those which are not significantly larger than the smallest losses are printed in bold. Here, a Student's t test was used to test whether larger values are significantly larger than the smallest value in a row.

α	PDVQR	NPDVQR	LQR	BAQR	NPQR
0.1	**0.039**	**0.035**	0.041	**0.035**	0.090
0.5	0.082	**0.069**	0.078	**0.064**	0.250
0.9	0.042	**0.032**	0.036	**0.032**	0.295

Table 2: Averaged in-sample tick-losses of parametric D-vine quantile regression (PDVQR), nonparametric D-vine quantile regression (NPDVQR), linear quantile regression (LQR), boosted additive quantile regression (BAQR) and nonparametric quantile regression (NPQR), each applied to the bike sharing data.

NPDVQR and BAQR produce the best results, significantly beating LQR and NPQR. Between the two new D-vine copula based quantile regression methods introduced in Schallhorn, Kraus, Nagler, and Czado (2017), the nonparametric one significantly outperforms the parametric one for $\alpha = 0.5$ and $\alpha = 0.9$. The reason is that most of the covariates enter the models in a non-monotone fashion, as we will see. The ranking of the covariates by the nonparametric sequential selection algorithm is: temperature — humidity — wind speed — month — weather situation — weekday — working day — season.

In Figure 7 the influence of each of the covariates in the nonparametric D-vine quantile regression model is visualized. To be precise, for a covariate X_j we calculate for all quantile levels α of interest $\hat{q}_\alpha^{(i)} = \hat{F}_{Y|\mathbf{X}}^{-1}(\alpha|\mathbf{X} = \mathbf{x}^{(i)})$, $i = 1, \ldots, 731$, plot it against x_{ij} and add a smooth curve through the point cloud (fitted by `loess`). Figure 7 shows this for the quantile levels 0.1 (lower line), 0.5 (middle line) and 0.9 (upper line).

Higher temperatures generally go along with more bike rentals, until it gets too warm. For temperatures higher than 32 degrees Celsius, each additional degree causes a decline in bike rentals. Similar observations can be made for humidity. Bike rentals increase up to a relative humidity of around 60% and decrease afterwards. Wind speed also has a strong

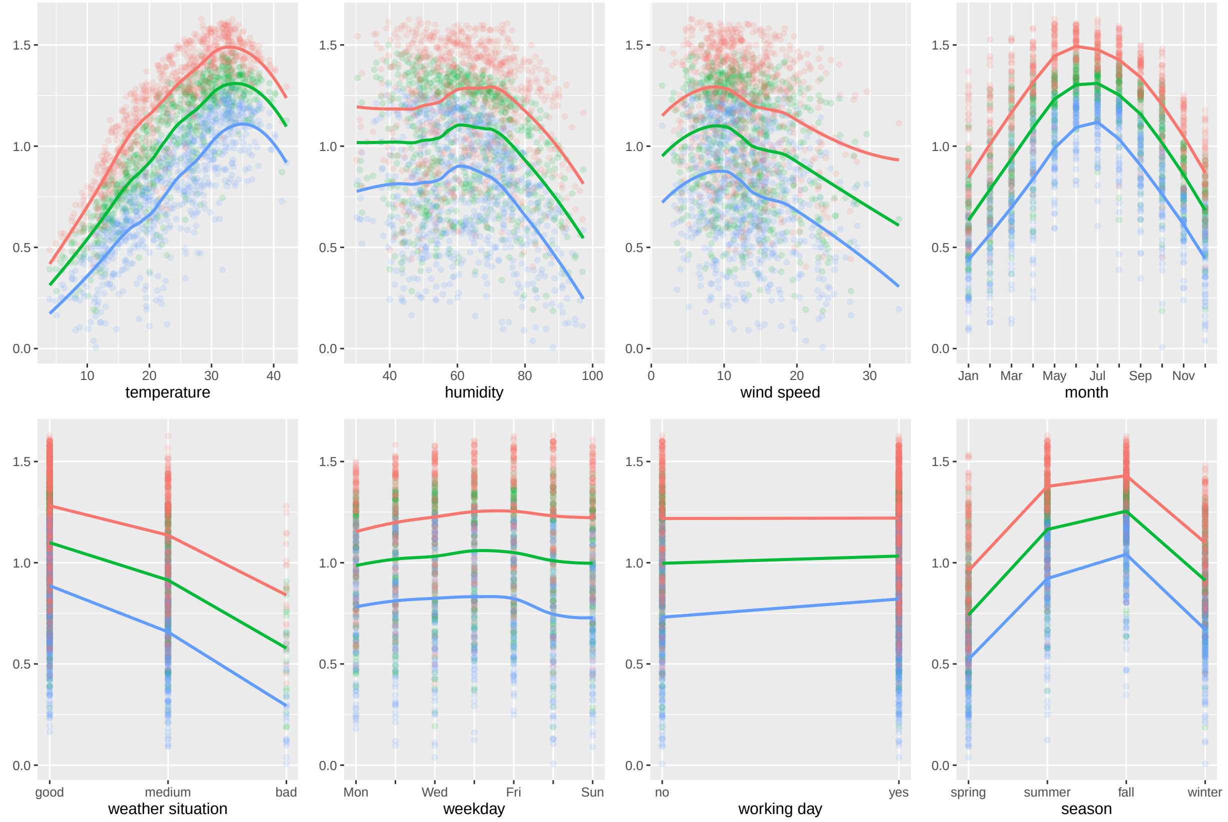

Figure 7: Influence of the different covariates on the response bike rentals using NPDVQR.

influence with fewer bike rentals on windy days. It is not surprising that the warm summer months encourage many citizens to rent bikes while in the cold winter rentals decrease on average by approximately 60%. The inclination to borrow bikes seems to grow during the week. On the weekend however, especially the 10% quantile drops considerably, which may be explained by many people leaving the city to visit their families or doing leisure activities on weekends. This is also supported by the influence of variable working day, with a few more rentals on working days. The variables weather situation and season support the thesis that more people tend to rent bicycles when the weather is good.
To investigate the differences between predictions of the various methods, we shall look more closely at the temperature variable. Figure 8 shows the effect of temperature on the predicted bike rentals using NPDVQR, PDVQR, LQR, BAQR and NPQR (from left to right).

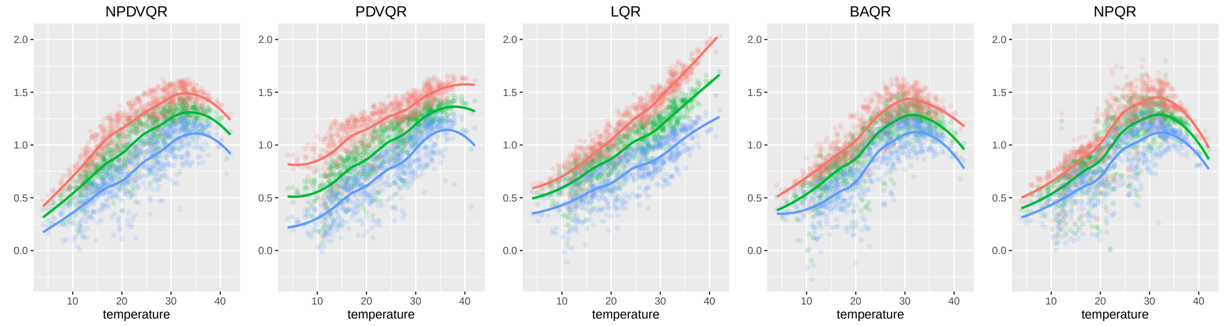

Figure 8: Influence of temperature on bike rentals for different quantile regression methods.

We see that the parametric D-vine as well as linear quantile regression are not really able to model the decline in rentals for very hot temperatures.

Apart from assessing the influence of covariates on the response, quantile regression can also be used to predict quantiles of the response in different scenarios. Suppose we know tomorrow is going to be a warm August Saturday with medium humidity and low wind-speed. Then, using our nonparametric D-vine copula based quantile regression model, we would predict a median of 8872 bikes to be rented with 10%- and 90% quantiles 7431 and 10485, respectively. In contrast, for a cold December Monday with heavy snow and high wind-speed the three predicted quantiles would be 22, 674 and 1152. As an operator of such a bike sharing system we could thus adapt our supply of rental bikes to the predicted demand.

4 Various aspects of the simplifying assumption

4.1 Examination and visualization of the simplifying assumption in three dimensions

Parts of Section 4.1 are very similar to the publication Killiches, Kraus, and Czado (2017a).

4.1.1 Introduction

When working with vine copulas one usually makes the simplifying assumption that pair-copulas of conditional distributions are independent of the values of the variables on which they are conditioned. Although enabling estimation and inference even in high dimensions, this assumption has also been criticized (e.g. Acar et al., 2012; Spanhel and Kurz, 2015). Our goal is to shed some light on the implications of this simplifying assumption by visualizing the densities of simplified and non-simplified models. For this purpose, we concentrate on the three-dimensional case. This has the advantage that the corresponding pair-copula construction contains only one copula describing the dependence between conditional variables, making the interpretation of the results easier. Further it is possible to visualize three-dimensional densities by plotting their contour surfaces. We will see that these plots contain much more information than the bivariate contour lines of the three two-dimensional margins.

4.1.2 Visualization of simplified vine copulas

The contour of a density $f\colon \mathbb{R}^d \to [0,\infty)$ corresponding to a level $y \in (0,\infty)$ is the set $\big\{\mathbf{z} \in \mathbb{R}^d \mid f(\mathbf{z}) = y\big\}$ of all points in $\mathbb{R}^d$ that are assigned the same density value y. For bivariate densities, plots of contour lines are well-known; in three dimensions this concept can be extended to contour surfaces. In this section we present contour plots of various simplified three-dimensional vine copula densities, ranging from very simple models such as a Gaussian copula, to more complex scenarios. The main goal is to get a feeling for what simplified vine copulas look like in order to properly compare them to non-simplified vine copulas. As well as the three-dimensional contour surfaces plotted from three different angles we present the contour lines of the two-dimensional marginals c_{12}, c_{23} and c_{13}. While c_{12} and c_{23} are explicitly specified in the vine copula construction, the margin c_{13} has to be calculated by integrating c_{123} with respect to u_2, either analytically (when possible) or numerically.
For all two- and three-dimensional contour plots we take the univariate marginals to have a standard normal distribution, i.e. we consider the random vector $\mathbf{Z} = (Z_1, Z_2, Z_3)^\top$, where $Z_j = \Phi^{-1}(U_j)$, $j = 1, 2, 3$, with Φ denoting the standard normal distribution function. This is done because on the uniform scale copula densities would be difficult to interpret and hardly comparable with each other. Further, in this way a Gaussian copula corresponds to a Gaussian distribution for $\mathbf{Z}$, so that all examples can be seen in comparison to this well-known case.
In Section 4.1.2 we will consider the simplified vine copula specifications from Table 3 (Scenarios 1 to 4). Later, in Section 4.1.3, we will also examine non-simplified vine copulas specifications and their simplified vine copula approximations. These are described in Scenarios 5 to 8 in Table 3. For each scenario the three pair-copulas are specified by their families and parameter(s). Further we state the corresponding Kendall's τ value in order to facilitate comparability. In the non-simplified scenarios (Scenarios 5–8) the τ values are

Table 3: Vine copula specifications considered in Section 4.1.2 (simplified, Scenarios 1 to 4) and Section 4.1.3 (non-simplified, Scenarios 5 to 8). In Scenario 7, AMH stands for the Ali-Mikhail-Haq copula. For a definition we refer to Kumar (2010). The definitions of $\tau^{(i)}_{13;2}(u_2)$, $i = 5, 6, 7, 8$, can be found in the text.

		copula c_{12}				copula c_{23}			
scenario	page	family	$\theta^{(1)}_{12}$	$\theta^{(2)}_{12}$	τ_{12}	family	$\theta^{(1)}_{23}$	$\theta^{(2)}_{23}$	τ_{23}
1	29	$\mathcal{N}$	0.6	–	0.41	$\mathcal{N}$	0.7	–	0.49
2	29	$\mathcal{C}$	2	–	0.50	$\mathcal{C}$	2	–	0.50
3	31	$\mathcal{F}$	7	–	0.56	$\mathcal{G}$	2	–	0.50
4	31	$\mathcal{T}_{(1)}$	3	0.3	0.25	$\mathcal{J}^{270}$	-2	–	-0.36
5	33	$\mathcal{N}$	0	–	0	$\mathcal{N}$	0	–	0
6	33	$\mathcal{C}$	-2	–	-0.50	$\mathcal{C}$	2	–	0.50
7	36	$\mathcal{F}$	8	–	0.60	$\mathcal{F}$	8	–	0.60
8	36	BB8	6	0.95	0.69	$\mathcal{G}^{270}$	-3.5	–	-0.71

		copula $c_{13;2}$			
scenario	page	family	$\theta^{(1)}_{13;2}(u_2)$	$\theta^{(2)}_{13;2}(u_2)$	$\tau_{13;2}(u_2)$
1	29	$\mathcal{N}$	0.5	–	0.33
2	29	$\mathcal{C}$	0.67	–	0.25
3	31	$\mathcal{N}$	-0.7	–	-0.49
4	31	BB1	2	1.5	0.67
5	33	$\mathcal{N}$	$0.9\sin(2\pi u_2)$	–	$\tau^{(5)}_{13;2}(u_2)$
6	33	$\mathcal{C}$	$9(-(u_2 - 0.5)^2 + 0.25)$	–	$\tau^{(6)}_{13;2}(u_2)$
7	36	AMH	$1 - \exp(-8u_2)$	–	$\tau^{(7)}_{13;2}(u_2)$
8	36	$\mathcal{T}_{(2)}/\mathcal{T}^{90}_{(2)}$	$\operatorname{sgn}(u_2 - 0.5)(4 - 3\cos(8\pi u_2))$	$0.1 + 0.8u_2$	$\tau^{(8)}_{13;2}(u_2)$

given by functions depending on u_2. They are defined as

$$\tau^{(5)}_{13;2}(u_2) = \frac{2}{\pi}\arcsin(0.9\sin(2\pi u_2)),$$

$$\tau^{(6)}_{13;2}(u_2) = \frac{9(-(u_2-0.5)^2+0.25)}{9(-(u_2-0.5)^2+0.25)+2},$$

$$\tau^{(7)}_{13;2}(u_2) = 1 - \frac{2}{3(1-\exp(-8u_2))} - \frac{2(1-(1-\exp(-8u_2)))^2\log(1-(1-\exp(-8u_2)))}{3(1-\exp(-8u_2))^2},$$

$$\tau^{(8)}_{13;2}(u_2) = \operatorname{sgn}(u_2-0.5)\int_0^1 \frac{t(1-t)}{A(t;u_2)}dA'(t;u_2),$$

where in the last row

$$A(t;u_2) = (0.9+0.8u_2)t + \left[(1-t)^{(4-3\cos(8\pi u_2))} + ((0.1+0.8u_2)t)^{(4-3\cos(8\pi u_2))}\right]^{(4-3\cos(8\pi u_2))^{-1}}.$$

Gaussian copula

The first scenario we consider concerns a Gaussian copula. Among others, Stöber et al. (2013) showed that every Gaussian copula can be represented as a simplified *Gaussian vine copula* (i.e. all pair-copulas are Gaussian) and vice versa. We specify the pair-copulas of the vine as follows: c_{12} is a bivariate Gaussian copula with parameter $\rho_{12} = 0.6$ (i.e. $\tau_{12} = 0.41$), c_{23} is a Gaussian copula with $\rho_{23} = 0.7$ ($\tau_{23} = 0.49$) and $c_{13;2}$ is a Gaussian copula with $\rho_{13;2} = 0.5$ ($\tau_{13;2} = 0.33$). This specification, which can be found in Table 3 (Scenario 1), directly implies that c_{13} is a Gaussian copula with $\rho_{13} = 0.71$ ($\tau_{13} = 0.50$), see for example Kurowicka and Cooke (2006), p. 69. The resulting elliptical-shaped contours displayed from three viewpoints in the top row of Figure 9 are the natural extension of the well-known ellipsoid-shaped contour plots of bivariate normal distributions. We chose the contour levels for the plots such that the four contour surfaces are representative of the entire density. For the remainder of Section 4.1 these levels are fixed with values 0.015, 0.035, 0.075 and 0.11 (from outer to inner surface). The contour plots of the two-dimensional margins in the bottom row of Figure 9 are those of bivariate normal distributions. We see that the contour plots of the bivariate margins already give a good impression of what the three-dimensional object looks like. It turns out that this property can be observed for all simplified vine copulas we will consider.

Clayton copula

A well-known representative of the class of Archimedean copulas is the Clayton copula. It is a one-parametric family with lower tail dependence. The Clayton copula is the copula underlying the multivariate Pareto distribution and is the only Archimedean copula that can be represented as a simplified vine copula as proved in Stöber et al. (2013), Theorem 3.1. It is easy to see that the bivariate margins of a three-dimensional Clayton copula with parameter θ are bivariate Clayton copulas with parameter θ, see for example Kraus and Czado (2017a), Appendix B. There it was also shown that the copula of the conditioned variables (in our decomposition $c_{13;2}$) again is a Clayton copula, in this case with parameter $\theta/(\theta+1)$. Hence, in order to obtain a three-dimensional Clayton copula with parameter $\theta = 2$ we specify a vine copula as described in Scenario 2 of Table 3. The top row of Figure 10 displays the contours of the resulting copula, the strong lower tail dependence is clearly visible. As already stated, the (unconditional) bivariate margin c_{13} is also a Clayton copula with parameter 2 and therefore all contour plots of the margins in the bottom row of Figure 10 are identical. Again we observe that the shape of the contours of the three-dimensional density is anticipated quite well already by the two-dimensional marginal plots.

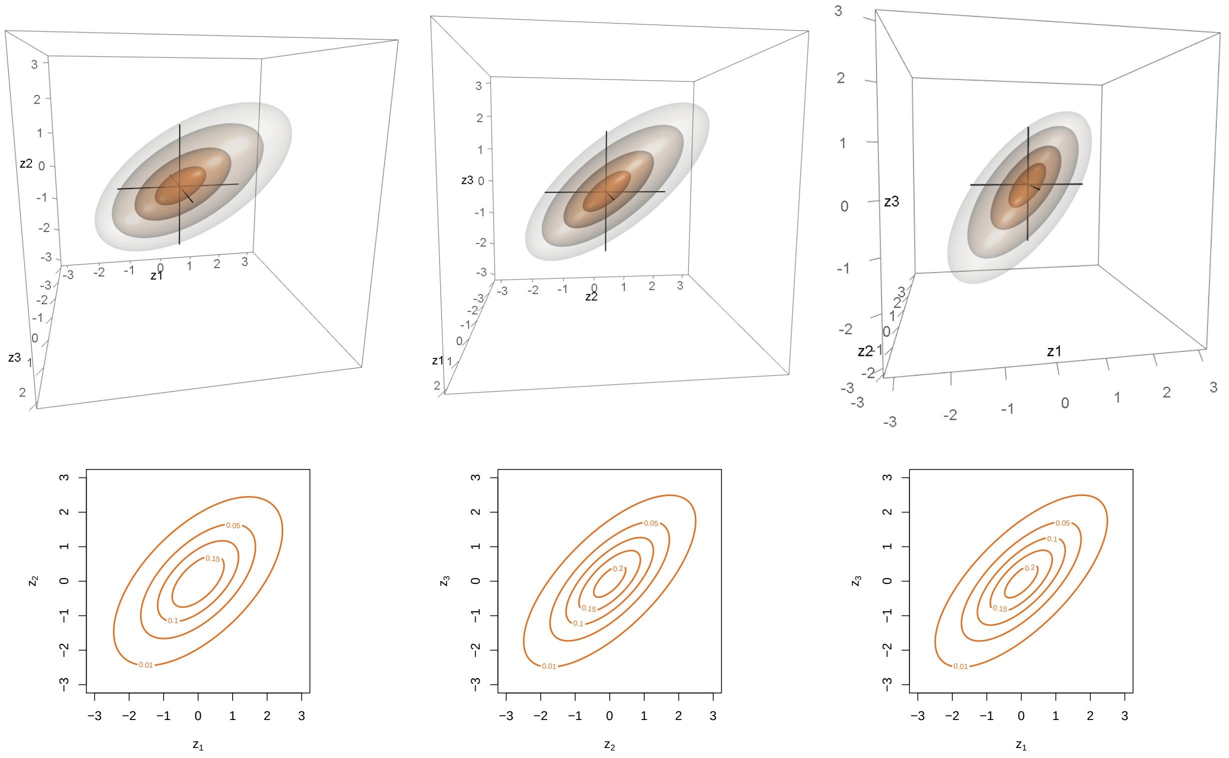

Figure 9: Top: Contours of the three-dimensional vine copula density specified by c_{12}: $\mathcal{N}(0.6)$, c_{23}: $\mathcal{N}(0.7)$, $c_{13;2}$: $\mathcal{N}(0.5)$. Bottom: Contours of the corresponding bivariate marginal densities.

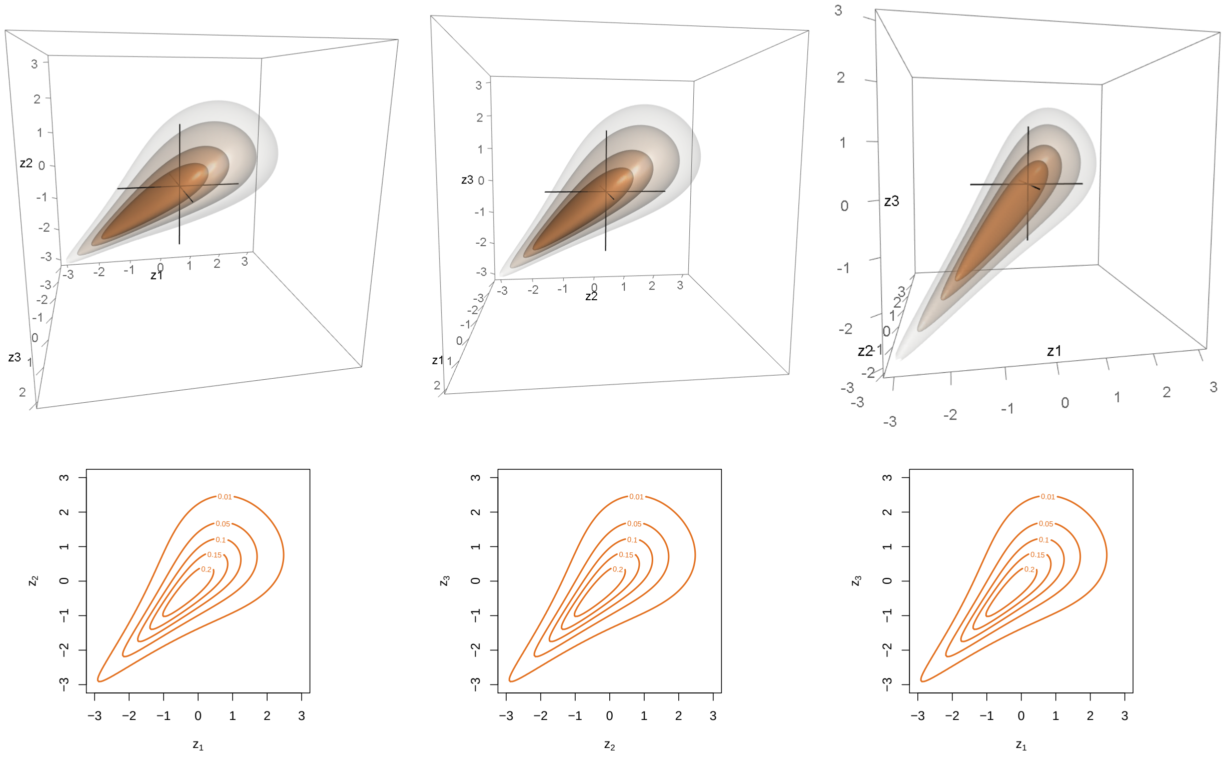

Figure 10: Top: Contours of the three-dimensional vine copula density specified by c_{12}: $\mathcal{C}(2)$, c_{23}: $\mathcal{C}(2)$, $c_{13;2}$: $\mathcal{C}(0.67)$. Bottom: Contours of the corresponding bivariate marginal densities.

Mixed simplified vine copula 1

Up to now we have only considered vine copulas where all pair-copulas belong to the same family of parametric copulas. Of course, one of the main advantages of vine copulas is that one can specify each pair to be from a different copula family with its own parameter(s). The resulting model class is very flexible and able to describe many different kinds of dependencies. As an example for this, we present Scenario 3 (Table 3): c_{12} is a bivariate Frank copula with parameter $\theta_{12} = 7$ (i.e. $\tau_{12} = 0.56$), c_{23} is a Gumbel copula with $\theta_{23} = 2$ ($\tau_{23} = 0.5$) and $c_{13;2}$ is a Gaussian copula with $\rho_{13;2} = -0.7$ ($\tau_{13;2} = -0.49$). In the resulting contour plots of Figure 11 (top row) one can clearly see the shapes of the Frank and the Gumbel copula in the left and the middle plot, respectively. Although the dependency of each pair-copula is fairly strong, we observe rather weak dependence for c_{13}. The negative conditional dependence seems to cancel out with the positive dependencies implied by c_{12} and c_{23} (compare Figure 11, bottom row).

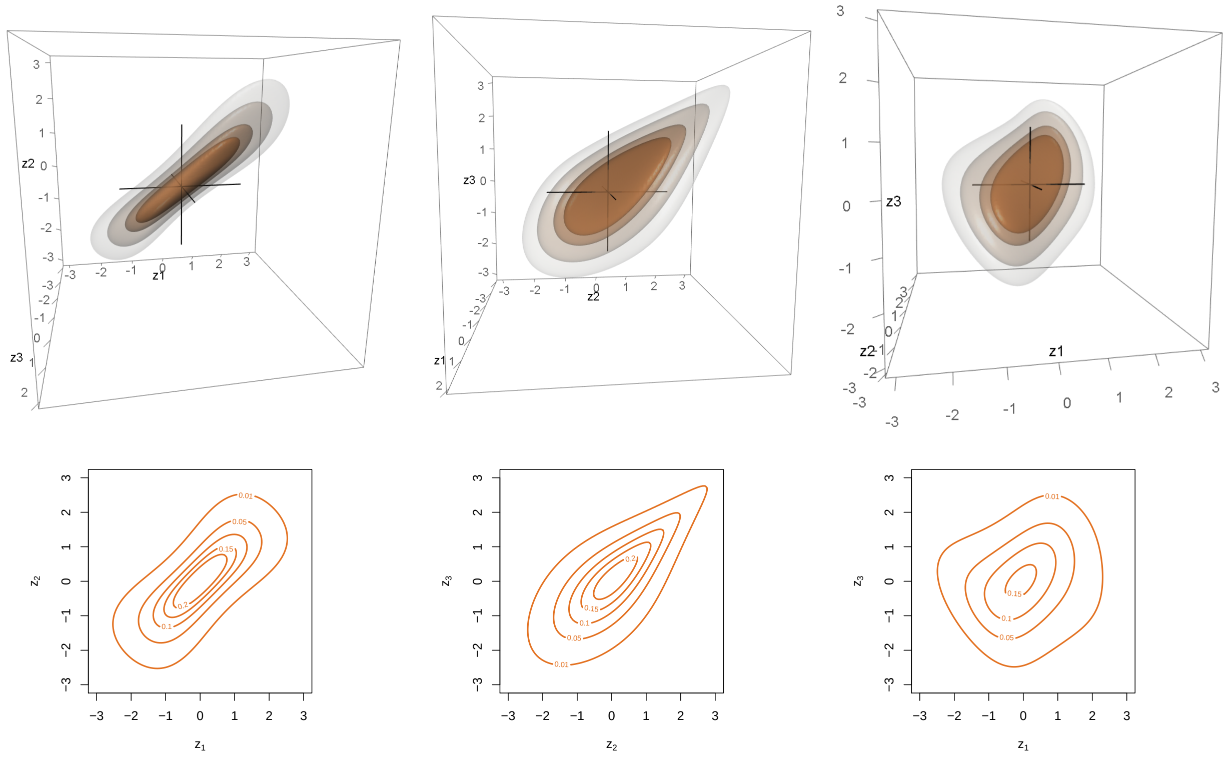

Figure 11: Top: Contours of the three-dimensional vine copula density specified by c_{12}: $\mathcal{F}(7)$, c_{23}: $\mathcal{G}(2)$, $c_{13;2}$: $\mathcal{N}(-0.7)$. Bottom: Contours of the corresponding bivariate marginal densities.

Mixed simplified vine copula 2

We consider a second example of a mixed vine copula (Scenario 4 from Table 3) with the following specifications: c_{12} is a Tawn Type 1 copula with parameters $\boldsymbol{\theta}_{12} = (3, 0.3)^\top$ (implying $\tau_{12} = 0.25$), c_{23} is a Joe copula rotated by 270 degrees with $\theta_{23} = -2$ ($\tau_{23} = -0.36$) and $c_{13;2}$ is a BB1 copula with $\boldsymbol{\theta}_{13;2} = (2, 1.5)^\top$ ($\tau_{13;2} = 0.67$). The shape of the resulting contours in the top row of Figure 12 appears to be very non-standard. Especially the dependence between the first and third marginals is quite contorted. The dependence structure of the copula of the conditioned variables (BB1) cannot be detected at all. Further the non-exchangeable nature of the Tawn copula is noticeable both in the three- and the two-dimensional contour plots (cf. Figure 12, bottom row).

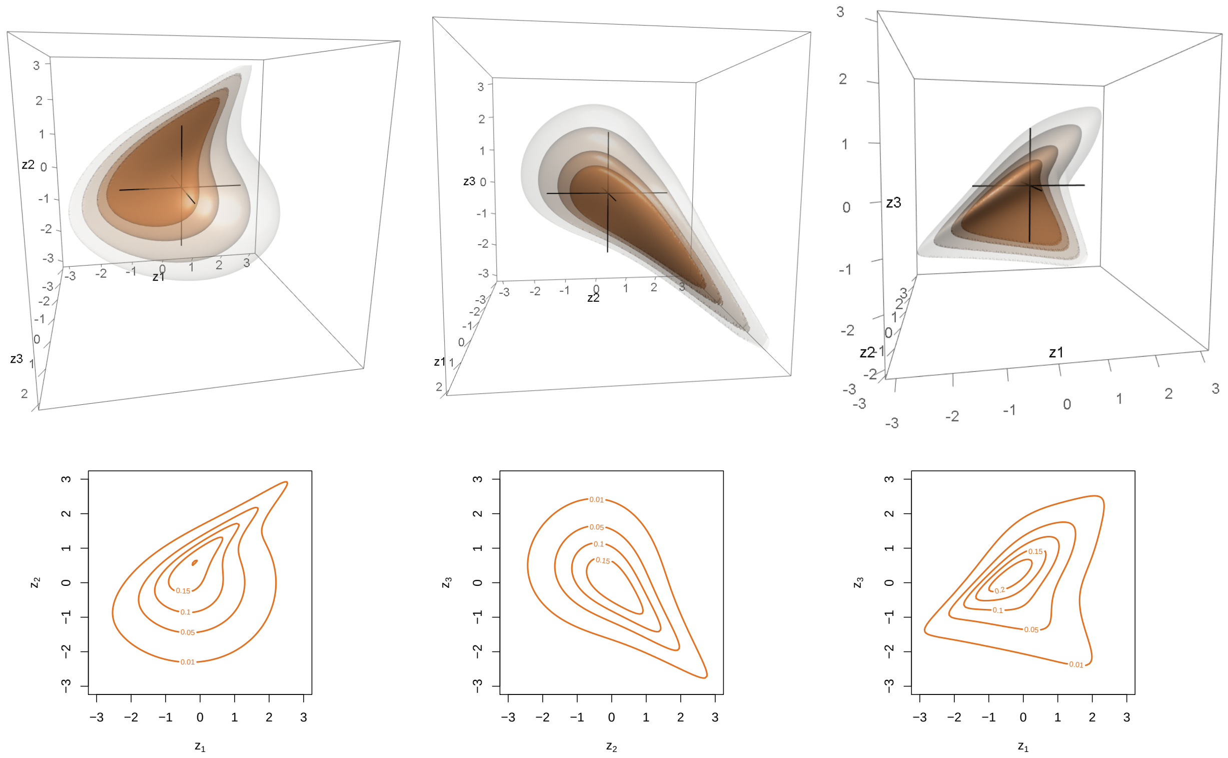

Figure 12: Top: Contours of the three-dimensional vine copula density specified by c_{12}: $\mathcal{T}_{(1)}(3, 0.3)$, c_{23}: $\mathcal{J}^{270}(-2)$, $c_{13;2}$: BB1$(2, 1.5)$. Bottom: Contours of the corresponding bivariate marginal densities.

Note that even for the rather bent examples in this section the shape of the bivariate marginal contour plots resembles what we see in the three-dimensional plots. Thus all considered simplified vine copulas share the property that knowledge of just the three bivariate margins already provides a fairly good idea of the shape of the contours of the three-dimensional copula density.

4.1.3 Visualization of non-simplified vine copulas

Having seen several examples of visualized simplified vine copulas we also aim to visually explore the meaning of the simplifying assumption. For this purpose we now present a series of contour plots of non-simplified vine copula densities and compare them to the ones of the corresponding simplified vine copula approximations. Similar to Hobæk Haff et al. (2010) and Stöber et al. (2013) we determine the simplified vine copula approximation of a non-simplified vine copula with pair-copulas c_{12}^{NS}, c_{23}^{NS} and $c_{13;2}^{\text{NS}}$ by setting the unconditional pair-copulas c_{12}^{S} and c_{23}^{S} to the true ones (c_{12}^{NS} and c_{23}^{NS}, respectively) and finding the pair-copula $c_{13;2}^{\text{S}}$ (independent of u_2) that minimizes the Kullback-Leibler distance to the true conditional copula $c_{13;2}^{\text{NS}}$. Since in most scenarios considered in this chapter this minimization is analytically infeasible, we estimate $c_{13;2}^{\text{S}}$ by generating a sample $\left(u_1^{(i)}, u_2^{(i)}, u_3^{(i)}\right)_{i=1,\ldots,N}^{\top}$ of size N from the non-simplified model and fitting the likelihood maximizing parametric bivariate copula $c_{13;2}^{\text{S}}$ to the pseudo copula data $\left(u_{1|2}^{(i)}, u_{3|2}^{(i)}\right)_{i=1,\ldots,N}^{\top}$, where $u_{j|2}^{(i)} = C_{j|2}^{S}\left(u_j^{(i)}|u_2^{(i)}\right)$, $j = 1, 3$. Even though we found that the estimated parameters had converged up to the second digit for $N = 10{,}000$ we used $N = 100{,}000$ due to low computational effort.
In Section 4.1.3 we will consider the non-simplified vine copula specifications from Table 3 (Scenarios 5 to 8).

Gaussian vine copula with sinusoidal conditional dependence function

In our first non-simplified example (Scenario 5 from Table 3) we set the two unconditional copulas c_{12} and c_{23} to the independence copula in order to isolate the effect of the dependence of the conditional copula $c_{13;2}$ on u_2. We choose $c_{13;2}$ to be Gaussian with parameter function $\rho_{13;2}(u_2) = 0.9\sin(2\pi u_2)$, i.e. one full period of a sine curve with amplitude 0.9. Hence for values of u_2 ranging between 0 and 0.5 the dependence is positive with Kendall's τ between 0 and 0.71 and negative for $0.5 < u_2 < 1$ (see also the left panel of the second row of Figure 13). The shift from positive to negative dependence is clearly visible in the contour plots shown in the top row of Figure 13. We also observe that the resulting contour surfaces for higher levels are no longer connected and the density is bimodal. Further, from the numerically integrated contour plot of c_{13} in the right panel of the second row of Figure 13 we conclude that marginally the strong positive and negative dependencies cancel each other out resulting in a bivariate marginal copula with almost no dependence, resembling a t copula with association of zero and low degrees of freedom.

In opposition to the simplified examples from Section 4.1.2, now the bivariate contour plots do no longer anticipate the three-dimensional object in a reasonable way. The sinusoidal structure of this copula cannot be guessed from the two-dimensional plots in the second row of Figure 13. In fact, had we used $\rho_{13;2}(u_2) = -0.9\sin(2\pi u_2)$, the copula density would have changed drastically (90 degree rotation along the z_2-axis) while the bivariate margins would have stayed exactly the same.

In contrast the corresponding simplified vine copula approximation, whose contours are displayed in the third row of Figure 13, resembles the smooth extension of the bivariate margins (bottom row of Figure 13) to three dimensions. This unimodal simplified vine copula, whose conditional copula $\hat{c}_{13;2}$ is indeed a t copula with almost no dependence ($\hat{\rho}_{13;2} = 0.01$) and low degrees of freedom ($\hat{\nu}_{13;2} = 2.15$), is not able to reproduce the twisted shape of the non-simplified vine copula at all. Also the implied bivariate margin of the first and third variable in the right panel of the bottom row of Figure 13 is almost identical to the one of the non-simplified copula in the second row.

Clayton vine copula with quadratic conditional dependence function

Next we consider a non-simplified Clayton vine copula, i.e. all pair-copulas are bivariate Clayton copulas where the parameters of the unconditional copulas may differ in contrast to the three-dimensional Clayton copula (cf. Scenario 2) for which the parameters of c_{12} and c_{23} have to coincide. In this Scenario 6 we set the dependencies of the unconditional pair-copulas as $\theta_{12} = -2$ ($\tau_{12} = -0.5$) and $\theta_{23} = 2$ ($\tau_{23} = 0.5$) and specify the parameter function as a downwardly open parabola taking only non-negative values: $\theta_{13;2}(u_2) = 9(-(u_2-0.5)^2 + 0.25)$. The corresponding $\tau_{13;2}$ values range between 0 and 0.53 and take their maximum for $u_2 = 0.5$ (see the left panel of the second row in Figure 14). The contours of the resulting density shown in the top row of Figure 14 bear some resemblance to those of the Clayton copula (cf. Figure 10) but are much more distorted. Especially the relationship between the first and third variables seems to change from positive to negative dependence for different values of the second variable. This implies that also the conditional copula of U_1 and U_3 given $U_2 = u_2$ exhibits a change from positive to negative dependence, which is an obvious indicator that the vine copula is non-simplified. The contours of the bivariate margin c_{13} in the right panel of the second row of Figure 14 also have a bent shape which is far from any of the standard parametric copulas. The bivariate contour plots of c_{12} and c_{23} suggest a smooth shape of the contours of the three-dimensional density such that one would not expect them to look as distorted as they do in the left and middle plots of the top row of Figure 14.

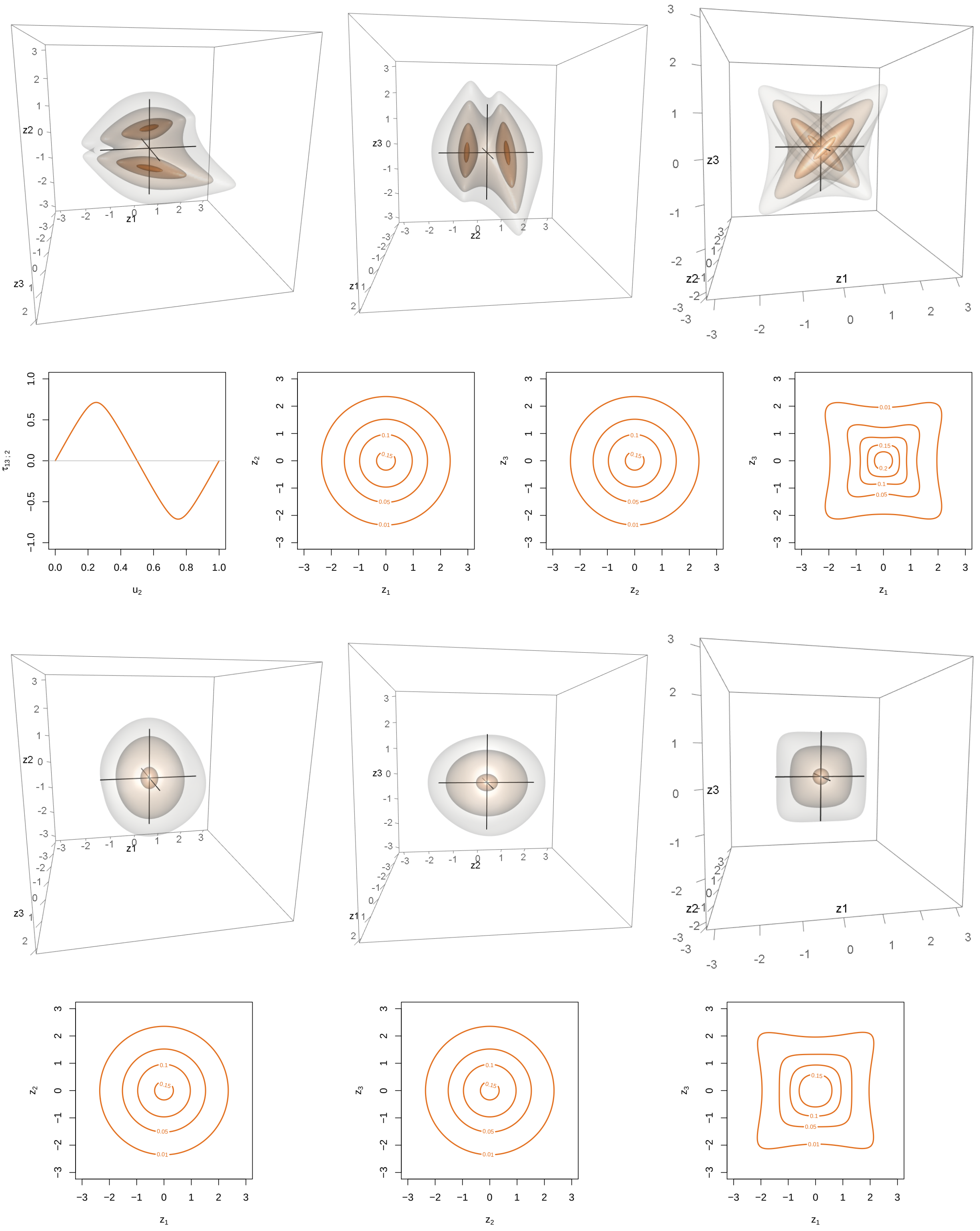

Figure 13: Top row: Contours of the three-dimensional non-simplified vine copula density specified by c_{12}: $\mathcal{N}(0)$, c_{23}: $\mathcal{N}(0)$, $c_{13;2}$: $\mathcal{N}(\rho_{13;2}(u_2))$ with $\rho_{13;2}(u_2) = 0.9\sin(2\pi u_2)$. Second row: $\tau_{13;2}$ depending on u_2 and contours of the bivariate margins corresponding to the specification of the top row. Third row: Contours of the three-dimensional simplified vine copula approximation specified by c_{12}: $\mathcal{N}(0)$, c_{23}: $\mathcal{N}(0)$, $\hat{c}_{13;2}$: $\mathrm{t}(0.01, 2.15)$. Bottom row: Contours of the bivariate margins corresponding to the specification of the third row.

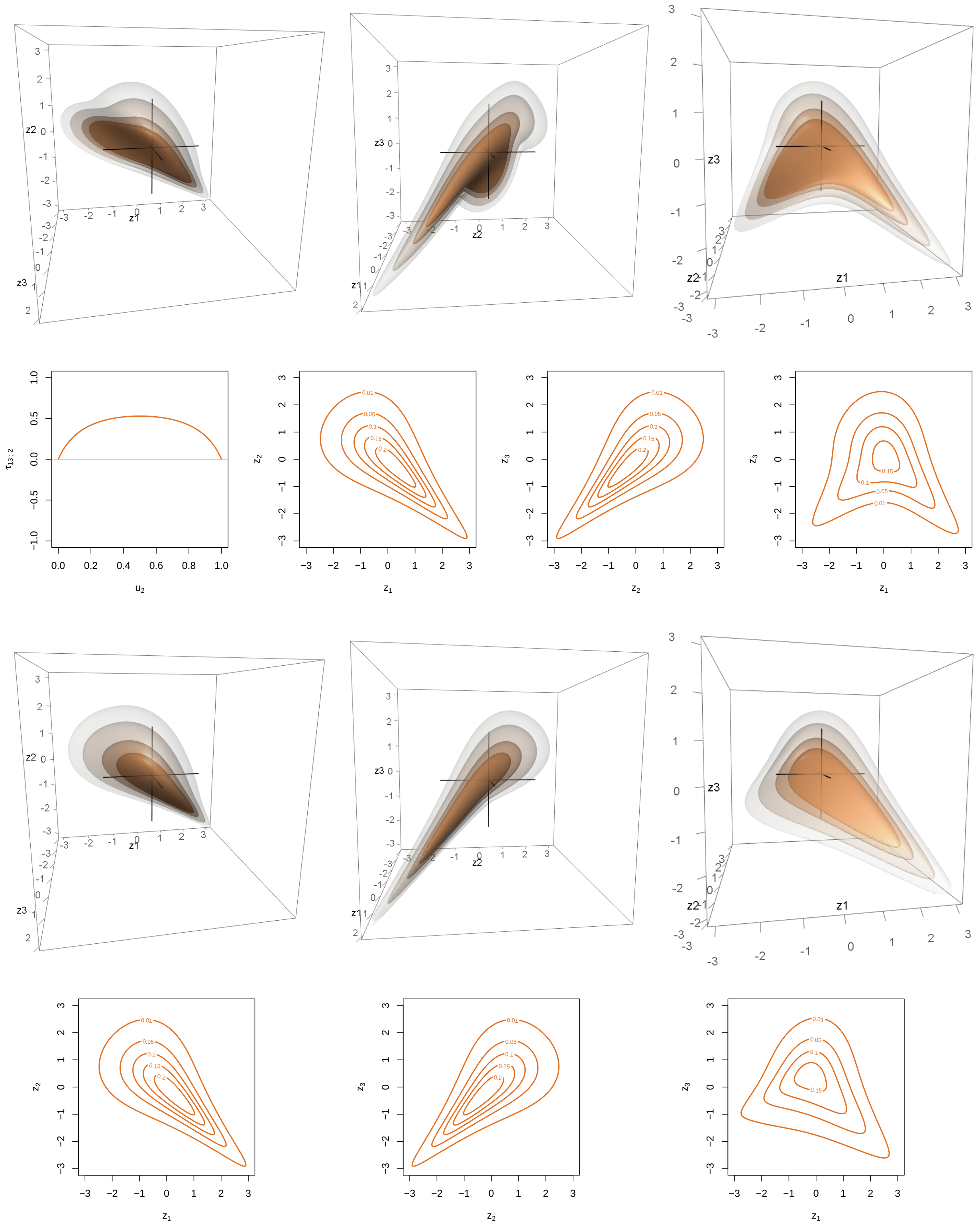

Figure 14: Top row: Contours of the three-dimensional non-simplified vine copula density specified by c_{12}: $\mathcal{C}^{90}(-2)$, c_{23}: $\mathcal{C}(2)$, $c_{13;2}$: $\mathcal{C}(\theta_{13;2}(u_2))$ with $\theta_{13;2}(u_2) = 9(-(u_2-0.5)^2+0.25)$. Second row: $\tau_{13;2}$ depending on u_2 and contours of the bivariate margins corresponding to the specification of the top row. Third row: Contours of the three-dimensional simplified vine copula approximation specified by c_{12}: $\mathcal{C}^{90}(-2)$, c_{23}: $\mathcal{C}(2)$, $\hat{c}_{13;2}$: $\text{BB6}^{180}(1.75, 1.16)$. Bottom row: Contours of the bivariate margins corresponding to the specification of the third row.

Again the simplified vine copula approximation, which uses a survival BB6 copula with parameter $\hat{\boldsymbol{\theta}}_{13;2} = (1.75, 1.16)$ as an approximation of the conditional copula $c_{13;2}$ (implying a Kendall's τ of 0.39), exhibits exactly this smooth behavior implied by the bivariate margins (see the third row of Figure 14). We further observe in the right plots of the last two rows of Figure 14 that the simplified vine copula approximation is not able to reproduce the altering dependence pattern between U_1 and U_3 due to its constant conditional dependence parameter.

Three-dimensional Frank copula

In contrast to the Clayton copula, which can be expressed as a simplified vine copula (cf. Section 4.1.2), we now turn our attention to an Archimedean copula without this property, the three-dimensional Frank copula. Its non-simplified vine decomposition can be found as Scenario 7 in Table 3 (with dependence parameter $\theta = 8$): The bivariate margins are again Frank copulas with the same dependence parameter θ (with corresponding Kendall's τ values equal to 0.6). The copula of the conditioned variables $c_{13;2}$ is also Archimedean, belonging to the Ali-Mikhail-Haq (AMH) family with functional dependence parameter $\gamma_{13;2}(u_2) = 1 - \exp(-\theta u_2)$ (see Kumar, 2010; Spanhel and Kurz, 2015). The corresponding τ values displayed in the left panel of the second row of Figure 15 show that the simplifying assumption is not severely violated. The strength of dependence is almost constant with the exception of small τ values for $u_2 < 0.2$. The contours depicted in the top row of Figure 15 exhibit the typical bone shape known from the two-dimensional contour plots of bivariate Frank copulas such as those shown in the second row of Figure 15. In order to assess how severe of a restriction the simplifying assumption would impose for modeling data generated by a Frank copula, we also present in the last two rows of Figure 15 the three- and two-dimensional contours of the simplified vine copula approximation of the Frank copula, respectively. For the trivariate Frank copula it is possible to analytically calculate the bivariate copula $c_{13;2}(\cdot, \cdot)$ that minimizes the Kullback-Leibler divergence to the conditional copula $c_{13;2}(\cdot, \cdot; u_2)$, for details see Spanhel and Kurz (2015). The visual difference between the Frank copula and its simplified vine copula approximation seems almost negligible. Only from the angle where the dependence between the first and third variable is visible the two contour plots can be distinguished (see the right plots of the first and third row of Figure 10). In the lower tail, where values of z_2 are small, the contours of the simplified vine copula approximation exhibit a higher dependence than the ones of the Frank copula, whose contours are less drawn into the corner implying less dependence. This is in line with what we would expect since the dependence function of $c_{13;2}$ of the non-simplified vine copula is decreasing for u_2 going to 0, while the dependence of $c_{13;2}$ of the simplified vine copula approximation is constant at $\tau_{13;2} = 0.28$.

Mixed non-simplified vine copula

The last example we consider is Scenario 8 (see Table 3). It is more extreme featuring pair-copulas with high dependence and more involved functions for the parameters of $c_{13;2}$. We specify c_{12} as a BB8 copula with parameters $\boldsymbol{\theta}_{12} = (6, 0.95)^\top$ (implying $\tau_{12} = 0.69$), c_{23} as a Gumbel copula rotated by 270 degrees with $\theta_{23} = -3.5$ ($\tau_{23} = -0.71$) and $c_{13;2}$ as a Tawn Type 2 copula with both parameters depending on u_2 via the functions $\operatorname{sgn}(u_2 - 0.5)(4 - 3\cos(8\pi u_2))$ and $\theta^{(2)}_{13;2}(u_2) = 0.1 + 0.8u_2$. The corresponding τ values ranging between -0.39 and 0.71 are shown in the left panel of the second row of Figure 16. For the values of $u_2 < 0.5$ that imply negative dependence we use the 90 degree rotated version of the Tawn type 2 copula. Figure 16 (top row) displays the contour plots of the resulting density. This is by far the most contorted density. The four peaks of the $\tau_{13;2}$ function

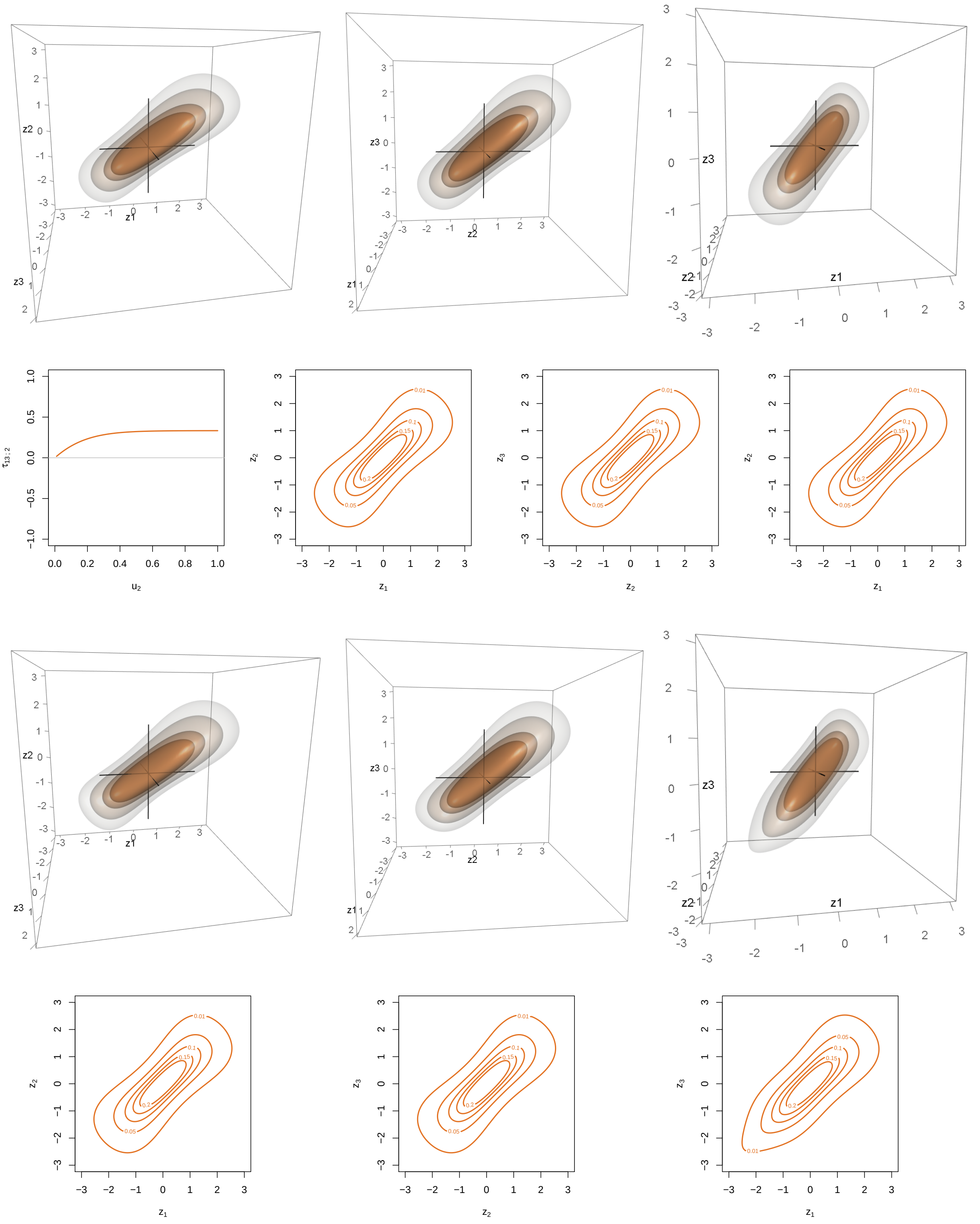

Figure 15: Top row: Contours of the three-dimensional non-simplified vine copula density specified by c_{12}: $\mathcal{F}(8)$, c_{23}: $\mathcal{F}(8)$, $c_{13;2}$: AMH$(\gamma_{13;2}(u_2))$ with $\gamma_{13;2}(u_2) = 1 - \exp(-8u_2)$. Second row: $\tau_{13;2}$ depending on u_2 and contours of the bivariate margins corresponding to the specification of the top row. Third row: Contours of the three-dimensional simplified vine copula approximation specified by c_{12}: $\mathcal{F}(8)$, c_{23}: $\mathcal{F}(8)$, $c_{13;2}$: see Spanhel and Kurz (2015). Bottom row: Contours of the bivariate margins corresponding to the specification of the third row.

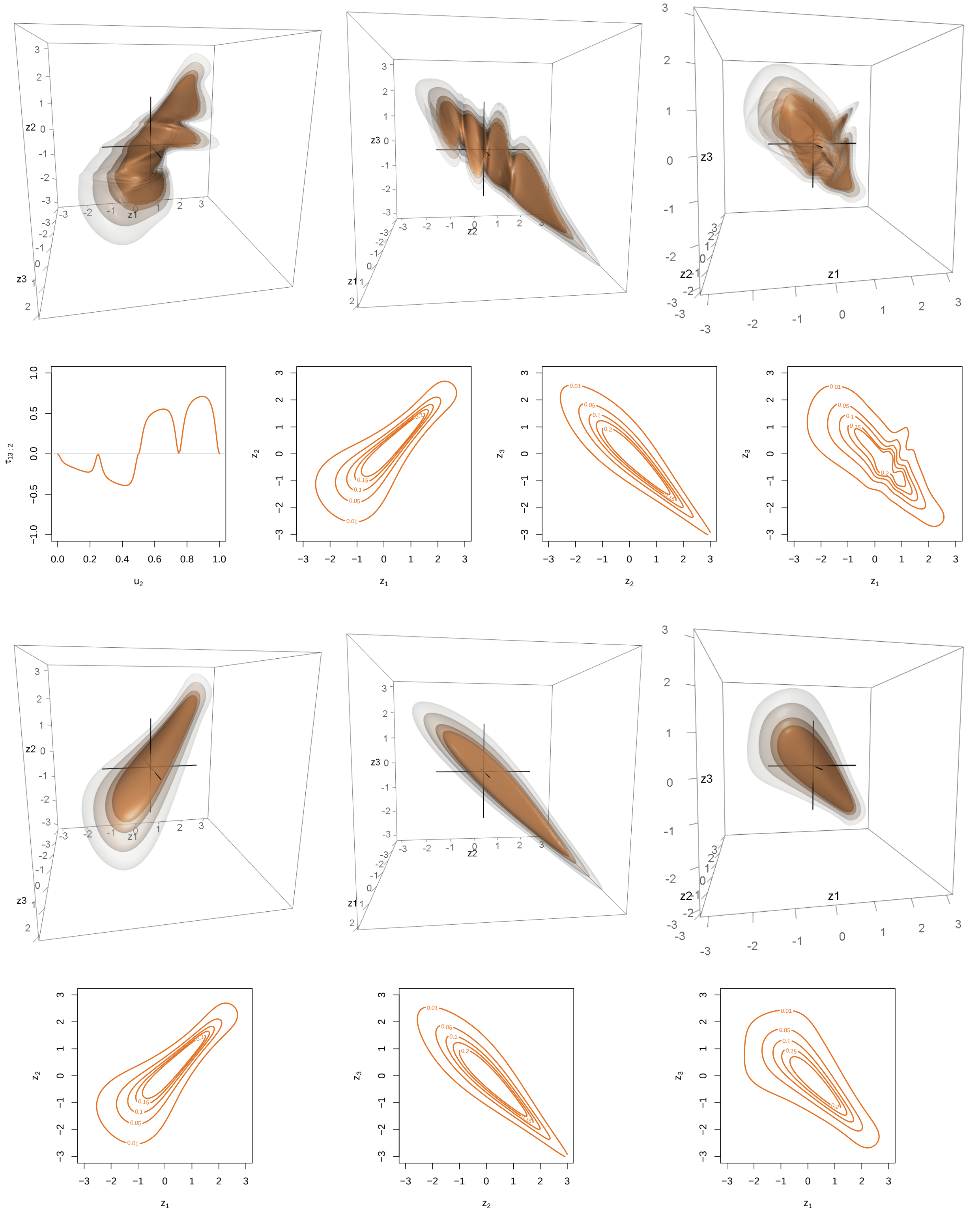

Figure 16: Top row: Contours of the three-dimensional non-simplified vine copula density specified by c_{12}: BB8$(6, 0.95)$, c_{23}: $\mathcal{G}^{270}(-3.5)$, $c_{13;2}$: $\mathcal{T}_{(2)}/\mathcal{T}^{90}_{(2)}(\theta^{(1)}_{13;2}(u_2), \theta^{(2)}_{13;2}(u_2))$ with $\theta^{(1)}_{13;2}(u_2) = \operatorname{sgn}(u_2 - 0.5)(4 - 3\cos(8\pi u_2))$ and $\theta^{(2)}_{13;2}(u_2) = 0.1 + 0.8u_2$. Second row: $\tau_{13;2}$ depending on u_2 and contours of the bivariate margins corresponding to the specification of the top row. Third row: Contours of the three-dimensional simplified vine copula approximation specified by c_{12}: BB8$(6, 0.95)$, c_{23}: $\mathcal{G}^{270}(-3.5)$, $\hat{c}_{13;2}$: t$(0.18, 2.6)$. Bottom row: Contours of the bivariate margins corresponding to the specification of the third row.

are clearly visible as bumps in the three-dimensional contour plots. Of course, one can argue about how realistic it is to assume that real data follows such a distribution but it illustrates the variety of densities which can be modeled using non-simplified parametric vine copulas.

For this scenario we can state that again the bivariate marginal contours do not really anticipate the complex shape of the corresponding three-dimensional object. The contour plots of c_{12} and c_{23} in the second row of Figure 16 look perfectly smooth and regular and do not at all suggest the extremely twisted and contorted structure which can be seen in the top row of Figure 16. For the conditional copula of the corresponding simplified vine copula approximation a t copula with $\hat{\rho}_{13;2} = 0.18$ and $\hat{\nu}_{13;2} = 2.6$ is fitted, which corresponds to a Kendall's τ of 0.11. Comparing the resulting bivariate margins in the second and last row of Figure 16 we see that apart from a little bump of c_{13} of the non-simplified vine copula their general shapes are fairly similar. However the three-dimensional contour plot reveals that the simplified vine copula approximation (third row of Figure 16) is completely smooth with no twists and dents at all, such that it is not able to capture all aspects of the actual interdependencies.

4.1.4 Application to simulated and real data

In this section we want to investigate how the contour plots can help to decide whether a simplified or a non-simplified specification for given data is needed. For this purpose we at first consider simulated data, where we know the true underlying distribution, and afterwards apply the method to real data.

Simulation study

For the simulated data example we specify the true non-simplified vine copula model as follows: We choose c_{12} to be a Gumbel copula with parameter $\theta_{12} = 1.5$ ($\tau_{12} = 0.33$), c_{23} as a t copula with $\rho_{23} = 0$ and 2.5 degrees of freedom ($\tau_{23} = 0$) and $c_{13;2}$ as a Frank copula with parameter function $\theta_{13;2}(u_2) = 3\arctan(10(u_2 - 0.5))$, implying negative dependence for $u_2 < 0.5$ and positive dependence for $u_2 > 0.5$ with absolute τ values smaller than 0.4 (compare Figure 18, top left panel). The rather low pairwise dependencies of this copula are clearly visible from the density's contour plots in the top row of Figure 17. However the surfaces look quite crumpled with lots of irregular bumps and deformations. Moreover it is eye-catching that in this example we only observe three contour surfaces. The inner surface is missing since the density only take values between 0 and 0.101 but the level of the inner surface is 0.11. Further due to the low dependence we cannot detect any corner with extraordinarily high probability mass.

We generated a sample of size $N = 3{,}000$ from this model and transformed the margins to be standard normal in order to make results comparable. For this transformed data sample, we performed a standard kernel density estimation with the function `kde` from the R package `ks` (Duong, 2016a) using Gaussian kernels. Note that using this method we only get approximately standard normal margins. The contours of the resulting estimated densities, which are shown in the second row of Figure 17, are very close to those of the true underlying density in the top row. Only the innermost contour surface is smaller because the peaks of the density tend to get averaged out by kernel density estimation. The second row of Figure 18 displays the contours of the corresponding kernel density estimated bivariate margins, which are again close to the true ones in the first row of Figure 18.

The idea is now to compare these contour plots to those of estimated simplified and non-simplified vine copula densities. We use `RVineStructureSelect` (from `VineCopula`) to fit

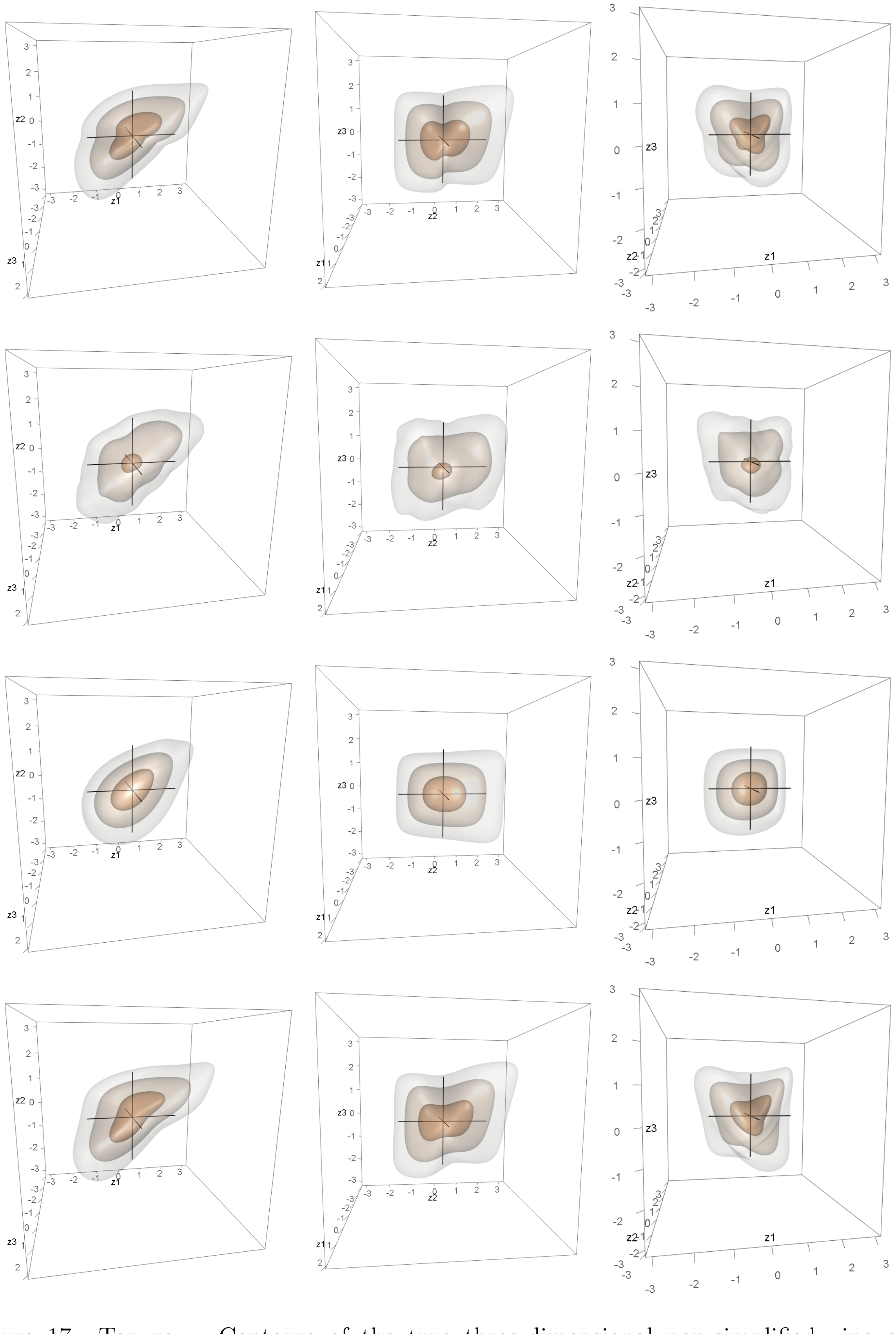

Figure 17: Top row: Contours of the true three-dimensional non-simplified vine copula density specified by c_{12}: $\mathcal{G}(1.5)$, c_{23}: $\mathrm{t}(0, 2.5)$, $c_{13;2}$: $\mathcal{F}(\theta_{13;2}(u_2))$ with $\theta_{13;2}(u_2) = 3\arctan(10(u_2 - 0.5))$. Second row: Contours of the density estimated via three-dimensional kernel density estimation. Third row: Contours of the fitted simplified vine copula density specified by $\hat{c}_{12}$: $\mathcal{G}(1.49)$, $\hat{c}_{23}$: $\mathrm{t}(0.04, 2.36)$, $\hat{c}_{13;2}$: $\mathrm{t}(0, 9.14)$. Bottom row: Contours of the fitted non-simplified vine copula density specified by $\hat{c}_{12}$: $\mathcal{G}(1.49)$, $\hat{c}_{23}$: $\mathrm{t}(0.04, 2.36)$, $\hat{c}_{13;2}$: $\mathcal{N}(\hat{\rho}_{13;2}(u_2))$.

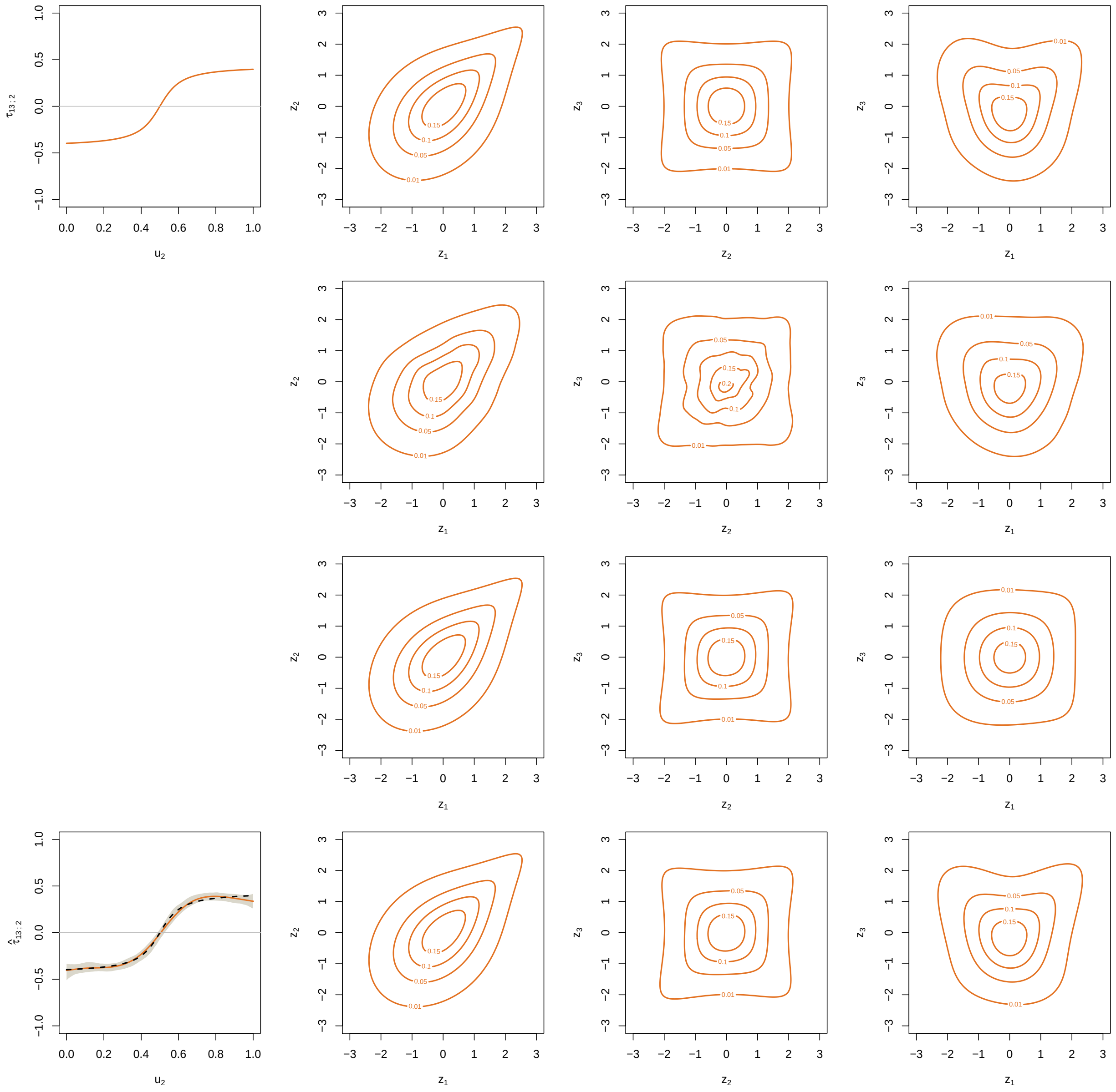

Figure 18: $\tau_{13;2}$ depending on u_2 (first column) and contour plots of the bivariate margins c_{12} (second column), c_{23} (third column) and c_{13} (fourth column). Top row: true vine copula. Second row: kernel density estimation. Third row: fitted simplified vine copula. Bottom row: fitted non-simplified vine copula.

a simplified vine copula and `gamVineStructureSelect` (from `gamCopula`, Vatter, 2016) to fit a non-simplified vine copula. Both algorithms estimate the same unconditional copulas: c_{12} is fitted as a Gumbel copula with parameter $\hat{\theta}_{12} = 1.49$ and c_{23} as a t copula with $\hat{\rho}_{23} = 0.04$ and $\hat{\nu}_{23} = 2.36$ degrees of freedom. In the simplified setting $c_{13;2}$ is estimated to be a t copula with $\hat{\rho}_{23} = -0.01$ and $\hat{\nu}_{23} = 3.42$. The corresponding contours are shown in the third row of Figure 17. They seem to be an over-smoothed version of the kernel estimated density contours. While the general strengths of dependence are represented fairly well, the simplified vine copula approximation does not feature the bumps and dents of the kernel density estimated surfaces. A look at the contours of the bivariate margins in the third row of Figure 18 reveals that the densities of the explicitly modelled margins c_{12} and c_{23} are fitted very well. The true copula families are chosen and the estimated parameters are close to the true values. However the contours of the implicitly defined margin c_{13} are far from the ones of the kernel density estimate. This is another indicator for an insufficient fit resulting from the underlying simplifying assumption, which is in this case too restrictive.

We now investigate whether these deficiencies can be remedied by fitting a non-simplified vine copula to the simulated data. The algorithm `gamVineStructureSelect` estimates the copula $c_{13;2}$ to be Gaussian with parameter function $\hat{\rho}_{13;2}(u_2)$ depending on u_2 via the functional form displayed in the bottom left panel of Figure 18 (in terms of $\hat{\tau}_{13;2}$), together with its bootstrapped 95%-confidence intervals (gray) and the true $\tau_{13;2}$ curve (dashed line). The τ-values range between -0.36 and 0.38 with negative values for $u_2 < 0.5$ such that the estimated function is quite close to the true underlying τ-function. Even though the wrong copula family is chosen for $c_{13;2}$ (Gaussian instead of Frank) the contours of the resulting non-simplified vine copula in Figure 17 (bottom row) are very similar to the kernel estimated ones and fit their shape considerably better than the estimated simplified vine copula. Also the contours of the bivariate margin c_{13} in the bottom right panel of Figure 18 now provide a much better fit. Hence we can conclude that in this example we are able to visually detect the violation of the simplifying assumption of the true distribution.

Real data application

In the following section we want to apply this method to a real data example. We investigate the well-known `uranium` data set, which can be found in the R package `copula`. This data set consists of 655 chemical analyses from water samples from a river near Grand Junction, Colorado (USA). It contains the log-concentration of seven chemicals, where we will focus on the three elements cobalt (X_1), titanium (X_2) and scandium (X_3) that have already been examined regarding the simplifying assumption in Acar et al. (2012). In order to obtain copula data we first apply the probability integral transform to the data using the empirical marginal distribution functions, i.e. the observations x_{ji}, $j = 1, 2, 3$, $i = 1, \dots, N$, are transformed via the rank transformation

$$u_{ji} = \frac{1}{N+1} \sum_{k=1}^{N} 1_{\{x_{jk} \leq x_{ji}\}},$$

where $1_{\{\cdot\}}$ is the indicator function. Then we transform the data to have standard normal margins in accordance to the previous examples.

We now want to take a look at the “true” model and perform a kernel density estimation. In the top rows of Figure 19 and Figure 20 the results of the three- and two-dimensional kernel density estimations are displayed, respectively. The three variables seem to be positively dependent. A few bumps and dents are noticeable. Next we explore how well estimated simplified and non-simplified vine copulas fit the data.

Using `RVineStructureSelect` we obtain the following simplified vine copula: c_{12} is a t copula with $\hat{\rho}_{12} = 0.74$ and $\hat{\nu}_{12} = 8.03$ ($\hat{\tau}_{12} = 0.53$), c_{23} is a t copula with $\hat{\rho}_{23} = 0.63$ and $\hat{\nu}_{23} = 5.93$ ($\hat{\tau}_{23} = 0.43$) and $c_{13;2}$ is a t copula with $\hat{\rho}_{13;2} = 0.08$ and $\hat{\nu}_{13;2} = 5.65$ ($\hat{\tau}_{13;2} = 0.05$). This t vine copula and its bivariate margins are depicted in Figure 19 (middle row) and Figure 20 (middle row), respectively. Since all three degrees of freedom are of medium size we observe modest lower and upper tail dependence. Again these contours resemble a smoothed version of the slightly bumpy kernel density estimated contour surfaces resulting in a rather unsatisfying fit of the data.

For the non-simplified vine copula, the estimates of c_{12} and c_{23} are the same as for the simplified one. The third pair-copula $c_{13;2}$ is still a t copula but now with $\hat{\nu}_{13;2} = 6.69$ degrees of freedom and an association parameter depending on u_2. We show the relationship

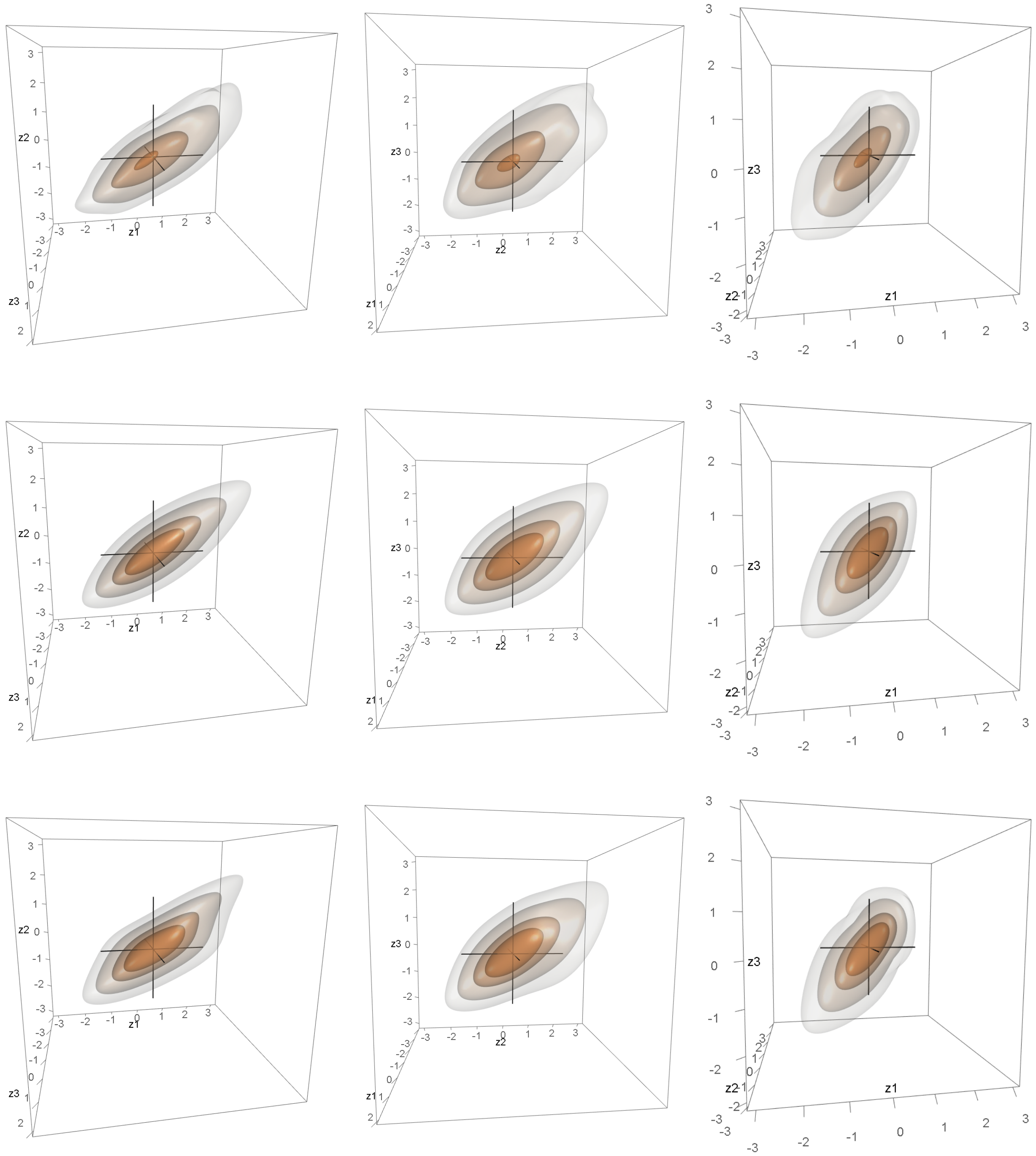

Figure 19: Top row: Contours of the density estimated via three-dimensional kernel density estimation. Middle row: Contours of the simplified vine copula specified by $\hat{c}_{12}$: $\mathrm{t}(0.53, 8.03)$, $\hat{c}_{23}$: $\mathrm{t}(0.43, 5.93)$, $\hat{c}_{13;2}$: $\mathrm{t}(0.08, 5.65)$. Bottom row: Contours of the non-simplified vine copula specified by $\hat{c}_{12}$: $\mathrm{t}(0.53, 8.03)$, $\hat{c}_{23}$: $\mathrm{t}(0.43, 5.93)$, $\hat{c}_{13;2}$: $\mathrm{t}(\hat{\rho}_{13;2}(u_2), 6.69)$.

between u_2 and $\hat{\tau}_{13;2}$ in the bottom left panel of Figure 20 (again with its bootstrapped 95%-confidence intervals). One can see that we have small positive values of Kendall's τ for $u_2 \leq 0.8$ and negative dependence for the remaining values of u_2. Although only the parameters of the copula $c_{13;2}$ are different compared to the simplified vine copula, the shapes of the contour surfaces display some interesting changes: Especially in the bottom left and right panel of Figure 19, we see that the smooth diamond-shaped contours from

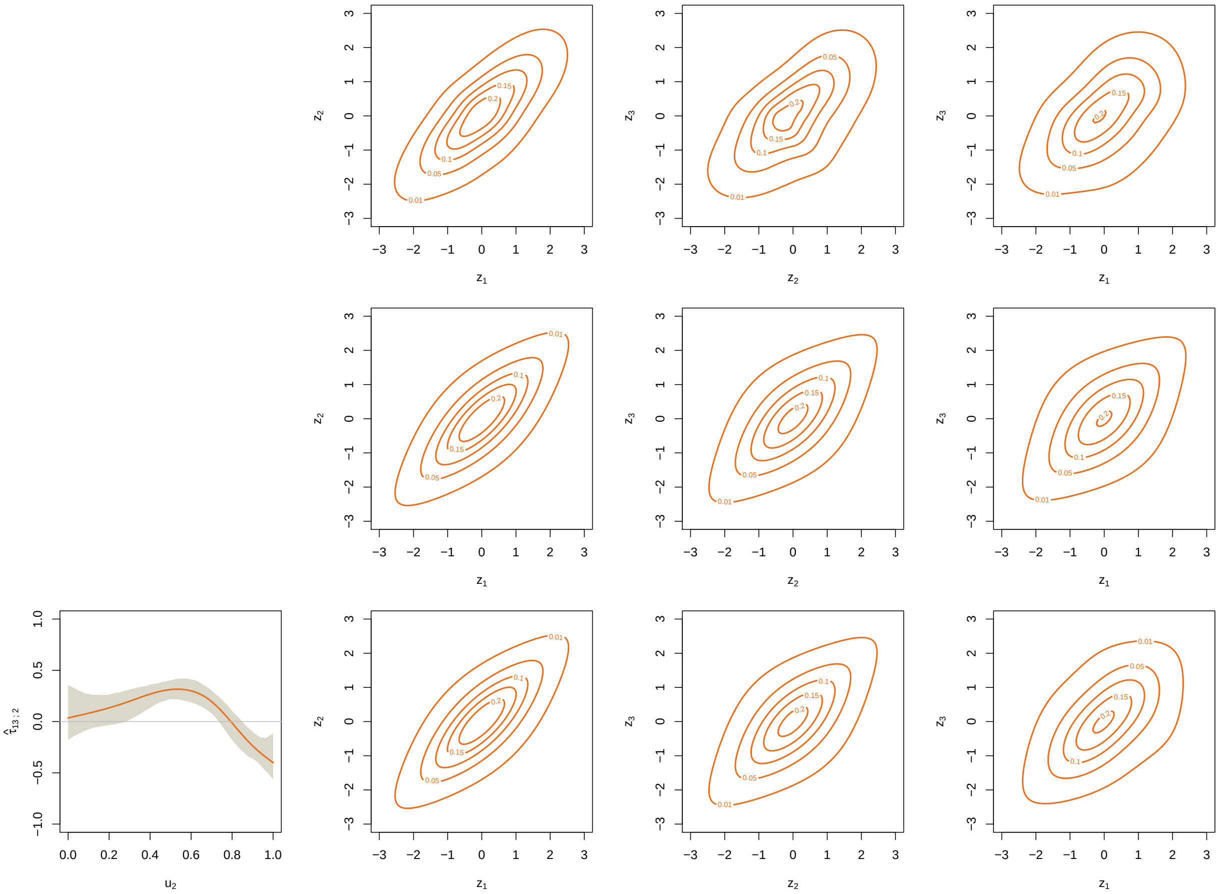

Figure 20: $\hat{\tau}_{13;2}$ depending on u_2 (bottom left) and contour plots of the bivariate margins c_{12} (second column), c_{23} (third column) and c_{13} (fourth column). Top row: kernel density estimation. Middle row: estimated simplified vine copula. Bottom row: estimated non-simplified vine copula.

Figure 19 (middle row) have developed several dents. While the contour plots of c_{12} and c_{23} are the same as before, the one of c_{13} exhibits some differences since it is no longer diamond-shaped.
Comparing these contours to the ones from the top rows of Figure 19 and Figure 20 we see that the non-simplified vine copula is able to capture the behavior of the data quite well. The most noticeable bumps and dents are reproduced and the bivariate contours resemble the kernel density estimated ones. Thus we come to the same conclusion as Acar et al. (2012), namely that the vine copula decomposition of this three-dimensional data set is of the non-simplified form.

4.1.5 Summary

In Section 4.1 we looked at the contour surfaces of several three-dimensional simplified and non-simplified vine copulas. The flexibility of simplified vine copulas in comparison to standard elliptical and Archimedean copulas was demonstrated. Using the 12 different one- and two-parametric bivariate pair-copula families currently implemented in `VineCopula` for the construction of a simplified vine copula, the shape of the resulting contour surfaces may deviate considerably from the well-known ellipsoid-shaped contours of a Gaussian distribution. Considering non-simplified vine copulas facilitates the modeling of even more irregular contour shapes exhibiting twists, bumps and altering dependence patterns. In our examples we have observed that contemplating three-dimensional contour surfaces gives more insight into the trivariate dependence structure than only looking at the two-dimensional marginal contour lines. While the consideration of the three bivariate marginal contour plots already gives a good impression of the shape of the three-dimensional object for simplified vine copulas, one might be surprised how twisted and contorted some non-simplified three-dimensional densities appear if one had only seen the smooth bivariate contour plots. In simulated and real data applications we have seen that non-simplified vine copulas are able to fit data with complex dependencies very well. However, we have observed that the estimated simplified vine copulas still capture the main features of the data providing a more smooth fit. Thus, for practical applications, especially in higher dimensions (when the number as well as the dimension of the parameter functions increase, causing numerical intractability) it might be preferable to use simplified vine copulas. Thereby overfitting might be avoided while the main properties of the data such as correlations and tail behavior are still well represented.

4.2 Testing the simplifying assumption using model distances

Parts of Section 4.2 are very similar to the publication Killiches, Kraus, and Czado (2017c).

4.2.1 Introduction

In model selection, the distance between statistical models plays a big role. Often the difference between two models is measured in terms of the Kullback-Leibler (KL) distance (Kullback and Leibler, 1951). Nikoloulopoulos and Karlis (2008a) were among the first to use the KL distance for model selection for copulas. Further, Joe (2014) used the KL distance to calculate the sample size necessary to discriminate between two copula densities. In the context of the simplifying assumption for vine copulas 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) assessed the strength of non-simplifiedness of the trivariate Farlie-Gumbel-Morgenstern (FGM) copula for different dependence parameters.
Nevertheless, the main issue of the Kullback-Leibler distance is that, as soon as it cannot be computed analytically, a numerical evaluation of the appearing integral is needed, which is hardly tractable once the model dimension exceeds three or four. In order to tackle this problem, several modifications of the Kullback-Leibler distance have been proposed in Killiches, Kraus, and Czado (2017b). They yield model distances which are close in performance to the classical KL distance, however with much faster computation times, facilitating their use in high dimensions. Using these model distances, we will investigate a major question arising when working with vines: Is the simplifying assumption justified for a given data set or do we need to account for non-simplifiedness? The importance of this topic can be seen from many recent publications such as Hobæk Haff et al. (2010), Stöber et al. (2013), Acar et al. (2012) or Spanhel and Kurz (2015).

4.2.2 Model distances for vine copulas

In this section we shortly review the definitions of the most important distance measures discussed in Killiches, Kraus, and Czado (2017b). For detailed information about the concepts consult this paper and references therein. Starting point is the so-called *Kullback-Leibler distance* (see Kullback and Leibler, 1951) between two d-dimensional copula densities c^f, $c^g\colon [0,1]^d \to [0,\infty)$, defined as

$$\mathrm{KL}(c^f, c^g) = \int_{\mathbf{u}\in[0,1]^d} \log\left(\frac{c^f(\mathbf{u})}{c^g(\mathbf{u})}\right) c^f(\mathbf{u})\,\mathrm{d}\mathbf{u}. \tag{4.9}$$

Note that due to the lack of symmetry the KL distance is not a distance in the classical sense and therefore is also referred to as *Kullback-Leibler divergence*. If c^f and c^g are the corresponding copula densities of two d-dimensional densities f and g, it can be easily shown that the KL distance between c^f and c^g is equal to the one between f and g if their marginal distributions coincide. It is common practice to use the Inference Functions for Margins (IFM) method: First the univariate margins are estimated and observations are transformed to the copula scale using the estimated margins; afterwards the copula is estimated based on the transformed data (cf. Joe, 1997, Section 10.1). Therefore, it can be justified that in the remainder of this section we restrict ourselves to data on the copula scale.
Since in the vast majority of cases the KL distance cannot be calculated analytically, the main problem of using the KL distance in practice is the computational intractability for

dimensions larger than 4. There, numerical integration suffers from the curse of dimensionality and thus becomes exceptionally inefficient. As a remedy for this issue, Proposition 2 from Killiches, Kraus, and Czado (2017b) expresses the KL between multivariate densities in terms of the sum of expected KL distances between univariate conditional densities:

$$\mathrm{KL}\left(c^f, c^g\right) = \sum_{j=1}^{d} \mathbb{E}_{c^f_{(j+1):d}}\left[\mathrm{KL}\left(c^f_{j|(j+1):d}\left(\,\cdot\,|\mathbf{U}_{(j+1):d}\right), c^g_{j|(j+1):d}\left(\,\cdot\,|\mathbf{U}_{(j+1):d}\right)\right)\right], \tag{4.10}$$

where for $j < d$ we use the abbreviation $(j+1)\!:\!d = \{j+1, j+2, \ldots, d\}$ with $(d+1)\!:\!d := \emptyset$ and $\mathbf{U}_{(j+1):d} \sim c^f_{(j+1):d}$. It would be a valid approach to approximate the expectations in Equation (4.10) by Monte Carlo integration, i.e. the average over evaluations of the integrand on a grid of points simulated according to $c^f_{(j+1):d}$. Since this would also be computationally challenging in higher dimensions and additionally has the disadvantage of being random, Killiches, Kraus, and Czado (2017b) propose to approximate the expectations through evaluations on a grid consisting of only warped diagonals in the respective unit (hyper)cube. The resulting *diagonal Kullback-Leibler* (dKL) distance between two d-dimensional R-vine models $\mathcal{R}^f$ and $\mathcal{R}^g$ is hence defined by

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

where the set of warped discrete diagonals $\mathcal{D}_j \in [0,1]^{d-j}$ is given by

$$\mathcal{D}_j = T_j\left(\left\{\{\mathbf{r} + \mu\mathbf{v}(\mathbf{r}) \mid \mu \in \mathcal{I}_{\varepsilon,n}\} \,\middle|\, \mathbf{r} \in \{0,1\}^{d-j}\right\}\right).$$

Here, $\mathbf{r} \in \{0,1\}^{d-j}$ are the corner points in the unit hypercube $[0,1]^{d-j}$, $\mathbf{v}\colon \{0,1\}^{d-j} \to \{-1,1\}^{d-j}$, $\mathbf{v}(\mathbf{r}) = \mathbf{1} - 2\mathbf{r}$ denotes the direction vector from $\mathbf{r}$ to its opposite corner point and $\mathcal{I}_{\varepsilon,n}$ is the equidistantly discretized interval $[\varepsilon, 1-\varepsilon]$ of length n. Hence $\{\mathbf{r} + \mu\mathbf{v}(\mathbf{r}) \mid \mu \in \mathcal{I}_{\varepsilon,n}\}$ represents a discretization of the diagonal from $\mathbf{r}$ to its opposite corner point $\mathbf{r} + \mathbf{v}(\mathbf{r})$. Finally, these discretized diagonals are transformed using is the inverse Rosenblatt transformation T_j with respect to $c^f_{(j+1):d}$ (Rosenblatt, 1952). Recall that the Rosenblatt transformation $\mathbf{u}$ of a vector $\mathbf{w} \in [0,1]^d$ with respect to a distribution function C is defined by $u_d = w_d$, $u_{d-1} = C^{-1}_{d-1|d}(w_{d-1}|u_d)$, $\ldots$, $u_1 = C^{-1}_{1|2:d}(w_1|u_2, \ldots, u_d)$. Often it is used to transform a uniform sample on $[0,1]^d$ to a sample from C. The concept is used to transform the unit hypercube's diagonal points to points with high density values of $c^f_{(j+1):d}$. Hence, the KL distance between c^f and c^g is approximated by evaluating the KL distances between the univariate conditional densities $c^f_{j|(j+1):d}$ and $c^g_{j|(j+1):d}$ conditioned on values lying on warped diagonals $\mathcal{D}_j$, $j = 1, \ldots, d-1$. Diagonals have the advantage that all components take values on the whole range from 0 to 1 covering especially the tails, where the substantial differences between copula models occur most often. With the above modifications the intractability of the KL for multivariate densities is overcome since only KL distances between univariate densities have to be evaluated. It was shown in Proposition 1 of Killiches, Kraus, and Czado (2017b) that these univariate conditional densities $c^f_{j|(j+1):d}$ and $c^g_{j|(j+1):d}$ can be easily derived for the vine copula model. Moreover, in Remark 1 they prove that for $\varepsilon \to 0$ and $n \to \infty$ the dKL converges to a sum of scaled line integrals. Further, they found heuristically that even for $n = 10$ and $\varepsilon = 0.025$ the dKL was a good and fast substitute for the KL distance.

4.2.3 Testing simplified versus non-simplified vine copulas

As already mentioned in the introduction, the validity of the simplifying assumption is a frequently discussed topic in the recent literature. For the case the simplifying assumption is not satisfied, Vatter and Nagler (2016) developed a method to fit a non-simplified vine to given data such that the parameters of the pair-copulas with non-empty conditioning sets are dependent on the conditioning variable(s). This functional relationship is modeled with a generalized additive model. The fitting algorithm is implemented in the R package `gamCopula` (Vatter, 2016) as the function `gamVineStructureSelect`. The selection of the vine structure is identical to the one selected by `RVineStructureSelect`.
In this section we will present how distance measures can be used to decide whether a (more complicated) non-simplified model is needed or the simplified model suffices. This can be done with the help of the parametric bootstrapping based model selection test introduced in Section 2.3 of Killiches, Kraus, and Czado (2017c). Let $\mathbb{C}^f$ and $\mathbb{C}^g$ be the class of simplified and non-simplified vine copula models, respectively. Since every simplified vine can be represented as a non-simplified vine with constant parameters, $\mathbb{C}^f$ and $\mathbb{C}^g$ are nested, i.e. $\mathbb{C}^f \subseteq \mathbb{C}^g$. Now, for data from an arbitrary, but unknown true distribution $\mathcal{R}^g \in \mathbb{C}^g$ we want to decide which of the classes to choose for modeling the data. Hence, the null hypothesis we want to test at significance level α is

$$H_0: \ \mathcal{R}^g \in \mathbb{C}^f.$$

This would mean that the true underlying model is in $\mathbb{C}^f$, or in other words that the model is simplified. If the null hypothesis was rejected, this would be a clear indicator that a non-simplified vine copula is needed to model the data. To test the null hypothesis (given a copula data set $\mathbf{u}_i^0 \in [0,1]^d$, $i = 1, \ldots, N$, from $\mathcal{R}^g$) we fit a simplified model $\hat{\mathcal{R}}_0^f$ and a non-simplified model $\hat{\mathcal{R}}_0^g$ to the data. Then, the test statistic is given by

$$d_0 = \mathrm{dKL}(\hat{\mathcal{R}}_0^f, \hat{\mathcal{R}}_0^g)$$

and the question is whether this distance is significantly different from 0 (since this would be the theoretical value under H_0). To answer this question the test statistic's distribution is approximated by a parametric bootstrapping scheme, which is described in detail in Section 2.3 of Killiches, Kraus, and Czado (2017c). In short, we generate $M = 100$ bootstrap replications $d_1, \ldots, d_M$ of the test statistic under H_0 and estimate the p-value as $\frac{1}{M}\sum_{i=1}^{M} \mathbb{1}_{d_i > d_0}$. The validity of this approach is also discussed in detail in Killiches, Kraus, and Czado (2017c). After investigating the power of the test in the following paragraph, we apply it to a hydro-geochemical and a financial data set in Section 4.2.4.

Power of the test

In a simulation study we investigate the performance of our test. For this purpose we consider a three-dimensional non-simplified vine consisting of the pair-copulas $c_{1,2}$, $c_{2,3}$ and $c_{1,3;2}$, where all pairs are bivariate Clayton copulas. The Kendall's τ values of the copulas $c_{1,2}$ and $c_{2,3}$ are $\tau_{1,2} = 0.7$ and $\tau_{2,3} = 0.5$, respectively. The third τ value depends linearly on u_2: $\tau_{1,3;2}(u_2) = a + (b-a)u_2$ with constants $a, b \in [-1, 1]$. For $a = b$ the function is constant such that the vine is simplified. By construction $\tau_{1,3;2}$ can become negative for some combinations of a, b and u_2; in such cases we use the 90 degree rotated version of the Clayton copula since the Clayton copula does not allow for negative dependence.
By fixing $a = 0.3$ and letting b range between -1 and 1 in 0.1 steps we obtain 21 scenarios. For each of the scenarios we generate a sample of size $N \in \{200, 500, 1000\}$ from the corresponding non-simplified vine copula and fit both a simplified and a non-simplified

model to the generated data. Since we are only interested in the parameters and their variability we fix both the vine structure and the pair-copula families to the true ones. We test the null hypothesis that the two underlying models are equal. In order to assess the power of the test, we perform this procedure $P = 250$ times (at significance level $\alpha = 5\%$) and check how many times the null hypothesis is rejected. As sample size we take the same N used to generate the original data. In each test we perform $M = 100$ bootstrap replications. In Figure 21 the proportions of rejections of the null hypothesis within the $P = 250$ performed tests are shown depending on b. The different sample sizes are indicated by the three different curves: $N = 200$ (dotted curve), $N = 500$ (dashed curve) and $N = 1000$ (solid curve).

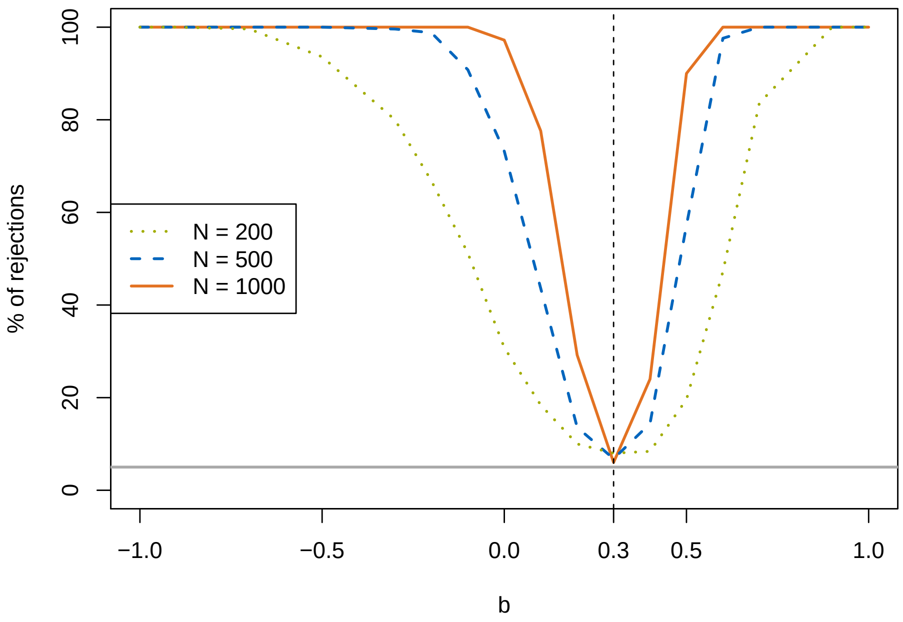

Figure 21: Percentage of rejections of H_0 (at level $\alpha = 5\%$) depending on b with constants $a = 0.3$, $M = 100$ and $P = 250$ for $N = 200$ (dotted curve), $N = 500$ (dashed curve) and $N = 1000$ (solid curve). The horizontal solid gray line indicates the 5% level.

We see that the observed power of the test is in general very high. Considering the dashed curve, corresponding to a sample size of $N = 500$, one can see the following: If the distance $|b - a|$ is large, $\tau_{1,3;2}$ is far from being constant. Hence, we expect the non-simplified vine and the simplified vine to be very different and therefore the power of the test to be large. We see that for $b \leq -0.1$ and $b \geq 0.6$ the power of the test is above 80% and for $b \leq -0.2$ and $b \geq 0.7$ it is even (close to) 100%. For values of b closer to a the power decreases. For example, for $b = 0.1$ the Kendall's τ value $\tau_{1,3;2}$ only ranges between 0.1 and 0.3 implying that the non-simplified vine does not differ too much from a simplified vine. Therefore, we cannot expect the test to always detect this difference. Nevertheless, even in this case the power of the test is estimated to be almost 44%. From a practical point of view, this result is in fact desirable since models estimated based on real data will always exhibit at least slight non-simplifiedness due to randomness even when the simplifying assumption is actually satisfied. Further, for $b = 0.3$ the function $\tau_{1,3;2}(u_2)$ is actually constant with respect to u_2 so that $\mathcal{R}^*$ is a simplified vine. Thus, H_0 is true and we hope to be close to the significance level $\alpha = 5\%$. With 6.4% of rejections, we see that this is the case here.
Looking at the dotted and the solid curve we find that the higher the sample size is, the higher is the power of the test, which is what one also would have expected. In the case of

$N = 1000$, we have a power of over 80% for $b \in [-1, 0] \cup [0.5, 1]$ and even 100% rejections for $b \in [-1, -0.1] \cup [0.6, 1]$. For $b = 0.3$ the test holds its level with 5.2% of rejections. Yet even for a sample size of as little as $N = 200$, the power of the test is above 80% for values of b between -1 and -0.3 and 0.5 and 1. A power of 100% is reached for $b \leq -0.8$ and $b \geq 0.9$. For $b = 0.3$ the test rejects the null hypothesis in 7.2% of the cases.

We can conclude that our test is a valid α-level method in finite samples to decide if a non-simplified model is necessary.

4.2.4 Real data examples

Three-dimensional subset of the uranium data set

To show an application of our test we revisit the three-dimensional subset of the `uranium` data discussed in Section 4.1.4 and fit both a simplified and a non-simplified vine copula to the data.

The fitted simplified vine $\hat{\mathcal{R}}_0^f$ is specified in the following way: $c_{1,2}$ is a t copula with $\tau_{1,2} = 0.53$ and $\nu_{1,2} = 8.03$, $c_{2,3}$ is a t copula with $\tau_{2,3} = 0.43$ and $\nu_{2,3} = 5.93$ and $c_{1,3;2}$ is a t copula with $\tau_{1,3;2} = 0.08$ and $\nu_{1,3;2} = 5.65$. For the non-simplified vine $\hat{\mathcal{R}}_0^g$, the pair-copulas $c_{1,2}$ and $c_{2,3}$ are the same as for the simplified vine, $c_{1,3;2}$ is also still a t copula but now has $\nu_{1,3;2} = 6.69$ degrees of freedom and its association parameter depends on u_2 as displayed as the solid line in Figure 22. For values of u_2 below 0.8 (roughly) we have small positive Kendall's τ values, whereas for the remaining values we observe small to medium negative association. For comparison, the (constant) Kendall's τ of the estimated simplified vine is plotted as a dashed line. Further, the pointwise bootstrapped 95% confidence bounds under H_0 are indicated by the gray area.

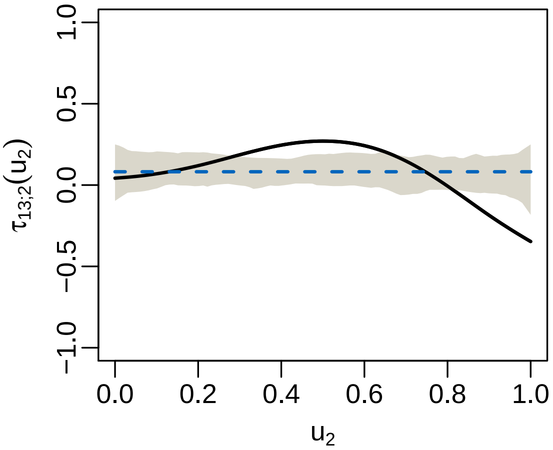

Figure 22: Estimated functional relationship between $\tau_{1,3;2}$ and u_2 (for the non-simplified model $\hat{\mathcal{R}}_0^g$). The dashed line represents the constant $\tau_{1,3;2}$ of the simplified model $\hat{\mathcal{R}}_0^f$ and the gray area indicates the pointwise bootstrapped 95% confidence bounds under H_0.

The fact that the estimated Kendall's τ function exceeds these bounds for more than half of the u_2 values suggests that the simplified and non-simplified vines are significantly different from each other. We now use our testing procedure to formally test this.

The distance between the two vines is $\mathrm{dKL}_0 = 0.058$. Using our test, we can reject H_0 at the 5% level (with a bootstrapped p-value of 0.01 based on 100 bootstrap iterations). Hence, we conclude that here it is necessary to model the dependence structure between the three variables using a non-simplified vine. Acar et al. (2012) and our findings in Section 4.1.4 also suggest that a simplified vine would not be sufficient in this example.

Four-dimensional subset of the EuroStoxx 50 data set

We examine a 52-dimensional EuroStoxx50 data set containing 985 observations of returns of the EuroStoxx50 index, five national indices and the stocks of the 46 companies that were in the EuroStoxx50 for the whole observation period (May 22, 2006 to April 29, 2010). Since fitting non-simplified vine copulas in high dimensions would be too computationally demanding we consider only a four-dimensional subset containing the following national indices: the German DAX (U_1), the Italian MIB (U_2), the Dutch AEX (U_3) and the Spain IBEX (U_4) (see also Example 1 of Killiches, Kraus, and Czado (2017b), where this data set was already investigated). In practice it is very common to model financial returns using t copulas (see e.g. Demarta and McNeil, 2005). From Stöber et al. (2013) we know that any t copula can be represented as a vine satisfying the simplifying assumption. With our test we can check whether this necessary condition is indeed fulfilled for this particular financial return data set.

We proceed as in the previous section and fit a simplified model $\hat{\mathcal{R}}_0^f$ as well as a non-simplified model $\hat{\mathcal{R}}_0^g$ to the data. The estimated structures of both models are C-vines with root nodes DAX, MIB, AEX and IBEX. Again, the pair-copulas in the first tree coincide for both models being fitted as bivariate t copulas with $\tau_{1,2} = 0.70$ and $\nu_{1,2} = 4.96$, $\tau_{1,3} = 0.72$ and $\nu_{1,3} = 6.23$, and $\tau_{1,4} = 0.69$ and $\nu_{1,2} = 6.80$. In the second tree of the simplified model the pair-copulas are also estimated to be t copulas with $\tau_{2,3;1} = 0.23$ and $\nu_{1,2} = 6.34$, and $\tau_{2,4;1} = 0.24$ and $\nu_{1,2} = 10.77$. The corresponding non-simplified counterparts fitted by the `gamVineStructureSelect` algorithm are also t copulas, whose strength of dependence varies only very little and stays within the confidence bounds of the simplified vine (see Figure 23, left and middle panel). The estimated degrees of freedom are also quite close to the simplified ones ($\nu_{2,3;1} = 6.47$ and $\nu_{2,4;1} = 11.56$), such that regarding the second tree we would presume that the distance between both models is negligible. Considering the copula $c_{3,4;1,2}$ in the third tree, the simplified fit is a Frank copula with $\tau_{3,4;1,2} = 0.11$, while the non-simplified fit is a Gaussian copula whose τ values only depend on u_1 (i.e. the value of the DAX). In the right panel of Figure 23 we see the estimated relationship, which is a bit more varying than the others but still mostly stays in between the confidence bounds. The broader confidence bounds can be explained by the increased parameter uncertainty for higher order trees of vine copulas arising due to the sequential fitting procedure.

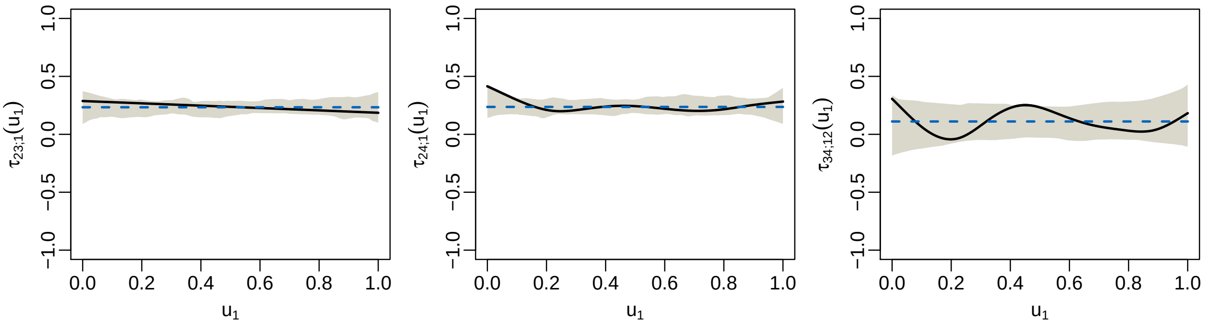

Figure 23: Estimated functional relationship of $\tau_{2,3;1}$ (left), $\tau_{2,4;1}$ (middle) and $\tau_{3,4;1,2}$ (right) in terms of u_1 from $\hat{\mathcal{R}}_0^g$. The dashed lines represent the constant τ values of the simplified model $\hat{\mathcal{R}}_0^f$ and the gray areas indicate the pointwise bootstrapped 95% confidence bounds under H_0.

The question is now, whether the estimated non-simplified vine is significantly different from the simplified one, or in other words: Is it necessary to use a non-simplified vine copula model for this data set or does a simplified one suffice? In order to answer this question we make use of our test using parametric bootstrapping and find out that with a p-value of 0.24 (based on 100 bootstrap iterations) the null hypothesis cannot be rejected. So we can conclude that for this four-dimensional financial return data set a simplified vine suffices to reasonably capture the dependence pattern.

Although we only presented applications in dimensions 3 and 4, in general the procedure can be used in arbitrary dimensions. The computationally limiting factor is the fitting routine of the non-simplified vine copula model, which can easily be applied up to 15 dimensions in a reasonable amount of time (for the methods implemented in Vatter, 2016).

4.3 Determination of simplified vine structures using a test for constant conditional correlations

Parts of Section 4.3 are very similar to the publication Kraus and Czado (2017b).

4.3.1 Motivation

Recall that an exemplary vine copula decomposition of a three-dimensional vine copula is given by:

$$c(u_1, u_2, u_3) = c_{12}(u_1, u_2)\, c_{23}(u_2, u_3) c_{13;2}\left(C_{1|2}(u_1|u_2), C_{3|2}(u_3|u_2); u_2\right). \tag{4.11}$$

In general, there are various ways of specifying a vine copula decomposition. For example, in the three-dimensional case, there are three possibilities depending on which variable is chosen to be the conditioning one. When we speak of the structure of the vine, we mean the specification which pairs of variables are modeled conditioned on which other variables. It is now a stylized fact that the flexibility of vines is able to overcome the problem of a limited model choice in the case when only elliptical or Archimedean copulas are used. Being able to assign a different bivariate copula to any pair of variables, the new issue arises of having too many possible models to choose from. Not only the selection of pair-copula families and corresponding parameters, but especially the vast number of possible model structures make the search of an "optimal" vine for a given data set a challenging task. Until now, almost unanimously, the method of choice for structure selection is the so-called Dißmann algorithm (Dißmann et al., 2013). This is also due to its prominent implementation as `RVineStructureSelect` in the R package `VineCopula` (Schepsmeier et al., 2017). Dißmann's algorithm is a heuristic that sequentially constructs the vine's structure, trying to capture most of the dependence in the lower trees.
The motivation of this section is the observation that the model structure of a vine copula has a considerable influence on the validity of the simplifying assumption. To illustrate this, reconsider the three-dimensional vine copula model from Equation (4.11), now in its simplified form:

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

This density can also be written as another vine copula using a different tree structure, e.g. the one containing the pairs (1,2) and (1,3) in the first and (2,3; 1) in the second tree:

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

This decomposition does not need to be of simplified form anymore, i.e. the copula $c_{23;1}$ might now depend on the conditioning value u_1. Hence, given data from the simplified vine copula model we expect a fitted simplified vine copula with the original tree structure to have a significantly better model fit than one with a different tree structure. While Dißmann's structure selection algorithm only takes into account the strength of dependencies between pairs of variables, no consideration concerning the resulting conditional pair-copula is made. In this section, we propose two new structure selection algorithms that try to amend exactly this flaw. By using a test for constant conditional correlations developed in Kurz and Spanhel (2017) and implemented in Kurz (2017), we incorporate information about the violation of the simplifying assumption when constructing the vine's tree structure. The first algorithm selects the first tree similar to Dißmann's algorithm and uses the p-values of these tests as weights in the subsequent trees. The second algorithm we propose fits a C-vine, where each root node is selected in a way that minimizes the occurrence of non-simplifiedness in the next tree. As we will see in a simulation study, this improves the model fit in terms of the Akaike information criterion (AIC) compared to Dißmann most of the time, especially when the dimension is large. Finally, in revisiting several classic data sets that have already been studied in the vine copula context we demonstrate the practical relevance of our proposed methods.

4.3.2 Dißmann's algorithm for selecting simplified vine copulas

Since the sequential structure selection algorithm proposed in Dißmann et al. (2013) will serve as a benchmark vine model selection method, we shortly outline how it works. Assume we are given an i.i.d. sample of size n of a d-dimensional copula, denoted by $(u_j^{(i)})_{j=1,\dots,d}^{i=1,\dots,n}$. Since a vine copula model consists of three parts (the vine structure, the parametric families of each pair-copula and their corresponding copula parameters), each of these components have to be estimated.

Assume at first that the tree structure is given. Then for each of the $d(d-1)/2$ pair-copulas a family and copula parameter(s) have to be selected. Starting with the first tree, for each unconditional pair the maximum-likelihood estimate of the copula parameter is determined for each family studied (see Appendix A.1 for a list of families currently implemented in the `VineCopula` package). Then the family with the lowest Akaike information criterion is chosen (Akaike, 1998). In the higher trees, for each edge corresponding to a conditional pair-copula $c_{j_e,k_e;D_e}$, the estimated pseudo observations $u_{j_e|D_e}^{(i)} := \hat{C}_{j_e|D_e}(u_{j_e}^{(i)}|\mathbf{u}_{D_e}^{(i)})$ and $u_{k_e|D_e}^{(i)} := \hat{C}_{k_e|D_e}(u_{k_e}^{(i)}|\mathbf{u}_{D_e}^{(i)})$, $i = 1, \dots, n$, are calculated and used to find the optimal family and parameter for pair-copula $c_{j_e,k_e;D_e}$. This is repeated until all pair-copulas are fitted. Note that it is possible to incorporate a test for independence in the pair-copula selection process (see Genest and Favre, 2007): Based on the estimated Kendall's τ the null hypothesis that the (pseudo) observations come from the independence copula is tested. Then, if the null hypothesis is not rejected for some chosen level β, the pair-copula to be estimated is chosen to be the independence copula.

Regarding the selection of the tree structure, Dißmann's algorithm uses a heuristic that models the strongest pairwise dependencies (measured in terms of the absolute empirical Kendall's τ value) in the lower trees. To be precise, the algorithm starts with the first tree, where all pair-wise empirical Kendall's τ values are determined and then a maximum spanning tree using the absolute τ values as weights is selected (e.g. by the Algorithm of Prim, see Cormen et al., 2001, Section 23.2). For the next tree, all required pseudo observations are determined as above and for all edges allowed by the proximity condition the Kendall's τ values are estimated. Again, a maximum spanning tree is selected using these empirical Kendall's τ values as edge weights, now on the graph constrained by the proximity condition. This procedure is iterated until all $d-1$ trees are selected.

Dißmann et al. (2013) justify this heuristic by noting that the lowest trees have the greatest influence on the overall fit and thus it is important to model most of the dependence early. Further, they argue that this procedure minimizes estimation errors in higher trees, because typically the dependence decreases when using the algorithm, making rounding errors less severe. As a measure of the strength of dependence Kendall's τ is used since it is a rank-based dependence measure facilitating the comparison of different copula families (see e.g. Nelsen, 2007, Chapter 5.1.1).

Alternative methods using other edge weights for the determination of the maximum spanning tree have been proposed. For example, the fitted pair-copula's AIC or the p-value of a goodness-of-fit test have been used in Czado et al. (2013), but they are not commonly used since all possible pair-copulas have to be estimated first to obtain all the edge weights. This becomes time consuming especially in higher dimensions. Another approach for the tree selection of vines is given by Kurowicka (2011), who also uses the heuristic of modeling most of the dependence in the lowest trees by starting with the last tree, where the edge with the lowest partial correlation is chosen. This is repeated, going backwards, until all trees are specified.

So we see that there is already quite a range of structure selection methods. However, all these approaches do not take into account the selected structure's implications on the validity of the simplifying assumption and therefore on the overall model fit. The methods we will propose in Section 4.3.4 remedy this issue. They are based on a test whether the simplifying assumption is fulfilled for a considered pair-copula term as described in the following section.

4.3.3 Test for constant conditional correlations (CCC test)

Formally, the simplifying assumption requires that all the pair-copulas of the vine appearing in Equation (2.3) can be written as follows:

$$c_{j_e k_e;D_e}\left(\cdot,\cdot;\mathbf{u}_{D_e}\right) \equiv c_{j_e k_e;D_e}\left(\cdot,\cdot\right).$$

This means that the pair-copula is independent of the conditioning variables, implying that the dependence associated with the copula is constant with respect to $\mathbf{u}_{D_e}$. A stochastic representation of the simplifying assumption is given by $(U_{j_e|D_e}, U_{k_e|D_e}) \perp \mathbf{U}_{D_e}$, denoting that the random variables $U_{j_e|D_e} = C_{j_e|D_e}(U_{j_e}|\mathbf{U}_{D_e})$ and $U_{k_e|D_e} = C_{k_e|D_e}(U_{k_e}|\mathbf{U}_{D_e})$ are jointly independent of $\mathbf{U}_{D_e}$ (Kurz and Spanhel, 2017). Without the simplifying assumption, one would have to allow $c_{j_e k_e;D_e}$ to depend on $\mathbf{u}_{D_e}$, making it a $(|D_e|+2)$-dimensional function to be estimated (which is d-dimensional for the pair-copula in the last tree). This would defeat the whole purpose and convenience of vine models, whose idea it is to express a d-dimensional copula just in terms of *bivariate* building blocks. While for low dimensions of the conditioning vector researchers have found ways to relax the simplifying assumption (Acar et al., 2012; Vatter and Nagler, 2016; Schellhase and Spanhel, 2017), in higher dimensions this does not seem feasible. Further discussion and implications of the simplifying assumption can be found in (Hobæk Haff et al., 2010; Stöber et al., 2013; Spanhel and Kurz, 2015; Killiches et al., 2017c).

Recently, Kurz and Spanhel (2017) developed a statistical testing method assessing the severity of the violation of the simplifying assumption for each conditional pair-copula in a vine copula. It tests the null hypothesis that the conditional correlation $\rho_{j_e k_e|D_e}$ associated with pair-copula $C_{j_e k_e;D_e}$ is constant with respect to the conditioning variables $\mathbf{U}_{D_e}$. For this purpose, the support Ω_0 of $\mathbf{U}_{D_e}$ is divided by a partition $\Gamma := \{\Omega_1,\ldots,\Omega_L\}$, with $L\in\mathbb{N}$, $\Omega_i\cap\Omega_j=\emptyset$ for $i\neq j$, and $P(\mathbf{U}_{D_e}\in\Omega_i)>0$ for $i=1,\ldots,L$. The idea of the test is that under the null hypothesis the conditional correlation between $U_{j_e|D_e}$ and $U_{k_e|D_e}$ given $\mathbf{U}_{D_e}\in\Omega_i$ does not depend on i. Hence, denoting $\rho_{\Omega_i} := \operatorname{Corr}(U_{j_e|D_e}, U_{k_e|D_e}|\mathbf{U}_{D_e}\in\Omega_i)$, the null hypothesis is equivalent to testing $\rho_{\Omega_1} = \ldots = \rho_{\Omega_L}$, which we will call the constant conditional correlation (CCC) assumption. To derive a test statistic we observe that for the vector of estimated conditional correlations $\hat{R}^{(n)}(\Gamma) := (\hat{\rho}^{(n)}_{\Omega_1},\ldots,\hat{\rho}^{(n)}_{\Omega_L})'$, estimated by the sample Pearson's correlation coefficient based on an i.i.d. sample of size n, it holds

$$\sqrt{n}(\hat{R}^{(n)}(\Gamma) - R(\Gamma)) \xrightarrow[n\to\infty]{d} N(\mathbf{0},\Sigma(\Gamma)),$$

where $R(\Gamma)$ is the vector of true conditional correlations and $\Sigma(\Gamma)$ is the asymptotic variance-covariance matrix (see Kurz and Spanhel, 2017). The asymptotic normality of

$\hat{R}^{(n)}(\Gamma)$ is used to derive an asymptotically χ^2-distributed test statistic by taking differences and normalizing. Defining the matrix $A \in \{-1, 0, 1\}^{(L-1)\times L}$ with $A_{ij} = \mathbb{1}\{i = j\} - \mathbb{1}\{i = j - 1\}$, $i = 1, \dots L - 1$, $j = 1, \dots L$, i.e.

$$A = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & & \\ & & & 1 & -1 \end{pmatrix},$$

where the omitted entries equal zero, we have that

$$A\hat{R}^{(n)}(\Gamma) = \begin{pmatrix} \hat{\rho}^{(n)}_{\Omega_1} - \hat{\rho}^{(n)}_{\Omega_2} \\ \hat{\rho}^{(n)}_{\Omega_2} - \hat{\rho}^{(n)}_{\Omega_3} \\ \vdots \\ \hat{\rho}^{(n)}_{\Omega_{L-1}} - \hat{\rho}^{(n)}_{\Omega_L} \end{pmatrix}.$$

The real-valued test statistic is defined by

$$T_n(\Gamma) = (A\hat{R}^{(n)}(\Gamma))'(A\hat{\Sigma}^{(n)}(\Gamma)A')^{-1}A\hat{R}^{(n)}(\Gamma),$$

where $\hat{\Sigma}^{(n)}(\Gamma)$ is a consistent estimator for $\Sigma(\Gamma)$. Since Kurz and Spanhel (2017) show that under regularity conditions and H_0 it holds that

$$nT_n(\Gamma) \xrightarrow[n\to\infty]{d} \chi^2_{L-1},$$

an asymptotic β-level test is given by

$$\mathbb{1}\{nT_n(\Gamma) \geq F^{-1}_{\chi^2_{L-1}}(1 - \beta)\},$$

where $F^{-1}_{\chi^2_{L-1}}(1 - \beta)$ denotes the $1 - \beta$ quantile of the χ^2 distribution with $L - 1$ degrees of freedom.

Of course, the power of this test highly depends on the chosen partition Γ. Therefore, Kurz and Spanhel (2017) adjust the test statistic to accommodate for the combination of different partitions and further use decision trees to find partitions where a possible violation of the simplifying assumption is most pronounced. The partition size L is determined by the algorithm and grows with $|D_e|$ (see Kurz and Spanhel, 2017, for details). Finally, they show that the test has a very high observed power compared to benchmark methods. The test is implemented as function `pacotest` in the R package `pacotest` (Kurz, 2017).

4.3.4 Two new tree selection algorithms

Motivation

As a motivation we first consider three-dimensional data sets. Here we know that there are three different possible vine tree structures and that the specification of the first tree already fully describes the whole tree structure. In Section 4.3.2 we have seen that Dißmann's algorithm chooses the tree structure that maximizes the dependence between the variables in the first tree. However, the resulting tree structure might yield a non-simplified vine copula even though the true model is a simplified vine copula. Hence, the model fit of Dißmann in terms of log-likelihood might not be optimal.

This motivates us to select tree structure by testing which combination of variables is "most simplified". In detail, we use the R function `pacotest` to test for the hypothesis that the CCC assumption holds for $U_i, U_j|U_k$ with $k \in \{1, 2, 3\}$ and $\{i, j\} = \{1, 2, 3\}\backslash\{k\}$. Then, we choose the tree structure which has the highest p-value in the second tree.

Example 1 We show how this works for a real life data set, namely a subset of the well-known seven-dimensional hydro-geochemical data set first investigated by Cook and Johnson (1986), consisting of $N = 655$ observations of log-concentrations of the three chemicals cobalt (U_1), titanium (U_2) and scandium (U_3) in water samples taken from a river near Grand Junction, Colorado. This data set has been examined with regard to the simplifying assumption by many researchers, e.g. Acar et al. (2012), Killiches, Kraus, and Czado (2017a) and Killiches, Kraus, and Czado (2017c). After transforming the data to the copula scale by applying the empirical probability integral transform, we estimate the Kendall's τ values for each of the three pairs. They are $\hat{\tau}_{12} = 0.54$, $\hat{\tau}_{13} = 0.36$ and $\hat{\tau}_{23} = 0.44$. Consequently, Dißmann's algorithm chooses edges $(1, 2)$ and $(2, 3)$ corresponding to pair-copulas c_{12} and c_{23} for the first tree resulting in the conditional copula $c_{13;2}$ to be modeled in the second tree. The Dißmann algorithm fits all pair-copulas as t copulas. The log-likelihood and AIC-values of the vine copula fitted by Dißmann's algorithm are 428.8 and -845.6, respectively (see also Structure 2 in Table 4).

Structure	Tree 1	conditional copula	p-value	sum of τ	log-lik	AIC
1	2–1–3	$c_{23;1}$	0.29	$0.54 + 0.36 = 0.90$	434.9	-857.8
2	1–2–3	$c_{13;2}$	0.01	$0.54 + 0.44 = 0.98$	428.8	-845.6
3	1–3–2	$c_{12;3}$	0.19	$0.36 + 0.44 = 0.80$	418.5	-825.1

Table 4: The p-values, sums of Kendall's τ values, log-likelihoods and AIC-values of the three possible tree structures for the three-dimensional uranium dataset.

Let us now examine the p-values of the test for constant conditional correlations for the three possible tree structures. They are 0.29, 0.01, 0.19 for the conditional copulas $c_{23;1}$, $c_{13;2}$ and $c_{12;3}$, respectively. Hence, we see that the structure chosen by Dißmann's algorithm has a strong indication for non-simplifiedness, such that we would rather choose one of the other two structures. The structure corresponding to the highest p-value (Structure 1 in Table 4) also has a larger sum of Kendall's τ values in the first tree compared to the other structure with a large p-value ($0.54+0.36 = 0.90$ vs. $0.36+0.44 = 0.80$), such that we would choose the structure with edges (1,2) and (1,3) in the first and (2,3; 1) in the second tree. The fitted vine copula of Structure 1 has a Tawn and a t copula in the first tree and a BB7 copula in the second tree. With a log-likelihood of 434.9 and an AIC of -857.8 it provides a better model fit than the one selected by Dißmann's algorithm. For completeness we note that Structure 3 in Table 4 has the worst fit, with the lowest sum of τ in the first tree and rather low p-value for the conditional copula in the second tree. All in all, with Structure 1 we found a way of describing this data set using a simplified vine copula, whereas in the recent literature it seemed necessary to use a non-simplified vine copula model when Structure 2 was used.

Extending this idea to higher dimensions, things get more complicated because there are superexponentially more tree structures to choose from. While e.g. in five dimensions there are still "only" 480 possible tree structures, in ten dimensions they already explode to $4.8705 \cdot 10^{14}$. See Morales-Nápoles (2011) for a general formula for counting the number of different vine structures. The question is now how to incorporate the information about the tests concerning the simplifiedness of the conditional copulas for the determination of the best vine tree.

Algorithm 1: Regular vine structure selection using CCC test based weights

The first approach that we propose is to choose the first tree (where there are no explicit considerations about simplifiedness yet) according to Dißmann's approach and then to incorporate the CCC tests in the higher level trees. In each tree, the test's p-values of all edges allowed by the proximity condition are calculated and used to find the maximum spanning tree. This would constitute a compromise between both approaches reflecting as much of the dependence as possible in the first tree and accounting for non-simplifiedness in the trees where conditional copulas have to be fitted. Further, one can also assign a score to each edge combining both the edge's p-value and estimated absolute Kendall's τ. Hence, the score of an edge e would be defined as

$$s_\alpha(e) := \alpha \cdot f_p(e) + (1-\alpha) \cdot f_\tau(e),$$

where α is a weighting factor and the functions f_p and f_τ map each edge e allowed by the proximity condition to a score regarding its degree of simplifiedness and absolute Kendall's τ values, respectively. For example, f_p could be the function mapping each edge to its associated rank of the p-value arising from the CCC test, i.e. $f_p(e_i) = i$, if the edges e_i are ordered such that p-value$(e_1) <$ p-value$(e_2) < \ldots <$ p-value(e_K). Here, K is the number of edges allowed by the proximity condition. Similarly, f_τ could rank the edges by their absolute empirical Kendall's τ values, such that the scales of f_p and f_τ would coincide. The weighting factor α can be chosen by the user. We will see in Section 4.3.5 that values around 0.6 yield good fits in terms of the AIC. After having determined the score of each edge, a maximum spanning tree on the edges using the scores $s_\alpha(e)$ is constructed. Then, similar to Dißmann's algorithm the pair-copulas corresponding to the chosen tree are fitted (with or without an independence test). Algorithm 1 implements this procedure.

Algorithm 1 Regular vine structure selection using CCC test based weights

```
Input: $d$-dimensional copula data, weighting factor $\alpha$, edge score functions $f_p$ and $f_\tau$.
1: Tree 1: estimate absolute empirical Kendall's $\tau$ values between all variables and find
   the corresponding maximum spanning tree. For each selected edge of the tree, fit a pair-
   copula based on the copula data and determine the corresponding pseudo observations
   for the next tree.
2: for $m = 2, \ldots, d-1$ do
3:     Tree $m$: for all edges $e$ allowed by the proximity condition, calculate the score
       $s_\alpha(e) = \alpha \cdot f_p(e) + (1-\alpha) \cdot f_\tau(e)$ using the pseudo observations and find the corresponding
       maximum spanning tree. For each selected edge of the tree, fit a pair-copula based on
       the pseudo observations and determine the corresponding pseudo observations for the
       next tree.
Output: R-vine with tree structure focused on constant conditional dependencies as well
as large dependencies in lower trees.
```

Example 2 We show how the algorithm works in detail for a 5-dimensional example. At first we generate a random R-vine object with random tree structure, families and parameters and simulate a sample of size 1000 from it (exact details on how the random vine is generated can be found in Section 4.3.5). The resulting true vine specification is a D-vine with pair-copulas given in Table 5.

The chosen vine's first tree connects the edges (2,3), (2,4), (1,5) and (4,5). In a first step we fit an R-vine to the simulated data using the function `RVineStructureSelect` in which Dißmann's algorithm is implemented. The resulting first two trees are displayed in Figure 24 as the gray graphs.

Tree 1	Tree 2
C_{23}: Clayton (0.44) ($\tau_{23} = 0.18$) C_{24}: type 1 Tawn (2.43, 0.81) ($\tau_{24} = 0.51$) C_{15}: type 2 Tawn (6, 0.43) ($\tau_{15} = 0.39$) C_{45}: survival Gumbel (2.69) ($\tau_{45} = 0.63$)	$C_{34;2}$: survival BB6 (1.07,2.31) ($\tau_{34;2} = 0.58$) $C_{25;4}$: BB8 (1.62, 0.7) ($\tau_{25;4} = 0.1$) $C_{14;5}$: 270 degree rotated type 1 Tawn (-3.89, 0.54) ($\tau_{14;5} = -0.45$)
Tree 3	Tree 4
$C_{35;24}$: survival Joe (2.31) ($\tau_{35;24} = 0.42$) $C_{12;45}$: Frank (18.67) ($\tau_{12;45} = 0.8$)	$C_{13;245}$: survival BB1 (0, 2.39) ($\tau_{13;245} = 0.58$)

Table 5: D-vine copula specification of Example 2.

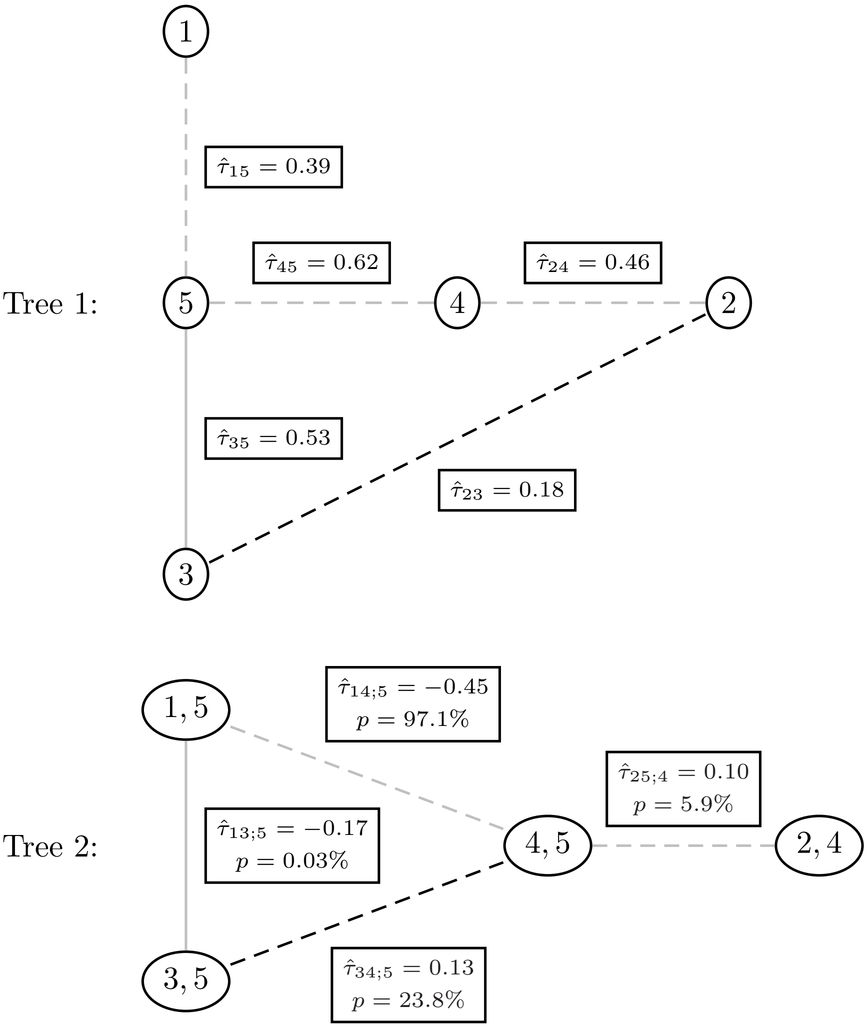

Figure 24: First two trees selected by Dißmann's algorithm (gray). In Tree 1, the empirical Kendall's τ values of each edge are given in the boxes and the dashed lines correspond to the true tree structure. In Tree 2, additionally to Kendall's τ, the p-values of the conditional correlation test are given and the dashed lines mark the tree selected by Algorithm 1.

The first tree of the fitted R-vine is quite close to the true one (highlighted by a dashed line). Edges (2,4), (1,5) and (4,5) are selected and the true copula families are chosen with parameters very close to the true ones ($\hat{\tau}_{24} = 0.46$, $\hat{\tau}_{15} = 0.39$ and $\hat{\tau}_{45} = 0.62$). Only edge (3,5) is selected instead of (2,3) since it has a higher empirical Kendall's τ (0.53 compared to 0.18).

Due to the proximity condition in the second tree, node $(2,4)$ can only be connected to

(4,5) (since it is the only other node containing a 4 or a 2), resulting in the edge (2,5; 4). Regarding nodes (1,5), (3,5) and (4,5) the proximity condition allows all of these nodes to be connected. Only focusing on the absolute empirical Kendall's τ value, Dißmann's algorithm chooses edges (1,3; 5) and (1,4; 5) since $|\hat{\tau}_{14;5}| > |\hat{\tau}_{13;5}| > |\hat{\tau}_{34;5}|$. Altogether, the selected second tree constitutes a path and therefore the higher level trees are already determined by the proximity condition. The AIC of the R-vine fitted by Dißmann's algorithm is −7775.5.
Next, we consider the results of our proposed algorithm on this data set. Of course, the selected first tree coincides with the one from Dißmann's algorithm. However, in the second tree, when choosing which of the nodes (1,5), (3,5) and (4,5) to connect, we see that even though edge (1,3; 5) has a slightly larger absolute Kendall's $\hat{\tau}$ than edge (3,4; 5), its p-value (0.03%) is much smaller than the 23.8% of edge (3,4; 5). Thus, for $\alpha \geq 0.5$ our proposed algorithm would choose edges (3,4; 5) and (1,4; 5) for the second tree (cf. the dashed graph in the lower panel of Figure 24). After having fit the third and fourth trees accordingly, the overall resulting R-vine proves to yield a much better fit than Dißmann's algorithm with an AIC of −8272.9.
In the simulation study presented in Section 4.3.5 we repeat this procedure 1000 times and find that Algorithm 1 achieves a better or equal AIC-value in 80.5 percent of the simulations.

Algorithm 2: C-vine structure selection using CCC test based weights
One major disadvantage of Algorithm 1 is that the overall tree structure is already strongly affected by the first tree due to the proximity condition. For example, if the first tree is fitted with a path-like structure, then the whole vine is automatically a D-vine since the proximity condition only allows for one possible (also path-like) structure in the higher trees. In this case we cannot account for a possible non-simplifiedness of the model at all. For example, in three dimensions Algorithm 1 would always yield the same structure as the Dißmann algorithm since for three nodes every tree constitutes a path. So, e.g. for the three-dimensional uranium example described in Section 4.3.4, where the consideration of the test for the simplifying assumption improved the structure fit, the proposed algorithm would be futile.
Thus we have to find a way how to sequentially select a vine structure incorporating in the current tree the information which conditional correlations are not constant in the subsequent tree while still keeping the subsequent graph as flexible as possible. We have seen that a path-like structure in the first tree already determines the structure of the remaining trees. Conversely, we know that we retain the highest flexibility by fitting a star-like structure in the first tree since then the proximity condition imposes no restriction (such that all nodes in the subsequent tree are allowed to be linked by an edge). Following this logic for all trees, a star-like structure is reasonable for every tree level. The root node v_m of each star should be chosen such that on the one hand the sum of absolute empirical Kendall's τ values between this root node and all the other variables (conditioned on the earlier root nodes $v_1, \ldots, v_{m-1}$) is large. On the other hand, conditioned on the root node v_m (and all the earlier root nodes) pairs of the remaining variables should have rather constant conditional correlations, i.e. the p-values of the CCC test conditioning on the root nodes should be large.
Formally, at tree level m we assign each remaining node $v \in N_m := \{1, \ldots, d\} \backslash \{v_1, \ldots, v_{m-1}\}$ the score $\psi_\alpha(v)$ (operating on nodes v instead of edges e as in Algorithm 1) defined as

$$\psi_\alpha(v) = \alpha \cdot g_p(v) + (1 - \alpha) \cdot g_\tau(v).$$

We then choose the one with the highest score as a root node for tree m. Again, α is a weighting factor and g_p and g_τ are functions mapping a node to ranked scores regarding the CCC tests and the Kendall's τ values, respectively. To be precise, let $p_{ij;v_1,\ldots,v_{m-1},v}$ denote the p-value of the test for constant conditional correlation of $C_{ij;v_1,\ldots,v_{m-1},v}$ and let r be the function mapping a p-value $p_{i_0j_0;v_1,\ldots,v_{m-1},v_0}$ to its rank among all possible p-values $p_{ij;v_1,\ldots,v_{m-1},v}$ with $i,j,v \in N_m$ pairwise distinct. Then, g_p calculates for all $v \in N_m$ the p-value score

$$p(v) := \sum_{i,j\in N_m\setminus\{v\},\, i\neq j} r(p_{ij;v_1,\ldots,v_{m-1},v}),$$

and then maps each v to its rank among all p-value scores $\{p(v)\}_{v\in N_m}$ (e.g. the root node with the smallest p-value score would be assigned rank 1). Instead of choosing the function r to be the rank transformation, other transformations such as the logarithm are possible. However, the results of the simulation study show that the rank transformation yields the best results (see Appendix A.6).

Regarding Kendall's τ, g_τ similarly calculates the τ scores

$$t(v) := \sum_{i\in N_m\setminus\{v\}} |\hat{\tau}_{iv;v_1,\ldots,v_{m-1}}|,\ v \in N_m$$

and then assigns each v to its corresponding rank among these τ scores. The Kendall's τ associated to $c_{iv;v_1,\ldots,v_{m-1}}$ is estimated as the empirical Kendall's τ between the pseudo observations $\hat{\mathbf{u}}_{i|v_1,\ldots,v_{m-1}}$ and $\hat{\mathbf{u}}_{v|v_1,\ldots,v_{m-1}}$ and is denoted by $\hat{\tau}_{iv;v_1,\ldots,v_{m-1}}$.

After the optimal root node is found, similar to the other algorithms all pair-copulas implied by the resulting tree have to be fitted (with or without independence test) for the calculation of the pseudo observations of the next tree level (see Algorithm 2).

Algorithm 2 C-vine structure selection using CCC test based weights

Input: d-dimensional copula data, weighting factor α, node score functions g_p and g_τ.

```
for $m = 1, \ldots, d-1$ do
    Tree $m$: Determine the optimal root node by evaluating score $\psi_\alpha(v) = \alpha \cdot g_p(v) + (1-\alpha) \cdot g_\tau(v)$ for all nodes $v \in N_m$ and choose the maximal one. Fit pair-copulas between this root node and the remaining nodes based on the pseudo observations. Calculate the corresponding pseudo observations needed for the tree selection in the next step.
```

Output: C-vine with tree structure focused on constant conditional dependencies as well as large dependencies in lower trees.

Using this algorithm with any $\alpha > 0.5$, we retrieve the AIC-optimal tree structure for the uranium example discussed in Section 4.3.4.

Example 2 (continued) We revisit the 5-dimensional example from the previous section and want to apply Algorithm 2 to the simulated data (with $\alpha = 0.6$ and rank-transformed p-values). Table 6 displays the information needed to find the optimal root node v_1 in the first tree of the C-vine.

For example, the p-value score $p(1) = 91$ is the sum of the ranks of the 6 p-values $p_{ij;1}$, $i, j \in \{2,3,4,5\}$, $i \neq j$, among the 30 possible p-values. Since 91 is the third largest p-value score, its rank among the p-value scores is 3. Similarly, since $\sum_{i=2}^{5} |\hat{\tau}_{i1}| = 1.04$ is the smallest of the τ scores node 1 gets assigned rank 1. Hence, the overall score is calculated as $\psi_\alpha(v) = 0.6 \cdot 3 + 0.4 \cdot 1 = 2.2$. Doing this for the other possible root nodes we see that node 5 has the largest p-value and τ scores and therefore with an overall score of 5.0 is selected as the first root node. This procedure is repeated for all trees yielding a C-vine with root node ordering 5–4–2–1–3 and an AIC of -7446.6. While we see that this model has a higher

node v	$p(v)$	rank$(p(v))$	$t(v)$	rank$(t(v))$	$\psi_\alpha(v)$
1	91	3	1.04	1	2.2
2	89	2	1.47	3	2.4
3	105	4	1.31	2	3.2
4	69	1	1.72	4	2.2
5	111	5	1.98	5	5.0

Table 6: p-value scores $p(v)$ and τ scores $t(v)$ with associated ranks, and the overall score $\psi_\alpha(v)$ with $\alpha = 0.6$ for the determination of the root node v_1 in the first tree of the C-vine fitted by Algorithm 2.

(i.e. worse) AIC than the ones selected by Algorithm 1 and by Dißmann's algorithm (which may be due to the fact that the true model is a D-vine and Algorithm 2 fits a C-vine), it still finds the AIC-optimal out of the 60 possible C-Vines structures. Further, we will see in Section 4.3.5, where this procedure is repeated 1000 times, that in 73.6% of the time Algorithm 2 finds a C-vine with a better AIC than the one of Dißmann's chosen R-vine.

4.3.5 Simulation study

Setup

We perform an extensive simulation study, evaluating the performance of the two proposed algorithms in many different scenarios. For each scenario, we repeat $R = 1000$ simulations of samples with size n of randomly generated d-dimensional R-vine copulas. For these, we sample uniformly one out of the $2^{(d-2)(d-3)/2}$ different structure matrices with natural ordering as described in Joe et al. (2011) using the R function `RVineMatrixSample` from `VineCopula`; for each pair-copula, a random copula family out of the 12 families currently implemented in the package `VineCopula` is selected (see Appendix A.1 for a list of the implemented families). Regarding the copulas' parameters, $Beta(2, 2)$ distributed Kendall's τ values are generated and multiplied by -1 with a probability of 50% (note that the densities of some families have to be rotated by 90 degrees to accommodate negative dependence). The $Beta(2, 2)$ distribution is symmetric around its mode 0.5 with a variance of 0.05 and has a 95% confidence interval given by $[0.09, 0.91]$. Next, for the two-parametric copula families the second parameter is randomly generated from suitable distributions (see Appendix A.2 for details). Finally, the first parameter of all pair-copulas is derived from the sampled Kendall's τ values and the second parameters (where required) using the R function `BiCopTau2Par` (see Joe, 1997, for details). An exemplary random 5-dimensional R-vine is given in the Table 5 of Section 4.3.4.

In the simulation study, for each of the samples of size n from these random R-vines we will calculate the AIC-values of the vines fitted using our proposed algorithms (here without the mentioned independence tests). The AIC-values will be compared to those of the fitted Dißmann vines. Before presenting the results of the simulation study, we shortly investigate the influence of the weighting factor α on the AIC-values of the models fitted by our algorithms.

Choice of weighting factor α

In the simulation study we will consistently use the weighting factor $\alpha = 0.6$. This choice is justified because we heuristically observe that the AIC of our fitted values on average is lowest for medium sized weighting factors with the best performances for $\alpha = 0.6$. We generate 1000 random d-dimensional vine copulas (as described in Section 4.3.5), simulate a sample of size n from each copula and apply our algorithms to the sample using weighting factors $\alpha \in \{0, 0.1, 0.2, \ldots, 0.9, 1\}$. For each α we compute the AIC of the fitted models averaged over the 1000 repetitions. Overall, we find that in most scenarios $\alpha = 0.6$ yields the lowest average AIC or close to it. Figure 25 displays exemplary results for Algorithm 2 in the setting $n = 1000$ and $d = 5, 10, 30$.

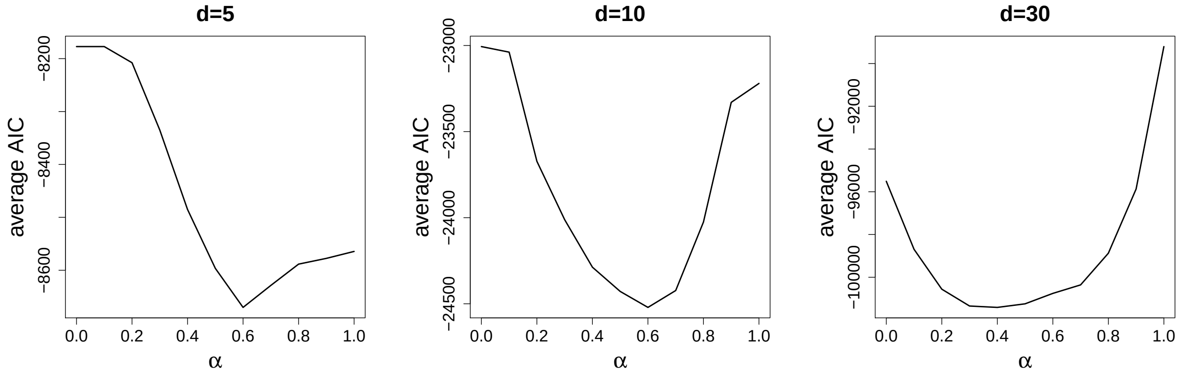

Figure 25: AIC-values of vine copulas fitted by Algorithm 2 depending on the weighting factor α in the setting $n = 1000$ and $d = 5, 10, 30$, averaged over 1000 repetitions.

We observe a U-shape of the plots indicating that choosing the weighting factor too small or too large does not yield optimal fits. So, focusing exclusively on the Kendall's τ values or the p-values of the CCC test apparently is not sufficient. For $d = 5$ and $d = 10$ the minimum of the average AIC-values is attained for $\alpha = 0.6$ and in dimension thirty for $\alpha = 0.4$ with the average AIC of $\alpha = 0.6$ being still quite close to the minimum. The results for sample sizes $n = 400$ and $n = 3000$ given in Appendix A.4 are very similar and lead the same conclusion of $\alpha = 0.6$ being optimal. Finally, the same study applied for Algorithm 1 also yields an optimum of $\alpha = 0.6$.

Results

In our simulation study we let the dimension d of the randomly generated vine copulas vary between 3 and 50 and the sizes of the simulated samples n between 400 and 3000. The percentage of times where our algorithms performs better or equal than Dißmann is given in Table 7 (the percentages of equal performance are given in brackets and are omitted if they are 0). The results of the table are also visualized in Figure 29 of Appendix A.7.

		d								
Algorithm	n	3	4	5	7	10	15	20	30	50
Algorithm 1	400	100 (100)	92.2 (89.0)	80.8 (69.3)	54.9 (30.8)	41.9 (5.9)	34.8 (0.2)	30.0	26.5	22.1
	1000	100 (100)	93.1 (89.0)	82.3 (69.7)	59.9 (32.3)	41.4 (4.8)	35.9 (0.1)	30.3	28.9	27.3
	3000	100 (100)	92.6 (88.6)	79.9 (68.0)	58.3 (31.3)	41.6 (4.2)	36.2 (0.2)	35.3	21.8	18.2
Algorithm 2	400	84.4 (49.6)	68.8 (3.1)	67.7	75.9	82.0	93.2	96.7	99.2	99.8
	1000	91.1 (51.5)	75.5 (3.8)	73.6	80.2	88.7	95.8	99.0	99.6	100
	3000	93.4 (47.6)	77.1 (3.7)	75.8	82.2	91.7	96.8	99.2	100	100

Table 7: Percentages of better or equal performance regarding the AIC-value of the two algorithms compared to Dißmann's algorithm for different dimensions d and sample sizes n based on 1000 data sets sampled from randomly generated R-vines (in brackets the percentages of equal performance are given).

We see that in general our two proposed algorithms perform very well compared to Dißmann's algorithm. Of course, in low dimensions Algorithm 1 performs very similar to Dißmann's algorithm since there the first tree greatly determines the model fit. So as we

already noted, in three dimensions Algorithm 1 and Dißmann's algorithm always find the same vine structure and even in four and five dimensions they coincide in almost 90% and 70% of the times, respectively. In higher dimensions, when the selected tree structures differ more often, Algorithm 1 manages to find a better structure than Dißmann's algorithm in more than one third of the simulations for dimensions 10-15 and around one fourth of the time for dimensions larger than 20. Further, it seems that the sample size has no effect on which of the two algorithms performs better since the percentages stay rather constant when the sample size changes from 400 to 3000.
Concerning Algorithm 2 the results are even more promising. In every scenario it is better or equal than Dißmann's algorithm in more than two thirds of the time. Especially in high dimensions ($d \geq 15$) it outperforms in more than 90% of simulations. Moreover, we note that a larger sample size helps to increase the advantage of Algorithm 2 even further with increasing percentages for n going from 400 to 3000, regardless of the considered dimension.

Regarding the question, how much better Algorithm 2 performs than Dißmann's algorithm we present in Figure 26 boxplots of the difference between the AIC-values per observation of the models found by Dißmann's and our second algorithm for $n = 1000$. Positive values imply a worse performance of the Dißmann algorithm.

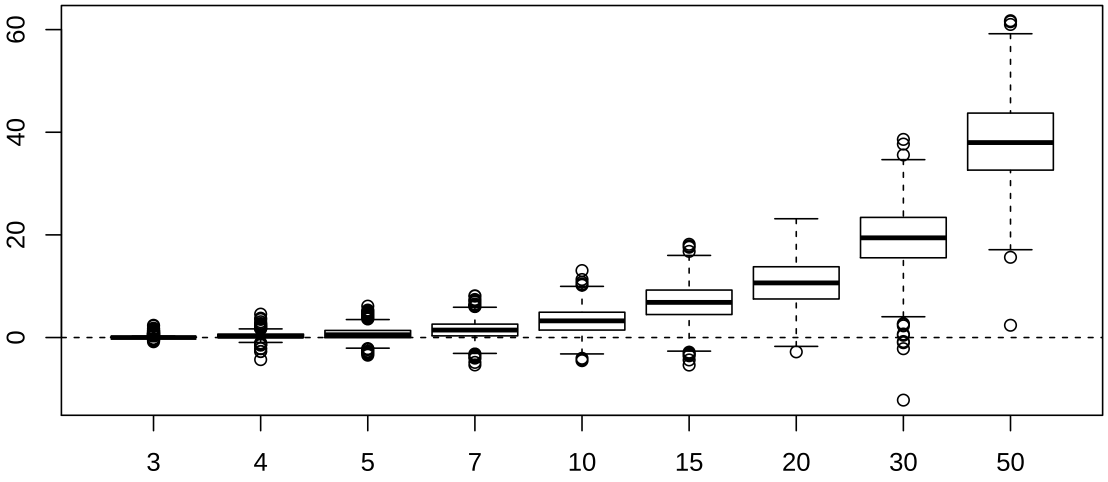

Figure 26: Boxplots of the difference between the AIC-values per observation of the vines chosen by Dißmann's algorithm and Algorithm 2.

The boxplots affirm the results from Table 7. The differences between the AIC of Dißmann's algorithm and Algorithm 2 are mostly positive and become larger as the dimension increases. We further see that the magnitude of AIC improvements is much larger than that of the AIC losses. Since the values are shown per observations, we observe AIC improvements of up to 60,000 in 50 dimensions. Due to the scaling the values in the lower dimensions appear to be close to zero. However, we performed t-tests with the null hypothesis that the mean of the AIC differences is zero and could reject it in favor of Algorithm 2 with p-values smaller than 10^{-30} for all dimensions. The boxplots in the scenarios $n = 400$ and $n = 3000$ are very similar and can be seen in Appendix A.3.
In our simulation study the pair copulas' Kendall's τ values were sampled from $[-1, 1]$. Since many real life data sets (especially in finance) consist of variables that are pairwise positively dependent we repeat our simulation study sampling only positive Kendall's τ values. An excerpt of the results is given in the first two rows of Table 11 in Appendix A.5. The results are quite similar: Algorithm 1 performs slightly worse than in the case where

the τ values were sampled from $[-1, 1]$ and Algorithm 2 has better performance in low and worse performance in high dimensions.

In the construction of the algorithms we noted that the pair-copula selection can be done with or without a prior independence test deciding whether the independence copula can be used. The results above were obtained without the use of independence test. In the last two lines of Table 11 in Appendix A.5 we present the results in the positive dependence case when our as well as Dißmann's algorithms use the independence test at level $\beta = 0.05$. All the results are very similar to those without the independence test implying that the relative performances of the algorithms do not strongly depend on the use of the independence test.

Finally, for both algorithms we chose the rank functions for scoring the Kendall's τ and p-values of the test for constant conditional correlations. This is justified in Appendix A.6.

Computational times and a faster version of Algorithm 2

We consider whether the better performance of our algorithm comes at a higher computational cost. The average computational times (based on 100 repetitions) of fitting a vine copula with the three competing algorithms depending on the dimension d are displayed for $n = 1000$ in Table 8.

	d								
Algorithm	3	4	5	7	10	15	20	30	50
Dißmann	0.42	0.92	1.47	3.14	6.63	15.6	24.6	67.8	155.3
Algorithm 1	0.41	0.83	1.42	2.94	6.42	15.2	23.5	65.1	157.7
Algorithm 2	0.49	1.20	2.37	6.65	20.08	72.9	154.6	731.2	3409.4

Table 8: Average computational times of the three algorithms for $n = 1000$ in different dimensions.

We see that the computational times of Dißmann's algorithm and Algorithm 1 are always very close to each other with Algorithm 1 being even faster most of the times. Algorithm 2 clearly is much slower than the other algorithms since for the determination of the optimal root node each possible pair-copula of the current tree level has to be fitted for the calculation of all the required p-values. This effect increases with the dimension making Algorithm 2 comparably fast in low dimensions, only 3 times slower in 10 dimensions, but roughly 20 times slower for $d = 50$.

A possible way to decrease the computational time of Algorithm 2 is given by the following adjustment: in a first step we restrict the allowed pair-copula families exclusively to the Gaussian copula and proceed as before to determine the estimated tree structure. In a second step all pair-copulas appearing in this tree structure are fitted again, now allowing for all pair-copula families. The idea is that the resulting vine copula models hopefully do not differ too much from the unadjusted fit since the p-values of the CCC tests should not be very sensitive to the chosen families used to determine the pseudo observations and thus yield similar choices for the optimal root nodes. At the same time the computational effort is greatly reduced since the most time consuming step of Algorithm 2 is the fitting of the pair-copulas to determine the p-values of the CCC tests. Table 9 displays the percentages of computational times and AIC-values of the adjusted version of Algorithm 2 relative to the unadjusted version for dimensions $d = 5, 10, 30$ and sample sizes $n = 400, 1000, 3000$.

We see that the adjustment greatly reduces the computational time making the algorithm almost as fast as Dißmann's algorithm. Further, the loss in performance is negligibly small with relative performances ranging between 97 and 99 percent. Thus, we found a

d	5			10			30		
n	400	1000	3000	400	1000	3000	400	1000	3000
% time	52.0	52.1	51.9	31.2	29.6	28.2	15.8	14.0	12.9
% AIC	98.5	97.7	97.1	98.4	98.7	97.4	99.0	98.2	98.2

Table 9: Percentages of computational times and AIC-values of the adjusted version of Algorithm 2 relative to the unadjusted version for dimensions $d = 5, 10, 30$ and sample sizes $n = 400, 1000, 3000$.

way to significantly improve the computational times of Algorithm 2 without losing too much performance. Nevertheless, in the following applications we will continue using the unadjusted version of Algorithm 2 in order to achieve the best results possible.

4.3.6 Real data examples

In order to assess the performance of our proposed algorithms when confronted with real data, we revisit several data sets that have been used in the recent vine copula literature. For the description of each of the data sets we refer the reader to the respective references. The uranium data set, of which we already examined a three-dimensional subset in Section 4.3.4 was introduced in Cook and Johnson (1986). The bike data sets (Schallhorn et al., 2017) are daily and hourly records of bike rentals in Washington, D.C., together with local climate data (temperature, perceived temperature, humidity and wind speed). The hourly bike data set additionally contains the variable hour. The MAGIC (Major Atmospheric Gamma-ray Imaging Cherenkov) data has been analyzed in the context of classification by Nagler and Czado (2016). We only show the results for the subset with the classification hadron. Further, the Norwegian data is a financial data set of Norwegian and international market variables used in the context of truncated regular vines by Brechmann et al. (2012) and Killiches, Kraus, and Czado (2017c). Two higher dimensional financial data sets are given by the CDS data (Brechmann et al., 2013; Kraus and Czado, 2017a) and the EuroStoxx 50 data (Brechmann and Czado, 2013; Killiches et al., 2017c). Finally, in order to include another non-financial data set, we also consider a subset of the Concrete Compressive Strength Data Set (Yeh, 1998), containing all of its continuous variables, namely *Cement*, *Coarse aggregate*, *Fine aggregate* and *Concrete compressive strength*.

The dimensions of the data sets range between 3 and 52 with sample sizes between 655 and 17379. All of the data was transformed to the copula scale using adequate preprocessing methods where necessary (e.g. GARCH models for time series data as in Liu and Luger, 2009) and empirical probability integral transforms. Table 10 shows the AIC-values of the vine copulas fitted by the three competing algorithms (note that we use the unadjusted version of Algorithm 2). We further provide in brackets for each vine copula fitted by the three algorithms the number of pair-copulas for which the test for constant conditional correlations is rejected at the 5% level. So, e.g. in the case of the 3-dimensional uranium data set the CCC test is rejected with a p-value of 0.01 for the conditional copula modeled by Dißmann's algorithm (yielding a "1" in brackets), while Algorithm 2 chooses a structure with no rejection (implying the "0").

Similarly to its three-dimensional subset the complete seven-dimensional uranium data set is best fitted by Algorithm 2. Out of the 15 conditional pair-copulas of the vine fitted by Dißmann's algorithm four are deemed non-simplified at 5% level. Algorithm 2 is able to reduce this number to one, thus yielding a better model fit. Regarding the four-dimensional concrete data set we see that Algorithm 2 is also able to improve the model fit. It may seem

Data	uranium	concrete	bike daily	bike hourly	uranium
d	3	4	5	6	7
n	655	1030	731	17379	655
Dißmann	−845.6 (1)	−522.9 (2)	−4423.5 (3)	−89859.6 (9)	−1759.9 (4)
Algorithm 1	−845.6 (1)	−522.9 (2)	−4423.5 (3)	−89842.4 (9)	−1756.4 (4)
Algorithm 2	**−857.8** (0)	**−551.6** (3)	**−4454.1** (3)	**−92040.5** (10)	**−1817.2**(1)

Data	MAGIC	Norwegian	CDS	EuroStoxx
d	10	19	38	52
n	6688	1187	1371	985
Dißmann	−61359.4 (23)	−12906.4 (18)	**−40985.7** (31)	**−62377.8** (87)
Algorithm 1	**−61649.7**(24)	**−12946.6** (10)	−40922.4 (25)	−62280.1 (65)
Algorithm 2	−58197.4 (17)	−12756.5 (12)	−40514.5 (32)	−62074.8 (68)

Table 10: AIC-values of the Dißmann's algorithm, Algorithm 1 and the unadjusted version of Algorithm 2 applied to real data sets. In brackets the number of pair-copulas is given for which the CCC test is rejected at 5% level.

surprising, that compared to the other methods it increases the number of pair-copulas that violate the simplifying assumption. However, if we have a closer look at the p-values of the fitted pair-copulas, we see that the Dißmann fitted vine contains one pair-copula with p-value equal to zero which apparently severely impairs the model's likelihood. Similar observations can be made for the two bike data sets.

Considering the higher dimensional data sets we notice that Algorithm 2 looses its competitive advantage. The ten-dimensional MAGIC data as well as the 19-dimensional Norwegian data achieve their best fit using Algorithm 1. For the MAGIC data set it increases the overall non-simplified pair-copulas while reducing the number of strong violations (with p-value equal to zero) from 12 to 8. The good performance of Dißmann's algorithm for the high-dimensional finance data sets is not surprising. It is a stylized fact that such financial data sets are modeled fairly well by t copulas (see for example Demarta and McNeil, 2005). Stöber et al. (2013) have shown that vine decompositions of t copulas are of the simplified form, independently of the vine's tree structure. Thus, the CCC test is rejected less often for vines modeling financial data. So for the 38-dimensional CDS data set, in the Dißmann vine only 31 of the 666 fitted pair-copulas violate the simplifying assumption, which is slightly less than the 5% that we would expect if the null hypothesis would be true for all pair-copulas. Thus, for structure selection purposes the p-value score is rather unimportant in this example. Consequently, Dißmann's heuristic of modeling the strongest dependencies in the lower trees yields the best results and the restriction to C-vines imposed by Algorithm 2 is apparently too severe. Similar arguments hold for the 52-dimensional EuroStoxx data set, where only 6.8% of the pair-copulas fitted by Dißmann's algorithm violate the simplifying assumption.

Regarding the pair-copula families selected by the three algorithms, we observe that they do not differ considerably between the algorithms. Not surprisingly, most of the pair-copula families of the financial data sets are t copulas, with some Tawn, Frank and BB1 copulas in the mix. The concrete data was modeled mainly by Tawn, Frank and Gauss copulas. The same copula families were chosen for the two bike data sets, with some additional BB8 copulas and rotations thereof. For the uranium data set, Frank, Joe, t and Tawn copulas were favored, while the most commonly selected families of the MAGIC data were Tawn, BB8 and t copulas. So we see that in these examples many non-Gaussian copulas are se-

lected such that Gaussian copulas would not provide satisfying model fits.
All in all, we have seen that the algorithms proposed in this section are able to improve the model fit compared to Dißmann's algorithm, especially when the Dißmann vine exhibits significant non-simplifiedness as assessed by the CCC test.

4.3.7 Summary

We proposed two new algorithms for the sequential selection of the tree structure of a vine copula model. We extended currently existing methods by incorporating tests which gage the validity of the simplifying assumption for every pair-copula. This resulted in vine copula models which frequently have a better model fit than the benchmark given by Dißmann's algorithm.
Further, we revisited the three-dimensional uranium data set, which is famous for its non-simplified nature when considering its vine decomposition using Dißmann's algorithm. However, we found out that using the vine decomposition where the conditioning in the second tree is done with respect to cobalt, the resulting vine is of the simplified form and provides a better model fit.
In the real data application we saw that our proposed algorithms work especially well, when the vine fitted by Dißmann's algorithm contains many pair-copulas violating the simplifying assumption. Thus, from a practitioner's point of view, after fitting a d-dimensional vine copula using Dißmann's algorithm one should always count the number of pair-copulas for which the test of constant conditional correlations is rejected at some level β. If this number is considerably larger than $\beta(d-1)(d-2)/2$ (the expected number of rejections), one should refit the model using our proposed algorithms, accepting larger computational times as a trade-off for a likely better model fit.

5 Conclusion and outlook

As Pierre-Simon Laplace already stated in 1825, “the most important questions of life are indeed, for the most part, only problems of probability.” In this thesis we tried to provide methods that answer some of these questions. The advantages of vine copulas as flexible models to describe the dependence between multiple random variables have been praised in many instances. During my time as a PhD student we aimed to add to the prominence of vines by treating two completely different aspects. On the one hand we developed a method to use all the advantageous flexibilities of vine copulas in order to perform quantile regression, a statistical method with almost endless applications. As soon as there is an influence of a set of covariates on some response to be measured, our D-vine copula based quantile regression is a valid tool to quantify these effects. Outperforming all established quantile regression methods in prediction ability it should be considered as one of the standard approaches when one is confronted with the task of quantile regression, especially thanks to its implementation in R allowing also for mixed discrete and continuous data sets.
On the other hand we illuminated several aspects regarding the simplifying assumption. Being the only assumption which mitigates the flexibility of vine copulas it is interesting to see what kind of implications it entails and in which cases it is indeed a model restriction. For this, we first had a look at three-dimensional vine copulas and their contour surfaces in several simplified and non-simplified scenarios. We saw that simplified and non-simplified vines in fact can differ quite a bit, however mostly in pathological examples. For the most part, simplified vines turned out to be smooth versions of their non-simplified counterparts representing their main features such as correlations and tail behavior. We further developed a statistical test for the decision whether to use a simplified or a non-simplified vine copula to model given data. Since simplified vine models are nested in the non-simplified ones, we argued that a significant distance between two fitted simplified and non-simplified vine copulas implies that the non-simplified model is superior to the simplified one. Using a modified version of the Kullback-Leibler distance, we constructed the test which turned out to have a high power and produce reasonable results in real data applications. Finally, we focused our attention on the relationship between the tree structure of a fitted vine copula and the validity of the simplifying assumption. We found out that the two are strongly connected and constructed two new structure selection algorithms with the focus of producing vine copula models that violate the simplifying assumption as little as possible. It turned out that we were successful, beating the performance of Dißmann’s algorithm, which is the structure selection heuristic that was most commonly used for the fitting of vine copulas in the past years.

Of course, the end of doctoral studies is rather determined by limited time than by the lack of research fields. All topics discussed in this thesis can be seen as starting points for further research. Consider for example the area of using vines to perform regression. The quantile regression methods could be generalized from using D-vines to C-vines or even a certain subclass of regular vines (i.e., the class of vines where the response is a leaf in every tree of the vine’s structure), to add even more flexibility to the model. Further, the limitation to the consideration of only one response could be alleviated to allow for a vector of responses, thus, e.g. admitting the quantification of the joint impact of a group of stressed financial entities on another group. Another idea would be to make the transition from quantile to mean regression. In an ongoing research project, the use of D-vine copula based quantile regression for post-processing ensembles for weather forecasting is examined.

Regarding the simplifying assumption, it would be interesting to see if one could find a way to visualize or quantify its implications for vine copulas with dimensions higher than three. Lastly, we admit that our new structure selection algorithms are still heuristically motivated. Many different other approaches are conceivable to suggest further answers to the still open question of how to select the best fitting simplified tree structure of a vine copula.

A Appendix to Section 4.3

A.1 Pair-copula families implemented in `VineCopula`

The pair-copula families currently implemented in the `VineCopula` package are Gaussian, t*, Clayton, Gumbel, Frank, Joe, BB1*, BB6*, BB7*, BB8*, Tawn type 1* and Tawn type 2* with respective rotations. The stars indicate two-parametric families. Definitions all of these copula families except the Tawn copula can be found in Joe (2014). The less well known Tawn copula (Tawn, 1988) is a bivariate extreme value copula with representation

$$C(u_1, u_2) = \exp\left\{ [\log(u_1) + \log(u_2)]\, A\left(\frac{\log(u_2)}{\log(u_1)\log(u_2)} \right) \right\},$$

where the Pickands dependence function A is given by

$$A(t) = (1 - \psi_1)(1 - t) + (1 - \psi_2)t + [(\psi_1(1 - t))^\theta + (\psi_2 t)^\theta]^{1/\theta}.$$

In the `VineCopula` package the Tawn type 1 copula corresponds to the Tawn copula with $\psi_2 \equiv 1$ and the Tawn type 2 copula corresponds to the Tawn copula with $\psi_1 \equiv 1$.

A.2 Simulation of the second parameter for two-parametric copula families

t copula The degrees of freedom of the t copula are sampled from $3 + G$, where $G \sim \mathit{Gamma}(3, 3)$. This implies an expected value of 4 and the 95% confidence interval $[3.2, 5.4]$.

BB1, BB6, BB7 The second parameters of the BB1, BB6 and BB7 copulas are sampled from $1 + 3B$, where $B \sim \mathit{Beta}(4, 2)$. This implies an expected value of 3 and the 95% confidence interval $[1.84, 3.85]$. When the dependence is negative, the second parameter is multiplied with -1.

BB8, Tawn The second parameters of the BB8 and Tawn copulas are sampled from a $\mathit{Beta}(4, 2)$ distribution. This implies an expected value of 2/3 and the 95% confidence interval $[0.28, 0.95]$. When the dependence is negative, the second parameter of the BB8 copula is multiplied with -1.

A.3 Differences of AIC-values

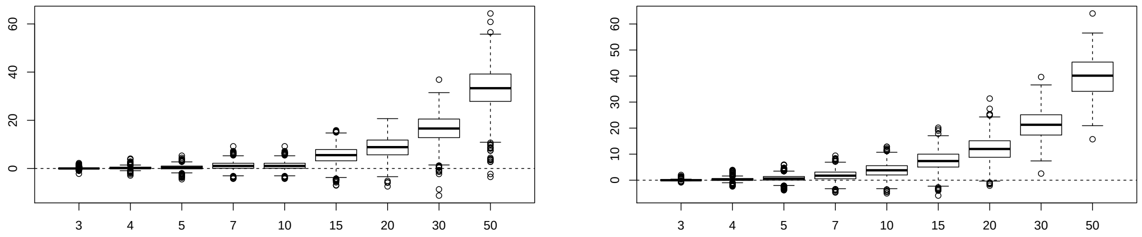

Figure 27: Boxplots of the difference between the AIC-values per observation of the vines chosen by Dißmann's algorithm and Algorithm 2 for $n = 400$ (left panel) and $n = 3000$ (right panel).

A.4 Choice of weighting factor α

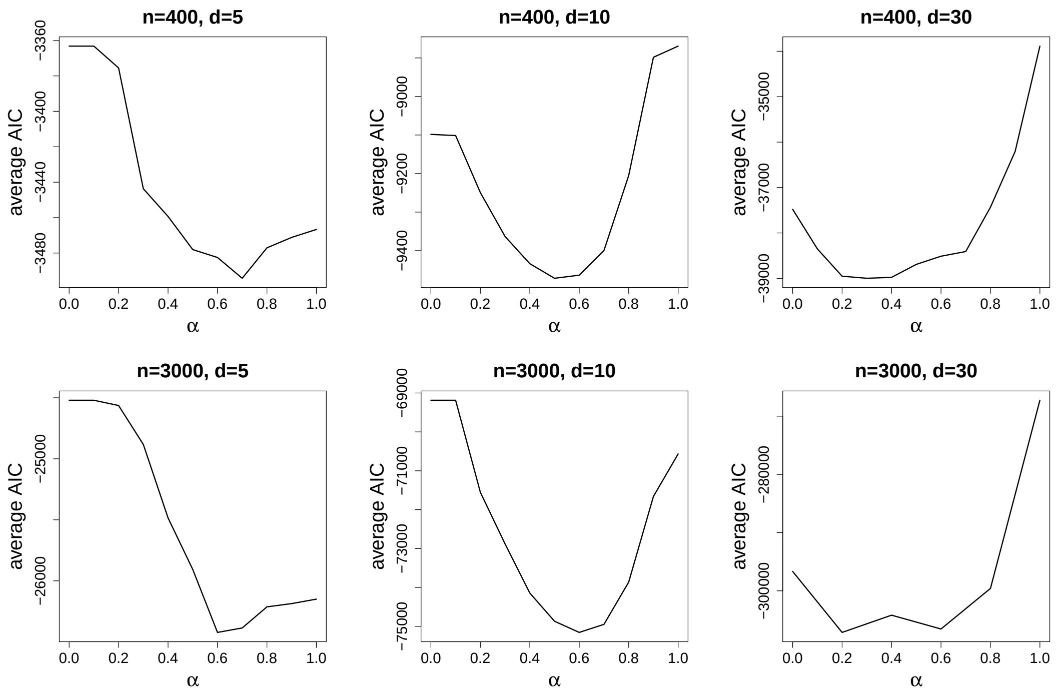

Figure 28: AIC-values of vine copulas fitted by Algorithm 2 depending on the weighting factor α in the settings $n = 400$ (top row) and $n = 3000$ (bottom row) for $d = 5, 10, 30$, averaged over 1000 repetitions.

A.5 Simulation study results for positive dependence and independence tests

		d			
Ind. test	Algorithm	3	5	10	30
Without	Algorithm 1	100 (100)	73.1 (53.4)	37.0 (3.2)	18.8
	Algorithm 2	94.1 (22.9)	83.1	86.1	91.9
With	Algorithm 1	100 (100)	73.6 (54.0)	33.0 (3.2)	16.8
	Algorithm 2	94.4 (22.9)	82.9	85.8	92.9

Table 11: Percentages of better or equal performance regarding the AIC-value of the two algorithms compared to Dißmann's algorithm for $n = 1000$ and random vine copulas with only positive dependence (in brackets the percentages of equal performance are given). In the first two rows pair-copula no independence tests were performed and in the last two rows they were performed with level $\beta = 0.05$.

A.6 Choice of p-value transformation function r

	d			
r	3	5	10	30
rank	91.1 (51.5)	73.6	88.7	99.6
identity	89.9 (48.5)	71.1	87.5	99.5
logarithm	89.9 (48.5)	72.4	77.0	94.8

Table 12: Percentages of better or equal performance regarding the AIC-value of Algorithm 2 compared to Dißmann's algorithm depending on the transformation function r for $n = 1000$ and $d = 3, 5, 10, 30$ (in brackets the percentages of equal performance are given).

A.7 Plots of simulation study results

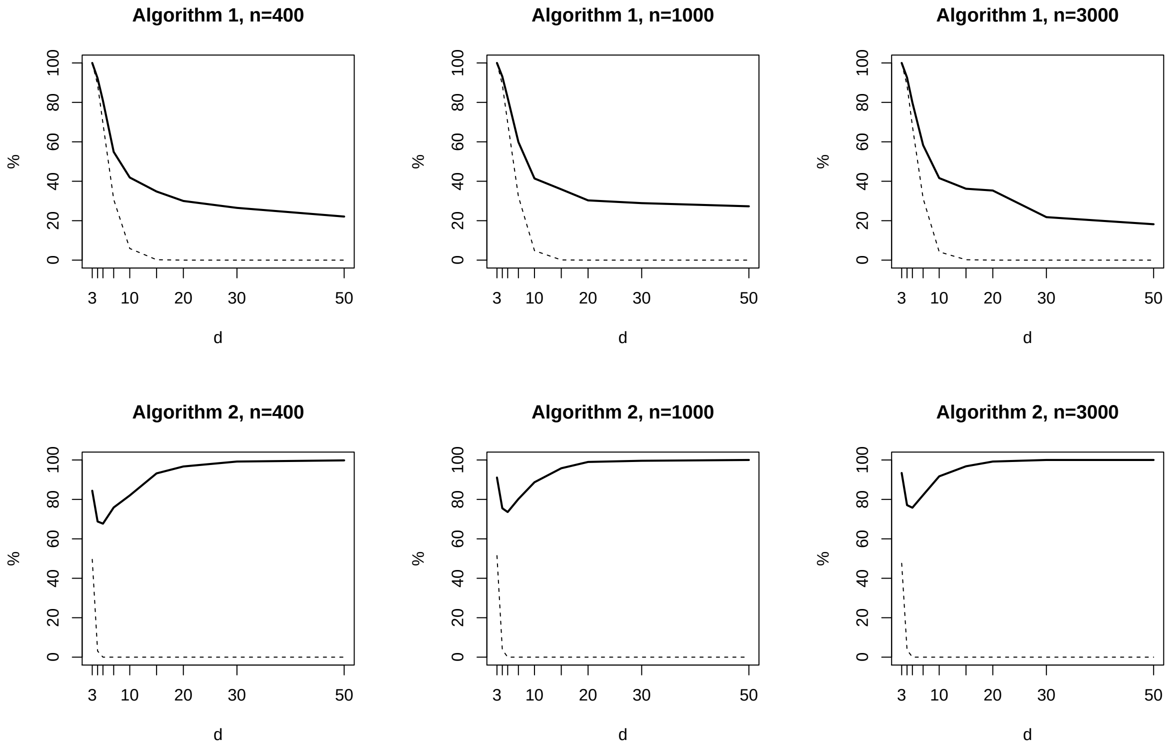

Figure 29: Percentages of better or equal performance regarding the AIC-value of the two algorithms compared to Dißmann's algorithm for different dimensions d and sample sizes n based on 1000 data sets sampled from randomly generated R-vines (the dashed lines represent the percentages of equal performance).

Bibliography

Aas, K. (2016), "Pair-Copula Constructions for Financial Applications: A Review," *Econometrics*, 4, 43.

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.

Adrian, T. and Brunnermeier, M. K. (2016), "CoVaR," *American Economic Review*, 106, 1705–1741.

Akaike, H. (1998), "Information theory and an extension of the maximum likelihood principle," in *Selected Papers of Hirotugu Akaike*, Springer, pp. 199–213.

Almeida, C., Czado, C., and Manner, H. (2016), "Modeling high-dimensional time-varying dependence using dynamic D-vine models," *Applied Stochastic Models in Business and Industry*, 32, 621–638.

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.*

Bedford, T. and Cooke, R. M. (2002), "Vines: A new graphical model for dependent random variables," *Annals of Statistics*, 30, 1031–1068.

Bernard, C. and Czado, C. (2015), "Conditional quantiles and tail dependence," *Journal of Multivariate Analysis*, 138, 104–126.

Bouyé, E. and Salmon, M. (2009), "Dynamic copula quantile regressions and tail area dynamic dependence in Forex markets," *The European Journal of Finance*, 15, 721–750.

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, 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, 68–85.

Brechmann, E. C., Czado, C., and Paterlini, S. (2014), "Flexible dependence modeling of operational risk losses and its impact on total capital requirements," *Journal of Banking & Finance*, 40, 271–285.

Brechmann, E. C., Hendrich, K., and Czado, C. (2013), "Conditional copula simulation for systemic risk stress testing," *Insurance: Mathematics and Economics*, 53, 722–732.

Brownlees, C. T. and Engle, R. F. (2016), "SRISK: A Conditional Capital Shortfall Measure of Systemic Risk," *Available at SSRN 1611229.*

Chen, X., Koenker, R., and Xiao, Z. (2009), "Copula-based nonlinear quantile autoregression," *The Econometrics Journal*, 12, S50–S67.

Cherubini, U., Luciano, E., and Vecchiato, W. (2004), *Copula methods in finance*, John Wiley & Sons.

Cook, R. and Johnson, M. (1986), “Generalized burr-pareto-logistic distributions with applications to a uranium exploration data set.” *Technometrics*, 28, 123–131.

Cooke, R., Kurowicka, D., and Wilson, K. (2015a), “Sampling, conditionalizing, counting, merging, searching regular vines,” *Journal of Multivariate Analysis*, 138, 4 – 18.

Cooke, R. M., Joe, H., and Chang, B. (2015b), “Vine Regression,” *Resources for the Future Discussion Paper*, 15–52.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001), *Introduction to algorithms*, vol. 6, MIT press Cambridge.

Czado, C. (2010), “Pair-Copula Constructions of Multivariate Copulas,” in *Copula Theory and Its Applications: Proceedings of the Workshop Held in Warsaw, 25-26 September 2009*, eds. Jaworski, P., Durante, F., Härdle, W. K., and Rychlik, T., Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 93–109.

Czado, C., Gärtner, F., and Min, A. (2011), “Analysis of Australian electricity loads using joint Bayesian inference of D-Vines with autoregressive margins,” *Dependence Modeling: Vine Copula Handbook, World Scientific Publishing*, 265–280.

Czado, C., Jeske, S., and Hofmann, M. (2013), “Selection strategies for regular vine copulae,” *Journal de la Société Française de Statistique*, 154, 174–191.

Demarta, S. and McNeil, A. J. (2005), “The t copula and related copulas,” *International Statistical Review/Revue Internationale de Statistique*, 111–129.

Diestel, R. (2005), *Graph Theory (Graduate Texts in Mathematics)*, Springer.

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.

Duong, T. (2016a), *ks: Kernel Smoothing*, R package version 1.10.1.

— (2016b), “Non-parametric smoothed estimation of multivariate cumulative distribution and survival functions, and receiver operating characteristic curves,” *Journal of the Korean Statistical Society*, 45, 33–50.

Embrechts, P., McNeil, A., and Straumann, D. (1999), “Correlation: Pitfalls and alternatives,” *RISK Magazine*, 69–71.

Erhardt, T. M. and Czado, C. (2015), “Standardized drought indices: A novel uni- and multivariate approach,” *arXiv preprint, arXiv:1508.06476.*

Erhardt, T. M., Czado, C., and Schepsmeier, U. (2015), “Spatial composite likelihood inference using local C-vines,” *Journal of Multivariate Analysis*, 138, 74–88.

Erhardt, V. and Czado, C. (2012), “Modeling dependent yearly claim totals including zero claims in private health insurance,” *Scandinavian Actuarial Journal*, 2012, 106–129.

Fanaee-T, H. and Gama, J. (2013), “Event labeling combining ensemble detectors and background knowledge,” *Progress in Artificial Intelligence*, 1–15.

Favre, A.-C., El Adlouni, S., Perreault, L., Thiémonge, N., and Bobée, B. (2004), "Multivariate hydrological frequency analysis using copulas," *Water Resources Research*, 40, W01101.

Fenske, N., Kneib, T., and Hothorn, T. (2012), "Identifying risk factors for severe childhood malnutrition by boosting additive quantile regression," *Journal of the American Statistical Association*, 106, 494–510.

Fischer, M., Kraus, D., Pfeuffer, M., and Czado, C. (2017), "Stress Testing German Industry Sectors: Results from a Vine Copula Based Quantile Regression," *Risks*, 5, 38.

Frahm, G., Junker, M., and Szimayer, A. (2003), "Elliptical copulas: applicability and limitations," *Statistics & Probability Letters*, 63, 275–286.

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, 347–368.

Genest, C., Gendron, M., and Bourdeau-Brien, M. (2009a), "The advent of copulas in finance," *The European Journal of Finance*, 15, 609–618.

Genest, C., Gerber, H. U., Goovaerts, M. J., and Laeven, R. J. (2009b), "Editorial to the special issue on modeling and measurement of multivariate risk in insurance and finance," *Insurance: Mathematics and Economics*, 44, 143–145.

Gruber, L. and Czado, C. (2015), "Sequential bayesian model selection of regular vine copulas," *Bayesian Analysis*, 10, 937–963.

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, 1296–1310.

Hobæk Haff, I., Frigessi, A., and Maraun, D. (2015), "How well do regional climate models simulate the spatial dependence of precipitation? An application of pair-copula constructions," *Journal of Geophysical Research: Atmospheres*, 120, 2624–2646.

Hobæk Haff, I. and Segers, J. (2015), "Nonparametric estimation of pair-copula constructions with the empirical pair-copula," *Computational Statistics & Data Analysis*, 84, 1–13.

Hwang, C. and Shim, J. (2005), "A simple quantile regression via support vector machine," in *Advances in Natural Computation*, Springer, pp. 512–520.

International Monetary Fund (2009), *Global Financial Stability Report*, Washington DC.

Joe, H. (1996), "Families of m-variate distributions with given margins and m(m-1)/2 bivariate dependence parameters," *Lecture Notes-Monograph Series*, 120–141.

— (1997), *Multivariate models and multivariate dependence concepts*, CRC Press.

— (2014), *Dependence modeling with copulas*, Boca Raton, FL: CRC Press.

Joe, H., Cooke, R. M., and Kurowicka, D. (2011), "Regular vines: generation algorithm and number of equivalence classes," *Dependence Modeling: Vine Copula Handbook*, 219–231.

Kauermann, G. and Schellhase, C. (2014), "Flexible pair-copula estimation in D-vines using bivariate penalized splines," *Statistics and Computing*, 24, 1081–1100.

Killiches, M. and Czado, C. (2015), “Block-Maxima of Vines,” in *Extreme Value Modelling and Risk Analysis: Methods and Applications*, eds. Dey, D. and Yan, J., Boca Raton, FL: Chapman & Hall/CRC Press, pp. 109–130.

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, 95–117.

— (2017b), “Model distances for vine copulas in high dimensions,” *Statistics and Computing*, doi:10.1007/s11222-017-9733-y.

— (2017c), “Using model distances to investigate the simplifying assumption, goodness-of-fit and truncation levels for vine copulas,” *arXiv preprint, arXiv:1610.08795.*

Kim, J.-M., Jung, Y.-S., Sungur, E. A., Han, K.-H., Park, C., and Sohn, I. (2008), “A copula method for modeling directional dependence of genes,” *BMC bioinformatics*, 9, 225.

Koenker, R. (2005), *Quantile Regression*, Cambridge University Press: Cambridge UK.

— (2011), “Additive models for quantile regression: Model selection and confidence bandaids,” *Brazilian Journal of Probability and Statistics*, 25, 239–262.

Koenker, R. and Bassett, G. (1978), “Regression quantiles,” *Econometrica: journal of the Econometric Society*, 46, 33–50.

Komunjer, I. (2013), “Quantile prediction,” in *Handbook of Economic Forecasting*, Elsevier, pp. 767–785.

Krämer, N., Brechmann, E. C., Silvestrini, D., and Czado, C. (2013), “Total loss estimation using copula-based regression models,” *Insurance: Mathematics and Economics*, 53, 829–839.

Kraus, D. and Czado, C. (2017a), “D-vine copula based quantile regression,” *Computational Statistics & Data Analysis*, 110C, 1–18.

— (2017b), “Growing simplified vine copula trees: improving Dißmann’s algorithm,” *arXiv preprint, arXiv:1703.05203.*

Krupskii, P. and Joe, H. (2015), “Structured factor copula models: Theory, inference and computation,” *Journal of Multivariate Analysis*, 138, 53–73.

Kullback, S. and Leibler, R. A. (1951), “On information and sufficiency,” *The Annals of Mathematical Statistics*, 22, 79–86.

Kumar, P. (2010), “Probability distributions and estimation of Ali-Mikhail-Haq copula,” *Applied Mathematical Sciences*, 4, 657–666.

Kurowicka, D. (2011), “Optimal truncation of vines,” *Dependence Modeling: Vine Copula Handbook*, 233.

Kurowicka, D. and Cooke, R. M. (2006), *Uncertainty analysis with high dimensional dependence modelling*, John Wiley & Sons.

Kurz, M. S. (2017), *pacotest: Testing for Partial Copulas and the Simplifying Assumption in Vine Copulas*, version 0.2.

Kurz, M. S. and Spanhel, F. (2017), "Testing the simplifying assumption in high-dimensional vine copulas," *Unpublished working paper.*

Li, Q., Lin, J., and Racine, J. S. (2013), "Optimal bandwidth selection for nonparametric conditional distribution and quantile functions," *Journal of Business & Economic Statistics*, 31, 57–65.

Lichman, M. (2013), "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences.

Liu, Y. and Luger, R. (2009), "Efficient estimation of copula-GARCH models," *Computational Statistics & Data Analysis*, 53, 2284–2297.

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, 535–549.

McNeil, A. J. and Nešlehová, J. (2009), "Multivariate Archimedean Copulas, d-Monotone Functions and l-Norm Symmetric Distributions," *The Annals of Statistics*, 3059–3097.

Meinshausen, N. (2006), "Quantile regression forests," *The Journal of Machine Learning Research*, 7, 983–999.

Min, A. and Czado, C. (2010), "Bayesian inference for multivariate copulas using pair-copula constructions," *Journal of Financial Econometrics*, 8, 511–546.

Morales-Nápoles, O. (2011), "Counting vines," in *Dependence Modeling: Vine Copula Handbook*, eds. Kurowicka, D. and Joe, H., Singapore, SG: World Scientific, chap. 9, pp. 189–218.

Nagler, T. (2017), "A generic approach to nonparametric function estimation with mixed data," *arXiv preprint, arXiv:1704.07457.*

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.

Nagler, T., Schellhase, C., and Czado, C. (2017), "Nonparametric estimation of simplified vine copula models: comparison of methods," *arXiv preprint, arXiv:1701.00845.*

Nelsen, R. B. (2007), *An Introduction to Copulas*, Springer Science & Business Media.

Nikoloulopoulos, A. K., Joe, H., and Li, H. (2012), "Vine copulas with asymmetric tail dependence and applications to financial return data," *Computational Statistics & Data Analysis*, 56, 3659–3673.

Nikoloulopoulos, A. K. and Karlis, D. (2008a), "Copula model evaluation based on parametric bootstrap," *Computational Statistics & Data Analysis*, 52, 3342–3353.

— (2008b), "Multivariate logit copula model with an application to dental data," *Statistics in Medicine*, 27, 6393–6406.

Noh, H., Ghouch, A. E., and Bouezmarni, T. (2013), “Copula-based regression estimation and inference,” *Journal of the American Statistical Association*, 108, 676–688.

Noh, H., Ghouch, A. E., and Van Keilegom, I. (2015), “Semiparametric Conditional Quantile Estimation through Copula-Based Multivariate Models,” *Journal of Business & Economic Statistics*, 33, 167–178.

Oh, D. H. and Patton, A. J. (2017), “Modeling dependence in high dimensions with factor copulas,” *Journal of Business & Economic Statistics*, 35, 139–154.

Panagiotelis, A., Czado, C., and Joe, H. (2012), “Pair copula constructions for multivariate discrete data,” *Journal of the American Statistical Association*, 107, 1063–1072.

Parzen, E. (1962), “On estimation of a probability density function and mode,” *The annals of mathematical statistics*, 33, 1065–1076.

Pereira, G., Veiga, Á., Erhardt, T., and Czado, C. (2016), “Spatial R-vine copula for streamflow scenario simulation,” in *Power Systems Computation Conference (PSCC), 2016*, IEEE, pp. 1–7.

R Core Team (2017), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

Renard, B. and Lang, M. (2007), “Use of a Gaussian copula for multivariate extreme value analysis: some case studies in hydrology,” *Advances in Water Resources*, 30, 897–912.

Rosenblatt, M. (1952), “Remarks on a Multivariate Transformation,” *Ann. Math. Statist.*, 23, 470–472.

Salvadori, G. and De Michele, C. (2007), “On the use of copulas in hydrology: theory and practice,” *Journal of Hydrologic Engineering*, 12, 369–380.

Schallhorn, N., Kraus, D., Nagler, T., and Czado, C. (2017), “D-vine quantile regression with discrete variables,” *arXiv preprint, arXiv:1705.08310.*

Schellhase, C. and Spanhel, F. (2017), “Estimating Non-Simplified Vine Copulas Using Penalized Splines,” *Statistics and Computing.*

Schepsmeier, U. (2016), “A goodness-of-fit test for regular vine copula models,” *Econometric Reviews*, 1–22.

Schepsmeier, U., Stöber, J., Brechmann, E. C., Graeler, B., Nagler, T., and Erhardt, T. (2017), *VineCopula: Statistical Inference of Vine Copulas*, version 2.1.1.

Sklar, A. (1959), “Fonctions dé Repartition á n Dimensions et leurs Marges,” *Publications de l’Instutut de Statistique de l’Université de Paris*, 8, 229–231.

Spanhel, F. and Kurz, M. S. (2015), “Simplified vine copula models: Approximations based on the simplifying assumption.” *arXiv preprint, arXiv:1510.06971.*

Spokoiny, V., Wang, W., and Härdle, W. K. (2013), “Local quantile regression,” *Journal of Statistical Planning and Inference*, 143, 1109–1129.

Stöber, J. and Czado, C. (2012), "Sampling pair copula constructions with applications to mathematical finance," in *Simulating Copulas: Stochastic Models, Sampling Algorithms, and Applications*, eds. Mai, J.-F. and Scherer, M., World Scientific Publishing Co, Singapore.

Stöber, J., Joe, H., and Czado, C. (2013), "Simplified pair copula constructions—limitations and extensions," *Journal of Multivariate Analysis*, 119, 101–118.

Stöber, J. and Schepsmeier, U. (2013), "Estimating standard errors in regular vine copula models," *Computational Statistics*, 28, 2679–2707.

Tawn, J. A. (1988), "Bivariate extreme value theory: models and estimation," *Biometrika*, 397–415.

Vatter, T. (2016), *gamCopula: Generalized additive models for bivariate conditional dependence structures and vine copulas*, R package version 0.0-1.

Vatter, T. and Nagler, T. (2016), "Generalized additive models for pair-copula constructions," *arXiv preprint, arXiv:1608.01593.*

Venables, W. N. and Ripley, B. D. (2002), *Modern Applied Statistics with S*, New York: Springer.

Wu, T. Z., Yu, K., and Yu, Y. (2010), "Single-index quantile regression," *Journal of Multivariate Analysis*, 101, 1607–1621.

Xiao, Z. and Koenker, R. (2009), "Conditional Quantile Estimation for Generalized Autoregressive Conditional Heteroscedasticity Models," *Journal of the American Statistical Association*, 104, 1696–1712.

Yeh, I.-C. (1998), "Modeling of strength of high-performance concrete using artificial neural networks," *Cement and Concrete research*, 28, 1797–1808.