Technische Universität München
Department of Mathematics

Kernel Methods for Vine Copula Estimation

Master's Thesis
by
Thomas Nagler

Supervisor:	Prof. Claudia Czado, Ph.D.
Advisor:	Prof. Claudia Czado, Ph.D.
Submission date:	25 August 2014

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

Garching, 25 August 2014

Acknowledgments

First and foremost, I want to thank Prof. Claudia Czado for giving me the opportunity to work on this topic and especially for her excellent supervision. The many fruit- and joyful discussions were not only beneficial to this thesis, but also helped me to evolve as a researcher. I also want to thank Prof. Elif Acar for her interest in my work as well as the insightful discussions about it and nonparametric methods in general. I am likewise grateful to Prof. Irène Gijbels for introducing me to nonparametric inference for copulas in the first place.

I also want to thank my fellow students and friends Chris, David, Gabriel, Jojo, and Lisa, all of whom certainly sweetened my five years of study and pulled me through the first couple of semesters.

Lastly and most importantly, I want to express my deep gratitude to my siblings and parents. Your unconditional support during my whole life was an indispensable source of energy and constituted the foundation for my personal development.

Abstract

Vine copulas are highly flexible models for high-dimensional dependence. So far, the vast majority of the literature focuses on parametric modeling, but most of the time parametric assumptions are made solely for practical convenience. In many situations this will lead to a misspecified model and consistency of estimators is lost. We present a novel, fully nonparametric approach for the estimation of a vine copula density which is a hybrid of kernel density estimation and kernel regression.

After laying the theoretical foundations, the performance of the parametric and kernel estimators will be compared in simulation examples. We find that in some situations the kernel approach leads to a significant improvement in performance and can serve as a powerful tool for exploratory data analysis. Lastly, we present a real-data application in biomedical sciences and discuss possible directions for future research.

Zusammenfassung

Vine Copulas sind flexible Modelle für hochdimensionale Abhängigkeit. Der Großteil der Forschung hat sich bisher auf parametrische Modelle beschränkt, allerdings sind parametrische Annahmen meist nur durch Bequemlichkeit motiviert. Oftmals führt dies zu fehlerhaft spezifizierten Modellen und die resultierenden Schätzer sind inkonsistent. Wir stellen einen neuartigen, gänzlich nichtparametrischen Ansatz zur Schätzung von Vine Copula Dichten vor, welcher sich aus Kerndichteschätzung und Kernel-Regression zusammensetzt.

Nach einer Einführung der theoretischen Grundlagen vergleichen wir parametrische und nichtparametrische Methoden anhand von Simulationen. Diese zeigen, dass die Kernel-Methode in manchen Situationen zu signifikanten Verbesserungen führt und darÃijber hinaus ein nützliches Werkzeug für die explorative Datenanalyse ist. Als Abschluss zeigen wir eine Anwendung der Methode in der Biomedizin und geben einen Ausblick auf zuküftige Forschungsgebiete.

Contents

Chapter 1

Introduction

Since the seminal work of Sklar (1959) copula modeling for multivariate stochastic dependence has been extensively studied in mathematical statistics. In the last two decades, copulas became a standard tool in many fields, such as finance, insurance and hydrology. However, classical copula models turned out to lack flexibility in higher dimensional settings. More recently, Aas et al. (2009) introduced the flexible class of *vine copulas* building upon the earlier work of Joe (1997) and Bedford and Cooke (2001, 2002). Here, copulas of arbitrary dimension are constructed using only bivariate blocks — so-called *pair-copulas*.

The vast majority of the literature on vine copulas focuses on parametric modeling, but most of the time parametric assumptions are made solely for practical convenience. In many situation this will lead to a misspecified model and consistency of estimators is lost. This becomes a major issue when one or more of the pair-copulas do not conform with any of the popular parametric copula families. A nonparametric approach can overcome this, but usually involves more complex and computationally intensive methods.

Hobæk Haff and Segers (2012) analyze a nonparametric estimator of the *cdf* of a vine copula based on empirical copulas. Other authors developed nonparametric estimators of the vine copula density. Weiß and Scheffer (2012) build on the popular Bernstein estimator, Schellhase (2012) develops a penalized maximum likelihood approach for Bernstein polynomials and hierarchical B-splines. Another very common nonparametric method is *kernel density estimation.* Despite its popularity in estimation of general densities, so far only Lopez-Paz et al. (2013) utilized a kernel estimator for vine copula densities.

In this thesis, we will thoroughly investigate kernel estimators of a vine copula density. Generally speaking, the estimation procedure can be split in two parts: Estimation of bivariate copula densities, and estimation of *h-functions*, i.e. conditional *cdfs* corresponding to a pair of uniformly distributed random variables. In parametric models as well as the mentioned nonparametric estimators, the h-function can be obtained as an immediate byproduct of the density estimate. In contrast, we will present kernel estimators for both parts separately and join them to get fully nonparametric kernel estimators of vine copula densities.

The remainder of this thesis is organized as follows. Chapter 2 gives a review of the statistical concepts our work builds on. Section 2.1 introduces copulas in general, and vine copulas in particular; Section 2.2 focuses on bivariate kernel density estimation. In Chapter 3, we investigate a variety of kernel estimators for bivariate copula densities and close with a comparison of all presented methods by means of a simulation study. Chapter 4 deals with kernel estimation of h-functions. In Chapter 5, we put the pieces together to a general kernel estimation approach for vine copula densities. The abilities of this method are illustrated with two simulation examples and a real-data application stemming from biomedical research. Furthermore, possible directions for future research are discussed. Chapter 6 concludes.

Chapter 2

Theoretical background

In this chapter we will present the necessary theoretical background for the remainder of this thesis. This includes the basic theory of copulas, dependence measures and Vine copula models as well as the foundations of bivariate kernel density estimation. The following pages will mainly give definitions and address particular issues that will reappear in later chapters. They are not at all meant as a comprehensive overview of any of the topics. For a more extensive treatment the reader is advised to consult the references given in the respective sections.

2.1 Dependence modeling with copulas

Copulas are objects that contain all information on the dependence in a multivariate random vector. In a copula model, a multivariate distribution is split into two parts: the marginal distributions and the dependency structure. This approach allows to separate the effects coming from one or the other part, which is impossible in classical multivariate models. It facilitates estimation of multivariate distributions and also allows for much more flexibility regarding their shape. As of today, copulas are widely used in many fields that call for multivariate modeling, such as finance, insurance, geostatistics and hydrology. Recommended readings are the excellent texts of Nelsen (2006) and Embrechts et al. (2003).

Throughout this thesis, we will assume that all random variables are continuous. From a statistical viewpoint, a copula is defined as the distribution function of a d-dimensional random vector with uniform margins.

Definition 2.1. *A function $C : [0,1]^d \to [0,1]$ is called a d-dimensional* ***copula*** *if there exists a random vector $(U_1, \dots, U_d)$, with $U_j \sim U[0,1]$, $j = 1, \dots, d$, such that*

$$\mathrm{P}(U_1 \leq u_1, \dots, U_d \leq u_d) = C(u_1, \dots, u_d),$$

i.e. C is the cumulative distribution function of $(U_1, \dots, U_d)$.

The following theorem is the core of copula theory. It is due to Abe Sklar (1959) and states that any multivariate distribution can be split into its margins and a copula.

Theorem 2.1 (Sklar's Theorem). *Let F be a continuous d-dimensional distribution function with marginal distributions $F_1, \dots, F_d$. Then there exists a unique d-dimensional copula C such that for all $(x_1, \dots, x_d) \in \mathbb{R}^d$ it holds*

$$F(x_1, \dots, x_d) = C\Big(F_1(x_1), \dots, F_d(x_d)\Big). \tag{2.1}$$

Conversely, if C is a d-dimensional copula and $F_1, \dots F_d$ are univariate distribution functions, F as defined in (2.1) *is a d-dimensional distribution function.*

The idea of Sklar's theorem is simple. Take $(X_1, \dots, X_d)$ as a random vector with marginal distributions $F_1, \dots, F_d$ and let F be its joint distribution. Recall also the well known fact that $F_j(X_j) \sim U[0,1]$, for all $j = 1, \dots, d$. The corresponding copula C is then defined as the distribution function of $(F_1(X_1), \dots, F_d(X_d))$. This explains why we define a copula as the distribution of uniformly distributed random variables. The most trivial examples of a copula correspond to the cases of independence and perfect positive and negative dependence.

Example 2.1 (Independence copula). *For $U_1, \dots, U_d \overset{iid}{\sim} U[0,1]$, we have*

$$P(U_1 \leq u_1, \dots, U_d \leq u_d) = \prod_{j=1}^{d} u_j = \Pi(u_1, \dots, u_d),$$

where we call Π the independence copula.

Example 2.2 (Comonotonicity copula). *Perfect positive dependence is meant as follows. Let $U \sim U[0,1]$, then the random vector $(U_1, \dots, U_d) = (U, \dots, U)$ exhibits perfect positive dependence. In this case,*

$$\begin{aligned} P(U_1 \leq u_1, \dots, U_d \leq u_d) = P(U \leq u_1, \dots, U \leq u_d) &= \min\{u_1, \dots, u_d\} \\ &= M(u_1, \dots, u_d), \end{aligned}$$

where we call M the comonotonicity copula.

Example 2.3 (Countermontonicity copula). *Perfect negative dependence is meant as follows. Let $U \sim U[0,1]$, then the random vector $(U_1, U_2) = (U, 1-U)$ exhibits perfect negative dependence. In this case,*

$$\begin{aligned} P(U_1 \leq u_1, U_2 \leq u_2) = P(U \leq u_1, 1-U \leq u_2) &= \max\{u_1 + u_2 - 1, 0\} \\ &= W(u_1, \dots, u_d), \end{aligned}$$

where we call W the countermonotonicity copula. Note that it is only defined for a bivariate random vector, since perfect negative dependence is not possible in higher dimensions.

2.1.1 Parametric copula families

Sklar's theorem gives us a simple way to construct copula functions. By inversion of (2.1), we get

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

From this formula, we can also directly obtain a representation of the corresponding copula density $c(u_1, \ldots, u_d) = \partial^d C(u_1, \ldots, u_d)/(\partial u_1 \cdots \partial u_d)$. Denote $f, f_1, \ldots, f_d$ as the densities corresponding to $F, F_1, \ldots F_d$ respectively. Then,

$$c(u_1, \ldots, u_d) = \frac{f\Big(F_1^{-1}(u_1), \ldots, F_d^{-1}(u_d)\Big)}{\prod_{j=1}^d f_i\Big(F_j^{-1}(u_j)\Big)}.$$

In the above expressions, we can use arbitrary parametric distribution functions F to construct parametric copula families.

Example 2.4 (Gaussian and t-copulas). *Let Φ_Γ, be the cdf of a d-dimensional vector following a multivariate normal distribution with zero means, unit variances and correlation matrix Γ. Further, denote Φ as the univariate standard normal cdf. The Gaussian copula with parameter matrix Γ is given by*

$$C_\Gamma^{Gauss}(u_1, \ldots, u_d) = \Phi_\Gamma\Big(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)\Big).$$

Similarly, let $t_{\nu,\Gamma}$, be the cdf of a d-dimensional vector following a multivariate t-distribution with zero mean and association matrix Γ and degrees-of-freedom parameter ν. Denote further t_ν the univariate cdf of a t-distribution with degrees-of-freedom parameter ν. The Student or t-copula with parameters ν, Γ is given by

$$C_\Gamma^t(u_1, \ldots, u_d) = t_{\nu,\Gamma}\Big(t_\nu^{-1}(u_1), \ldots, t_\nu^{-1}(u_d)\Big).$$

In both cases, it can be shown that the use of non-zero means and non-unit variances does not change the copula. In the bivariate case a single parameter ρ is sufficient, since the correlation matrix is fully determined by one off-diagonal entry. The Gauss and t-copulas are the most prominent instances of the class of elliptical copulas. This class arises as the dependency structure underlying elliptical multivariate distributions.

Example 2.5 (Gaussian mixture copulas). *The same procedure also works for more complicated distributions such as mixtures. For instance, let us consider a two-fold Gaussian mixture distribution with mean vectors $\boldsymbol{\mu}, \mathring{\boldsymbol{\mu}}$ and covariance matrices $\Sigma, \mathring{\Sigma}$. Denote $\Phi_{\boldsymbol{\mu},\Sigma}, \Phi_{\mathring{\boldsymbol{\mu}},\mathring{\Sigma}}$ as the corresponding cdfs. For a mixing probability $\alpha \in (0, 1)$, we can write the Gaussian mixture cdf as*

$$\Psi(x_1, \ldots, x_d) = \alpha\Phi_{\boldsymbol{\mu},\Sigma}(x_1, \ldots, x_d) + (1 - \alpha)\Phi_{\mathring{\boldsymbol{\mu}},\mathring{\Sigma}}(x_1, \ldots, x_d).$$

By defining Φ_{μ,σ^2} as the univariate Gaussian cdf with mean μ and variance σ^2, also the univariate margins are normal mixture distributions given by

$$\Psi_i(x_i) = \alpha\Phi_{\mu_i,\Sigma_{ii}}(x_i) + (1-\alpha)\Phi_{\mathring{\mu}_i,\mathring{\Sigma}_{ii}}(x_i), \quad \textit{for all } i = 1,\ldots,d.$$

Now we have everything we need to define the Gaussian mixture copula with parameters $\boldsymbol{\mu}, \mathring{\boldsymbol{\mu}}, \Sigma, \mathring{\Sigma}, \alpha$ via

$$C^{GM}(u_1,\ldots,u_d) = \Psi\Big(\Psi_1^{-1}(u_1),\ldots,\Psi_d^{-1}(u_d)\Big).$$

The Gaussian mixture copula allows for very irregular shapes. Its construction will be illustrated in more detail after introducing some tools for visualization in Section 2.1.3. It is a highly flexible family, but also has a lot of parameters. The simplest case of a bivariate two-fold mixture already has eleven parameters. This makes it prone to overfitting the data when estimating copulas on small or moderate sample sizes.

Archimedean copulas

A second important and rich collection of copula families is the class of so-called *Archimedean copulas*. For simplicity, we will only consider the bivariate case here and refer the reader to Nelsen (2006) for a more general treatment. Let $\phi : [0,1] \to [0,\infty]$ be continuous, strictly monotonic decreasing, convex and satisfy $\phi(1) = 0$. Define further ϕ^- as the generalized inverse of ϕ, i.e.

$$\phi^-(y) := \inf\{x \in [0,\infty] : \phi(x) \geq y\}, \quad y \in [0,1].$$

Then,

$$C(u_1,u_2) = \phi^-\Big(\phi(u_1) + \phi(u_2)\Big)$$

is a proper copula function and called an *Archimedean copula*.

Example 2.6 (Frank copula). *For $\phi_\theta(x) = -\log\Big(\frac{\exp(-\theta x)-1}{\exp(-\theta)-1}\Big)$, with $\theta \in \mathbb{R}\setminus\{0\}$, we get*

$$C_\theta^{Frank}(u_1,u_2) = -\frac{1}{\theta}\log\left(1 + \frac{\Big(\exp(-\theta u_1)-1\Big)\Big(\exp(-\theta u_2)-1\Big)}{\exp(-\theta)-1}\right).$$

Example 2.7 (Gumbel copula). *For $\phi_\theta(x) = \Big(-\log(x)\Big)^\theta$, with $\theta \in [1,\infty)$, we get*

$$C_\theta^{Gumbel}(u_1,u_2) = \exp\left\{-\left[\Big(-\log(u_1)\Big)^\theta + \Big(-\log(u_2)\Big)^\theta\right]^{1/\theta}\right\}.$$

Example 2.8 (Clayton copula). *For $\phi_\theta(x) = \Big(x^{-\theta} - 1)/\theta$, with $\theta \in (0,\infty)$, we get*

$$C_\theta^{Clayton}(u_1,u_2) = \Big(u_1^{-\theta} + u_2^{-\theta} - 1\Big)^{-1/\theta}.$$

Example 2.9 (Joe copula). *For* $\phi_\theta(x) = -\log\Big(1-(1-x)^\theta\Big)$*, with* $\theta \in [1,\infty)$*, we get*

$$C_\theta^{Joe}(u_1,u_2) = 1 - \Big((1-u_1)^\theta + (1-u_2)^\theta - (1-u_1)^\theta(1-u_2)^\theta\Big)^{1/\theta}.$$

Extreme value copulas

Another class of parametric copula families arises naturally in the context of multivariate extreme value theory (see e.g. Salvadori et al., 2007). *"Being the limits of copulas of componentwise maxima in independent random samples, extreme-value copulas can be considered to provide appropriate models for the dependence structure between rare events"* (Gudendorf and Segers, 2009). Formally, a copula C is called *extreme-value copula* when there exists another copula C^*, such that

$$C^*\Big(\sqrt[n]{u_1},\dots,\sqrt[n]{u_d}\Big)^n \to C(u_1,\dots,u_d), \qquad \text{as } n\to\infty.$$

More information on extreme-value copulas can be found in the two references given above. We will just consider one particular example.

Example 2.10 (Tawn copula). *The Tawn copula family was created as an extension of the Gumbel copula allowing for asymmetry in its components. It is a three-parameter family defined as*

$$C^{Tawn}_{(\theta,\alpha_1,\alpha_2)} = \exp\left\{\Big(\log(u_1)+\log(u_2)\Big)A\left(\frac{\log(u_2)}{\log(u_1u_2)}\right)\right\},$$

where

$$A(x) = (1-\alpha_1)x + (1-\alpha_2)(1-x) + \Big((\alpha_1(1-x))^\theta + (\alpha_2 x)^\theta\Big)^{1/\theta},$$

and $(\theta,\alpha_1,\alpha_2)\in(1,\infty)\times[0,1]^2$*. For* $\alpha_1=\alpha_2=1$*, we recover the Gumbel copula; whenever* $\alpha_1\neq\alpha_2$ *it will be asymmetric in its components.*

2.1.2 Dependence measures

In this section we present some of the most popular dependence measures for bivariate random vectors (X,Y). Their goal is to summarize the strength (and direction) of dependence in just one number.

The dependence measure that is most widely used is the *Pearson correlation coefficient*

$$\mathrm{Corr}(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_X\sigma_Y}, \quad \text{provided } \sigma_X\sigma_Y \neq 0,$$

where σ_X and σ_Y are the standard deviations of X and Y respectively. We have $\mathrm{Corr}(X,Y)\in[-1,1]$, and $\mathrm{Corr}(X,Y)=0$ in case of independence. Generally speaking, it measures the strength of linear dependence. Its popularity is caused by the widespread use of multivariate normal models, where dependence is strictly linear and all information on dependence is contained in this number. For general multivariate distributions however, it reveals some unfavorable features. Foremost, the Pearson correlation depends on the marginal distributions and is thus not scale-free. As a consequence, random vectors (X, Y) whose copula corresponds to perfect dependence do not necessarily have $\operatorname{Corr}(X, Y) \in \{-1, 1\}$. Embrechts (2009) gives a nice example, where he shows that the correlation of two log-normal random variables $X \sim \mathcal{LN}(0, 1)$ and $Y \sim \mathcal{LN}(0, 16)$ is restricted to the interval $[-0.00025, 0.01372]$. A naive view on these numbers would lead to very misleading conclusions with possibly horrendous consequences in application.

Concordance measures

A more adequate concept to measure dependence is *concordance*. Let $(x_i, y_i)_{i=1,\dots,n}$ be *iid* observations of a random vector (X, Y). We call the couple (x_i, y_i) and (x_j, y_j) for $i \neq j$ *concordant* if

$$(x_i - x_j)(y_i - y_j) > 0$$

and *discordant* if

$$(x_i - x_j)(y_i - y_j) < 0.$$

Concordance means that either $x_i > x_j$ and $y_i > y_j$ or $x_i < x_j$ and $y_i < y_j$. Hence, it describes a type of conformable behavior of the observations x_i and y_i. A similar characterization for discordance shows that it describes opposing behavior.

There are several ways to construct dependence measures based on the idea of concordance and discordance. These measures are usually directly related to the copula of a random vector and do not depend on its marginal distributions. One such measure was introduced by Kendall (1938) and is called *Kendall's rank correlation coefficient* or simply *Kendall's τ*. It is defined as the probability of concordance minus the probability of discordance.

Definition 2.2. *Let $(\widetilde{X}, \widetilde{Y})$ be an independent copy of the random vector (X, Y). Then,* ***Kendall's $\boldsymbol{\tau}$*** *is defined as*

$$\tau(X, Y) = \mathrm{P}\Big((X - \widetilde{X})(Y - \widetilde{Y}) > 0\Big) - \mathrm{P}\Big((X - \widetilde{X})(Y - \widetilde{Y}) < 0\Big).$$

In the following Lemma, we summarize the most important properties of Kendall's τ (c.f. Embrechts et al., 2003).

Lemma 2.2 (Properties of Kendall's τ). *Let C be the copula of an arbitrary random vector (X, Y).*

- $\tau(X, Y) \in [-1, 1]$.
- $C = M \;\Leftrightarrow\; \tau(X, Y) = 1$, *i.e. Kendall's τ is one if and only if X and Y are perfectly positively dependent.*

Family	Kendall's τ
Gaussian	$2/\pi \arcsin(\rho)$
Student	$2/\pi \arcsin(\rho)$
Frank	$1 + 4(D(\theta) - 1)/\theta$, with $D(\theta) = \int_0^\theta \frac{x/\theta}{e^x - 1} dx$
Gumbel	$1 - 1/\theta$
Clayton	$\theta/(\theta + 2)$
Joe	$1 + \left(-2 + 2\gamma + 2\log(2) + \Psi(\frac{1}{\theta}) + \Psi(\frac{2+\theta}{2\theta}) + \theta\right)/(\theta - 2)$, with Euler's constant $\gamma \approx 0.57721$ and Digamma-function Ψ
Tawn	$\int_0^1 \frac{x(1-x)}{A(x)} dA'(x)$

Table 2.1: Kendall's τ in terms of the copula parameters of selected families.

- $C = W \Leftrightarrow \tau(X, Y) = -1$, *i.e. Kendall's τ is minus one if and only if X and Y are perfectly negatively dependent.*
- $C = \Pi \Rightarrow \tau(X, Y) = 0$, *i.e. Kendall's τ is zero when X and Y are independent.*
- $\tau(X, Y) = 4 \int_0^1 \int_0^1 C(u, v) dC(u, v)$. *Hence, Kendall's τ does only depend on the copula and not on the marginal distributions.*

For many copula families, there is a one-to-one correspondence between the copula parameter and Kendall's τ (c.f. Table 2.1).

A non-parametric estimator of Kendall's τ can be obtained by estimating the probabilities of concordance and discordance by the empirical proportions of concordant and discordant pairs in a sample.

Definition 2.3. *Let $\left(x^{(i)}, y^{(i)}\right), i = 1, \ldots, n$, be iid samples of a continuous random vector (X, Y). The* ***empirical Kendall's τ*** *is given by*

$$\hat{\tau}_n(X, Y) = \frac{N_C - N_D}{N_C + N_d},$$

where N_C is the number of concordant and N_D the number of discordant pairs amongst all realizations of $\left(x^{(i)}, y^{(i)}\right)_{i=1,\ldots,n}$.

There are other popular dependence measures based on the concept of concordance like Spearman's ρ (Spearman, 1904) or Blomqvist's β (Blomqvist, 1950). They will not be used in the remainder of this thesis, so we omit definitions and proceed to a different type of dependence.

Tail dependence

Another interesting question regards the dependence between extreme values of two random variables. Let $X \sim F$ and $Y \sim G$ and assume we are interested in the

Family	λ_U	λ_L
Gaussian	0	0
Student	$2t_{\nu+1}\left(-\sqrt{\nu+1}\sqrt{\frac{1-\rho}{1+\rho}}\right)$	$2t_{\nu+1}\left(-\sqrt{\nu+1}\sqrt{\frac{1-\rho}{1+\rho}}\right)$
Frank	0	0
Gumbel	$2-2^{1\theta}$	0
Clayton	0	$2^{-1/\theta}$
Joe	$2-2^{1\theta}$	0
Tawn	$(\alpha_1+\alpha_2)-(\alpha_1^\theta+\alpha_2^\theta)^{1/\theta}$	0

Table 2.2: Coefficients of upper and lower tail-dependence in terms of the copula parameters of selected families.

dependence between extremely high values of them. We could look at the probability that X is big conditional on Y being big, i.e.

$$\mathrm{P}\Big(X > F^{-1}(u) | Y > G^{-1}(u)\Big),$$

for some value u close to but less than one. The *upper tail-dependence coefficient* $\lambda_U(X,Y)$ is defined as the limit of the above probability as $u \nearrow 1$, provided it exists. Similarly, we can define the *lower tail-dependence coefficient* $\lambda_L(X,Y)$ as

$$\lambda_L(X,Y) = \lim_{u \searrow 0} \mathrm{P}\Big(X < F^{-1}(u) | Y < G^{-1}(u)\Big).$$

It can be shown that tail-dependence is a property of the copula and independent of marginal distributions. A definition in terms of the copula is the following (c.f. Joe, 1997).

Definition 2.4. *Let C be the copula of a random vector (X,Y). Then the* ***coefficients of upper and lower tail-dependence*** *are given by*

$$\lambda_U(X,Y) = \lim_{u \nearrow 1} \frac{1-2u+C(u,u)}{1-u},$$
$$\lambda_L(X,Y) = \lim_{u \searrow 0} \frac{C(u,u)}{u}.$$

We say that X and Y are *upper/lower tail-dependent* whenever λ_U resp. λ_L exceeds zero. For many parametric families, there is an explicit relationship between the copula parameter and the tail-dependence coefficients (see Table 2.2).

Rotated copulas

Note that some of the presented parametric families have a restricted parameter space and only allow for positive dependence. Also, some of them only allow for either upper or lower tail dependence. Rotation of the copula is a convenient way to extend those families to give more flexibility. In the following definition we use *counter-clockwise* rotation.

Definition 2.5. *Let $c(u,v)$ be a copula density. The densities of rotated versions of this copula are given as follows:*

- *90 degrees rotation:* $c_{90}(u,v) := c(1-u,v)$
- *180 degrees rotation:* $c_{180}(u,v) := c(1-u,1-v)$
- *270 degrees rotation:* $c_{270}(u,v) := c(u,1-v)$.

2.1.3 Visualization

In statistics it is often very helpful to have a good visualization of the objects of interest. In the following we will explain common methods that proved beneficial for the visualization of bivariate copulas.

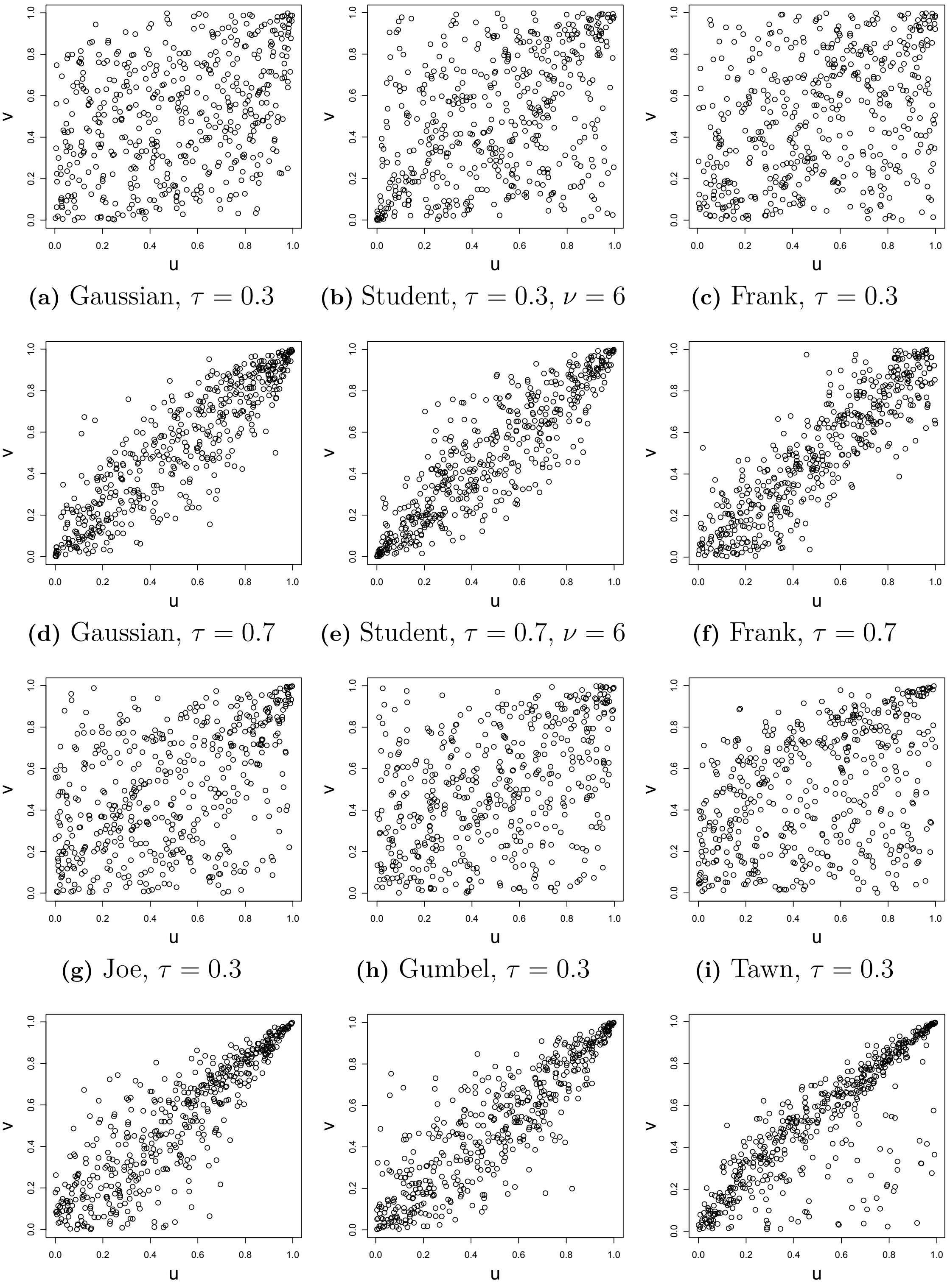
The first and most popular tool is the *scatter plot* of copula samples where each observation is represented by a point in the unit square. In Figure 2.1 this is shown for simulated data of selected parametric families. Parameters were chosen to give scenarios of weak ($\tau = 0.3$) and strong ($\tau = 0.7$) dependence. The sample size is $n = 500$. The strength of dependence is clearly distinguishable and asymmetries can be detected for the Clayton, Gumbel and Tawn copulas. In data analysis scatter plots often give a first indication about the appropriateness of a parametric model — despite the difficulties one might have to distinguish between some of the families.

The most obvious way to visualize a bivariate copula density on the other hand, is to plot it as a surface in the three-dimensional space. In Figure 2.2 we give these so-called *perspective plots* for the copulas considered in Figure 2.1. We can identify the general shape of the densities as well as high- and low density regions. Still, some of the families (e.g. Gaussian and t-copula) are very hard to distinguish. There is also one problem with some of the copulas as we see tails extending over the bounds of the plots. In fact, the density values of many copula families tend to infinity at some corner of the unit square. Unfortunately, this constitutes a rule rather than an exception and the problem becomes even more prominent when the strength of dependence is increased. In those cases, the perspective plot is a somewhat inconvenient tool for visualization.

A very powerful tool that overcomes this issue is the *marginal normal contour plot*. Instead of looking at the original copula density with uniform margins, the copula is coupled with standard normal margins resulting in a meta-copula density defined as

$$f(x,y) = c\Big(\Phi(x), \Phi(y)\Big)\phi(x)\phi(y).$$

The contours of this transformed density form the marginal normal contour plot. It is given for all the previous examples in Figure 2.3. Each copula family shows a characteristic shape. Furthermore, the strength of dependence is identifiable by the width of the contours, and tail-dependence is usually indicated by a spiky shape in the tail.



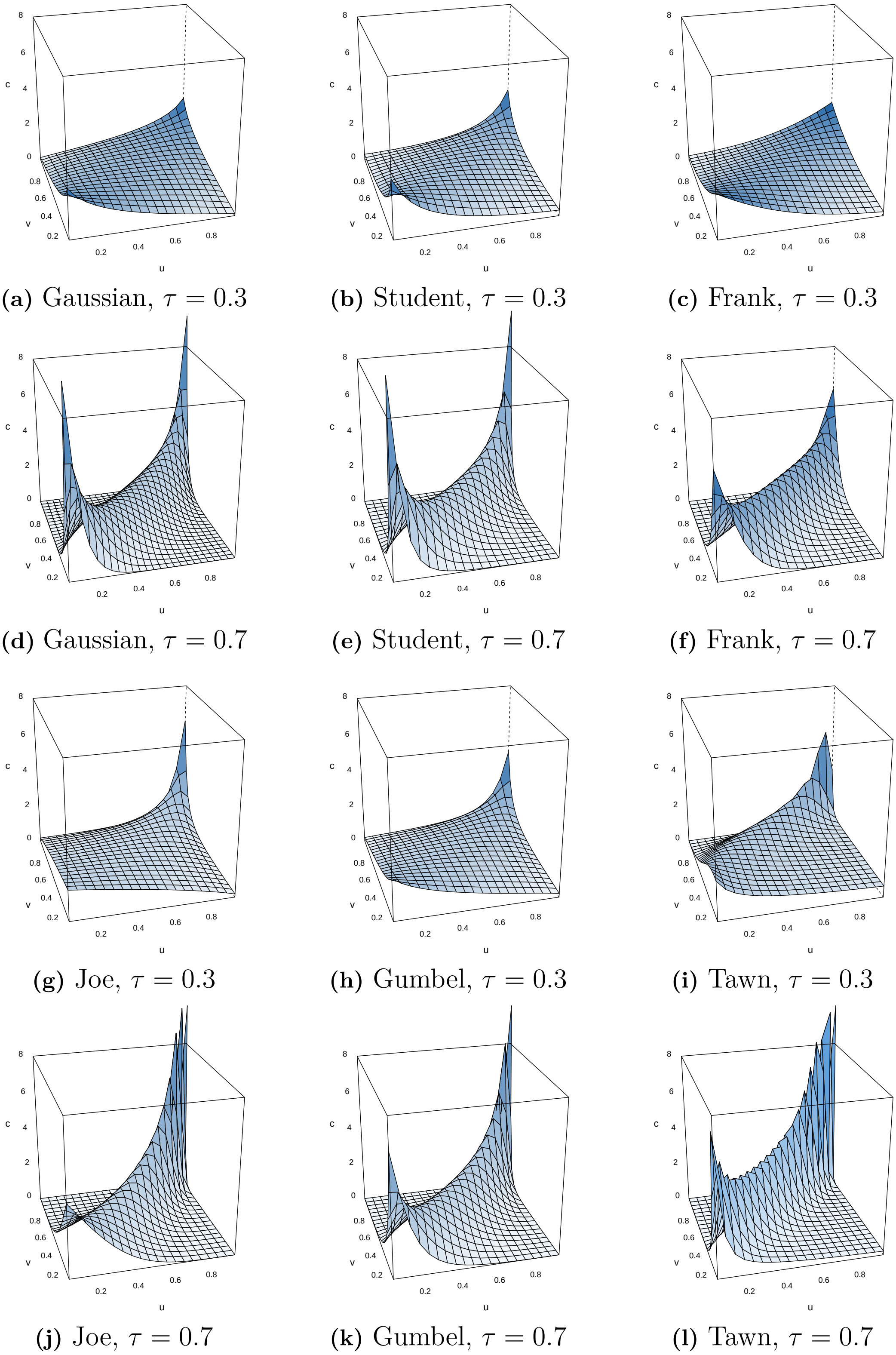
(**a**) Gaussian, $\tau = 0.3$ (**b**) Student, $\tau = 0.3$, $\nu = 6$ (**c**) Frank, $\tau = 0.3$

(**d**) Gaussian, $\tau = 0.7$ (**e**) Student, $\tau = 0.7$, $\nu = 6$ (**f**) Frank, $\tau = 0.7$

(**g**) Joe, $\tau = 0.3$ (**h**) Gumbel, $\tau = 0.3$ (**i**) Tawn, $\tau = 0.3$

(**j**) Joe, $\tau = 0.7$ (**k**) Gumbel, $\tau = 0.7$ (**l**) Tawn, $\tau = 0.7$

Figure 2.1: Scatter plots of simulated copula data for weak ($\tau = 0.3$) and strong ($\tau = 0.7$) dependence. Sample size is $n = 500$.



(a) Gaussian, $\tau = 0.3$

(b) Student, $\tau = 0.3$

(c) Frank, $\tau = 0.3$

(d) Gaussian, $\tau = 0.7$

(e) Student, $\tau = 0.7$

(f) Frank, $\tau = 0.7$

(g) Joe, $\tau = 0.3$

(h) Gumbel, $\tau = 0.3$

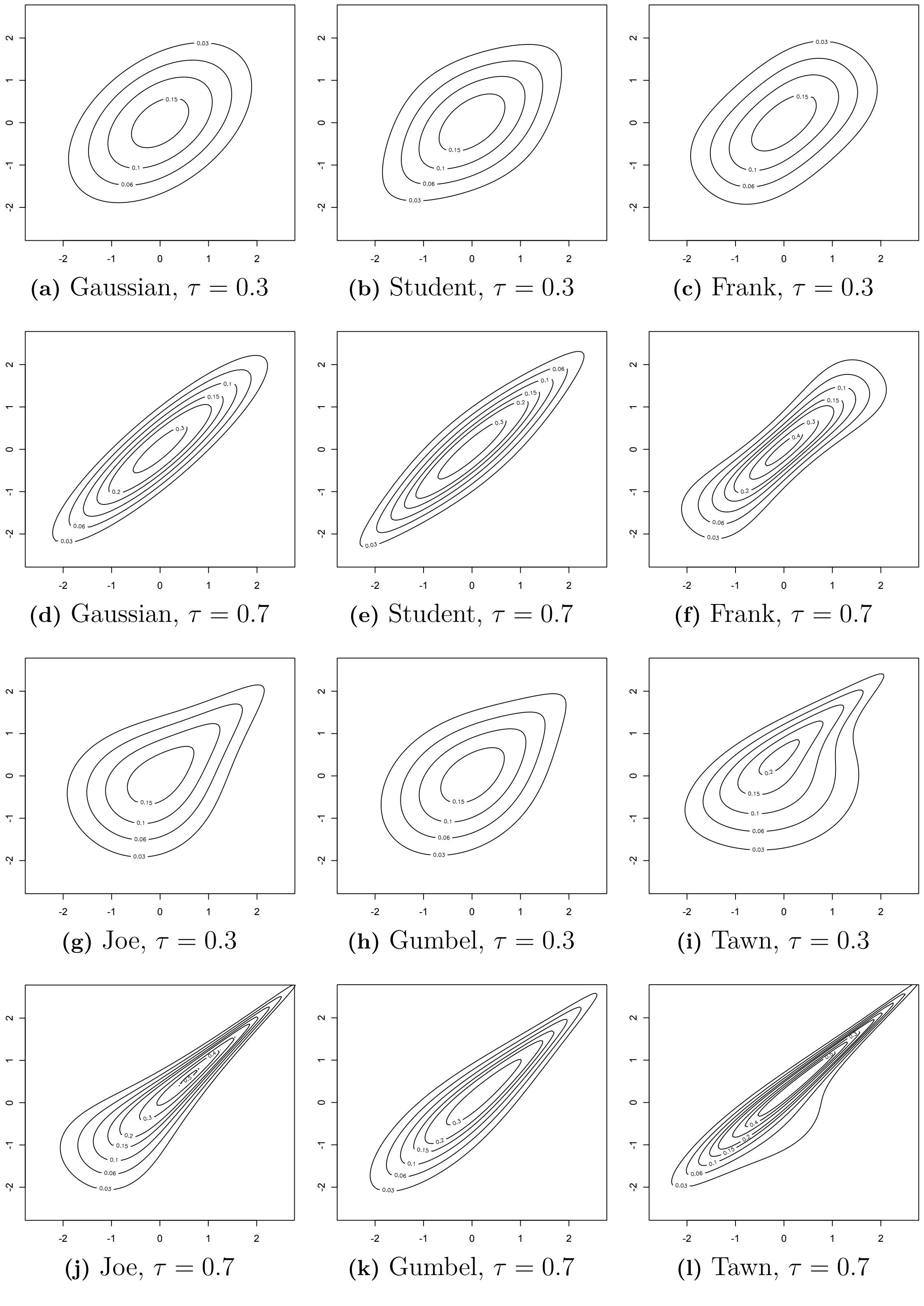
(i) Tawn, $\tau = 0.3$

(j) Joe, $\tau = 0.7$

(k) Gumbel, $\tau = 0.7$

(l) Tawn, $\tau = 0.7$

Figure 2.2: Perspective plots of copula densities for weak ($\tau = 0.3$) and strong ($\tau = 0.7$) dependence.



(a) Gaussian, $\tau = 0.3$ **(b)** Student, $\tau = 0.3$ **(c)** Frank, $\tau = 0.3$

(d) Gaussian, $\tau = 0.7$ **(e)** Student, $\tau = 0.7$ **(f)** Frank, $\tau = 0.7$

(g) Joe, $\tau = 0.3$ **(h)** Gumbel, $\tau = 0.3$ **(i)** Tawn, $\tau = 0.3$

(j) Joe, $\tau = 0.7$ **(k)** Gumbel, $\tau = 0.7$ **(l)** Tawn, $\tau = 0.7$

Figure 2.3: Marginal normal contour plots of copula densities for weak ($\tau = 0.3$) and strong ($\tau = 0.7$) dependence.

Example 2.11 (Gaussian mixture copula, continued). *Let us shed some light on the construction and appearance of Gaussian mixture copulas. We will illustrate two specific scenarios that will be used in a simulation study in Section 3.6. The parameters give a scenario with weak ($\tau = 0.3$) and one with strong ($\tau = 0.7$) dependence and are specified as follows:*

τ	*Parameters*
0.3	$\boldsymbol{\mu} = \left(\begin{smallmatrix} 2 \\ 2 \end{smallmatrix}\right), \mathring{\boldsymbol{\mu}} = \left(\begin{smallmatrix} 6 \\ 1.6 \end{smallmatrix}\right), \Sigma = \left(\begin{smallmatrix} 1 & 0.9 \\ 0.9 & 1 \end{smallmatrix}\right), \mathring{\Sigma} = \left(\begin{smallmatrix} 4 & 2 \\ 2 & 4 \end{smallmatrix}\right), \alpha = 0.65$
0.7	$\boldsymbol{\mu} = \left(\begin{smallmatrix} 2 \\ 2 \end{smallmatrix}\right), \mathring{\boldsymbol{\mu}} = \left(\begin{smallmatrix} 6 \\ 6 \end{smallmatrix}\right), \Sigma = \left(\begin{smallmatrix} 1.35 & 0.8 \\ 0.8 & 1 \end{smallmatrix}\right), \mathring{\Sigma} = \left(\begin{smallmatrix} 2.1 & 1.4 \\ 1.4 & 2.1 \end{smallmatrix}\right), \alpha = 0.65$

In Figure 2.4 (a) there are scatter-, perspective and contour plots of the Gaussian mixture distribution with weak dependence. The distribution is bimodal with each mode being centered at $\boldsymbol{\mu}$ and $\mathring{\boldsymbol{\mu}}$ respectively. When going to the copula level in (b), the two-part structure remains. Although, the locations of the centers as well as the overall structure of the density change. Analogue observations can be made for the scenario with strong dependence in Figures 2.4 (c) and (d). Note also that in both scenarios the tails of the mixture copula appear to be bounded. We want to stress, however, that this does not hold for every Gaussian mixture copula.

2.1.4 Parametric copula estimation

Assume we are given *iid* samples $\left(u_1^{(i)}, \dots, u_d^{(i)}\right), i = 1, \dots, n$, of a random vector $(U_1, \dots, U_d) \sim C$ and want to estimate the copula C. The most popular approach in parametric models is to assume a particular family and estimate its parameter(s) by *maximum likelihood.*

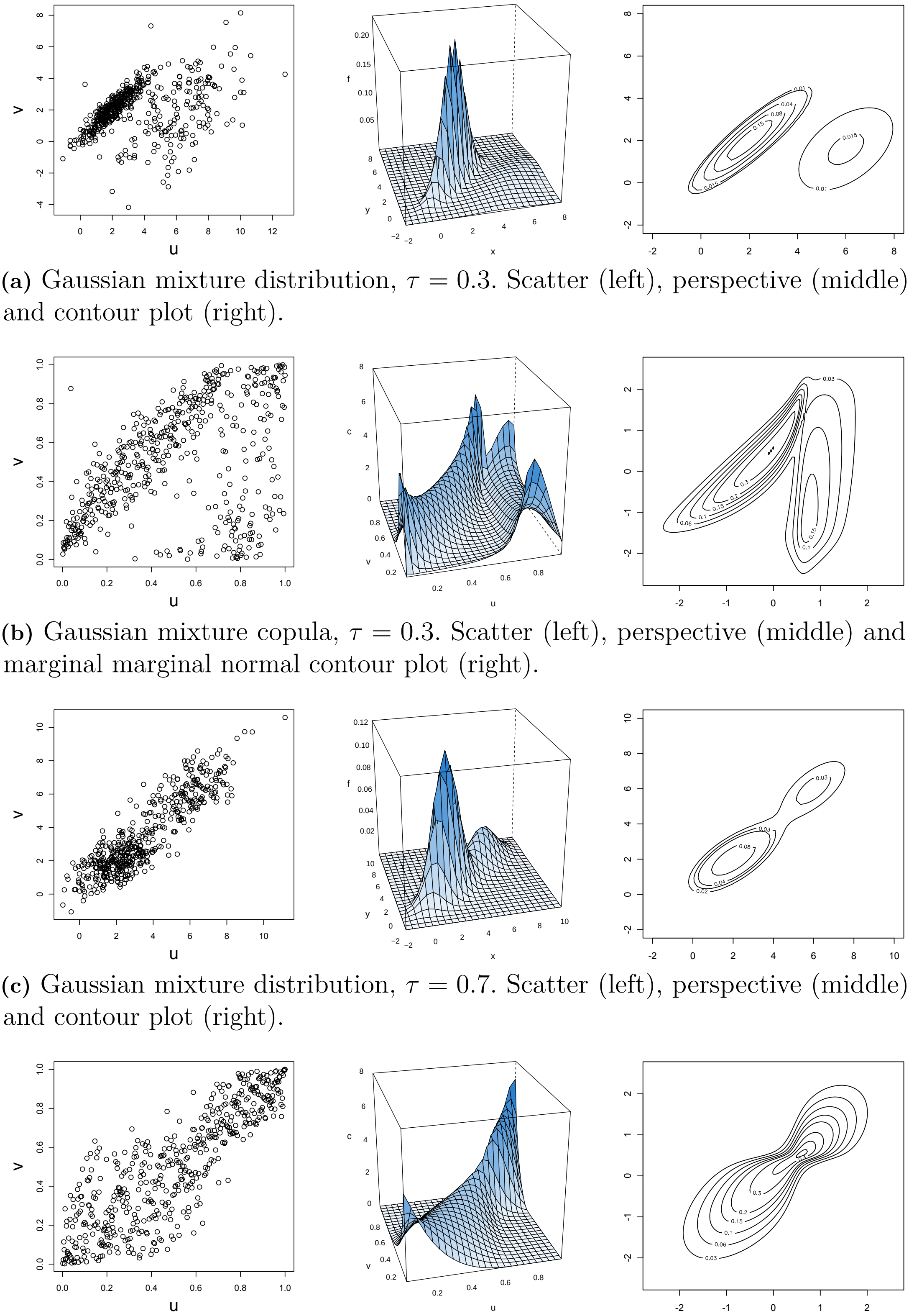
Definition 2.6. *Let $(U_1, \dots, U_d) \sim C_{\boldsymbol{\theta}}^{(\cdot)}$, where $\boldsymbol{\theta} \in \Theta$, and $\Theta \subset \mathbb{R}^p$, $p \in \mathbb{N}$, is the family's parameter space. Denote further $c_{\boldsymbol{\theta}}^{(\cdot)}$ as the density of $C_{\boldsymbol{\theta}}^{(\cdot)}$. The* ***maximum likelihood estimator*** *of the parameter vector $\boldsymbol{\theta}$ is defined as*

$$\widehat{\boldsymbol{\theta}}_n^{MLE} = \underset{\boldsymbol{\theta} \in \Theta}{\arg\max} \prod_{i=1}^{n} c_{\boldsymbol{\theta}}^{(\cdot)}\left(u_1^{(i)}, \dots, u_d^{(i)}\right).$$

For bivariate estimation of one-parametric families, an alternative estimation approach is motivated by the one-to-one relationship of Kendall's τ and the parameter for some families. By exploiting this relationship we can obtain a parameter estimate based on the empirical Kendall's τ.

Definition 2.7. *Let $(U_1, U_2) \sim C_{\theta}^{(\cdot)}$, where $\theta \in \Theta$, where $\Theta \subset \mathbb{R}$ is the family's parameter space. Assume further that there exists a bijective function $\psi : \Theta \rightarrow [-1, 1]$, such that $\psi(\theta) = \tau(U_1, U_2)$. An estimator of θ obtained by* ***inversion of empirical Kendall's $\boldsymbol{\tau}$*** *is defined as*

$$\widehat{\theta}_n^{itau} = \psi^{-1}\left(\widehat{\tau}_n(U_1, U_2)\right).$$



(a) Gaussian mixture distribution, $\tau = 0.3$. Scatter (left), perspective (middle) and contour plot (right).

(b) Gaussian mixture copula, $\tau = 0.3$. Scatter (left), perspective (middle) and marginal marginal normal contour plot (right).

(c) Gaussian mixture distribution, $\tau = 0.7$. Scatter (left), perspective (middle) and contour plot (right).

(d) Gaussian mixture copula, $\tau = 0.7$. Scatter (left), perspective (middle) and marginal normal contour plot (right).

Figure 2.4: Exploratory plots for the Gaussian mixture distribution and the corresponding Gaussian mixture copula (see Example 2.11).

Whenever the parametric assumption is true, i.e. the true copula C belongs to the chosen family, both of the presented estimators are consistent and give very good convergence rates. One should be careful, however, since consistency is lost, when the model is misspecified.

Model selection

There is an immediate question arising from this issue: How to choose the parametric family? Usually, the parameter is estimated for several different parametric families and the best model is selected by considering information criteria. The most popular criteria are *Akaike's information criterion (AIC)* (Akaike, 1974)

$$AIC_n^{(\cdot)} := -2\sum_{i=1}^{n} \ln\left(c_{\widehat{\boldsymbol{\theta}}_n}^{(\cdot)}\left(u_1^{(i)}, \dots, u_d^{(i)}\right)\right) + 2p,$$

and the *Bayesian information criterion (BIC)* (Schwarz, 1978)

$$BIC_n^{(\cdot)} := -2\sum_{i=1}^{n} \ln\left(c_{\widehat{\boldsymbol{\theta}}_n}^{(\cdot)}\left(u_1^{(i)}, \dots, u_d^{(i)}\right)\right) + \log(n)p,$$

where p is the number of parameters of the family and $\widehat{\boldsymbol{\theta}}_n$ is a parameter estimate. The 'best' model is the one that minimizes the information criterion. When the number of parameters across the considered models is the same, both criteria select the model that gives the highest likelihood. Otherwise, both models give a penalty for increasing the number of parameters in order to avoid overfitting. For $n \geq 8$, BIC employs a higher penalty than AIC.

Pseudo-observations

In basically all real-life problems, we are not provided with samples $(u_1^{(i)}, \dots, u_d^{(i)})$ from a copula directly, but with samples $(x_1^{(i)}, \dots, x_d^{(i)})$ from some general multivariate distribution. In the spirit of Sklar's Theorem we can use the distribution function F_j of X_j and define the copula sample as $F_j(x_j^{(i)}) =: u_j^{(i)}$ for all $j = 1, \dots, d$, $i = 1, \dots, n$. The true marginal distributions F_j are usually unknown and have to be estimated first. A popular method in this context is to use the empirical distribution as an estimator of F_j. Copula samples that are obtained in this manner are usually called *pseudo-samples* or *pseudo-observations*. The attribute 'pseudo' comes from the fact that they aren't truly samples from the copula, but estimated copula samples instead.

Definition 2.8. *Let $(X_1, \dots, X_d)$ be a random vector with marginal distributions $F_1, \dots, F_d$. Provided an iid sample $(x_1^{(i)}, \dots, x_d^{(i)})_{i=1,\dots,n}$ of this random vector, we take the empirical distribution*

$$\widehat{F}_{j,n}(x_j) = \frac{1}{n+1} \sum_{i=1}^{n} \mathbb{1}\Big(x_j^{(i)} \le x_j\Big),$$

as an estimator of F_j for all $j = 1, \dots, d$. ***Pseudo-observations*** *of the random vector $(U_1, \dots, U_d) = \big(F_1(X_1), \dots, F_d(X_d)\big)$ are then defined as*

$$\Big(\widehat{u}_1^{(i)}, \dots, \widehat{u}_d^{(i)}\Big) = \bigg(\widehat{F}_{1,n}\Big(x_1^{(i)}\Big), \dots, \widehat{F}_{d,n}\Big(x_d^{(i)}\Big)\bigg), \qquad \text{for all } i = 1, \dots, n.$$

2.1.5 Vine copulas

Vine copula models follow the idea that any d-dimensional copula density can be decomposed into a product of $d(d-1)/2$ bivariate (conditional) copula densities (Bedford and Cooke, 2001). Equivalently, we can build arbitrary d-dimensional copula densities by using only bivariate building blocks. Therefore, vine copulas are also called *pair-copula construction (PCC)*. Recommended readings are Aas et al. (2009), Kurowicka and Joe (2010) and Stöber and Czado (2012).

An exemplary PCC of a 3-dimensional copula density corresponding to a random vector $(U_1, U_2, U_3) \sim C$ is

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

Here, $C_{1,2}$ is the copula of (U_1, U_2), $C_{2,3}$ is the copula of (U_2, U_3) and $C_{1,3;2}(\cdot, \cdot; u_2)$ is the copula of $\Big(C_{1|2}(U_1|u_2), C_{3|2}(U_3|u_2)\Big)$, where $C_{1|2}(\cdot|u_2)$ and $C_{3|2}(\cdot|u_2)$ are the conditional distribution functions of $U_1|U_2 = u_2$ and $U_3|U_2 = u_2$ respectively. Note that, in general, the conditional copula $C_{1,3;2}$ may be different for each value u_2. As such a model can be very complex, one often assumes that the dependence on the conditioning variables (here u_2) can be ignored and, thus, only unconditional copulas are involved. In this case we speak of the *simplifying assumption* or a simplified PCC. However, the arguments $C_{1|2}(u_1|u_2)$ and $C_{3|2}(u_3|u_2)$ may still depend on the specific value of u_2. A discussion of the appropriateness of this assumption can be found in Hobæk Haff et al. (2010), classes of copulas where the simplifying assumption is satisfied are given in Stöber et al. (2012), estimation of non-simplified PCCs is tackled by Acar et al. (2012). In this thesis we will always assume that the simplifying assumption is valid.

Regular vine trees

A crucial observation is that the decomposition of the density is not unique. In higher dimensions a myriad of different decompositions is possible. Bedford and Cooke (2002)

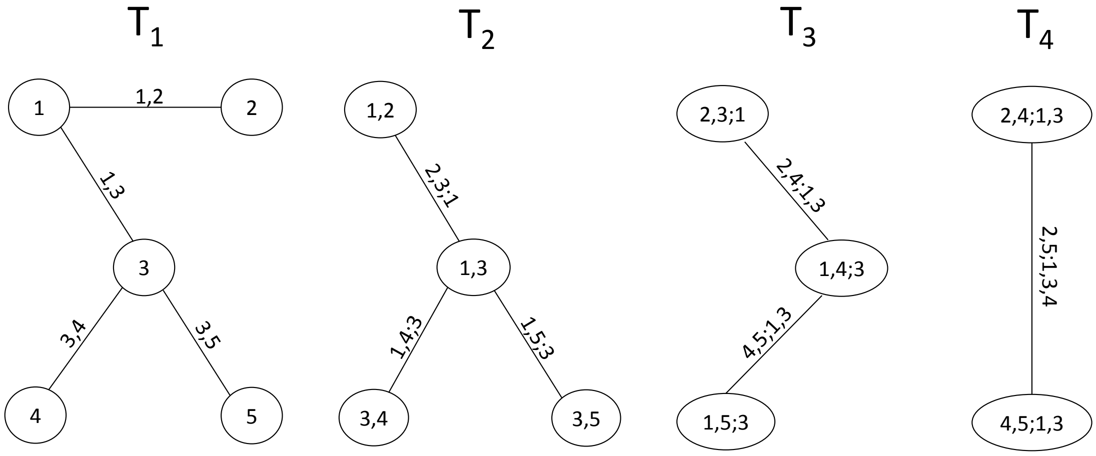


Figure 2.5: Example of a regular vine tree sequence.

discuss a graphical method to organize the structure of a d-dimensional vine copula in terms of linked trees $T_k = (V_k, E_k)$, $k = 1, \dots, d-1$. In this representation V_k is the set of nodes and E_k is the set of edges in tree T_k (for fundamentals of graph theory, see e.g. Gross and Yellen, 2005).

Definition 2.9. *A sequence $\mathcal{V} := (T_1, \dots, T_{d-1})$ of trees is called a **regular vine (R-vine) tree sequence** on d elements if the following conditions are satisfied:*

(i) T_1 is a tree with nodes $V_1 = \{1, \dots, d\}$ and edges E_1.

(ii) For $k \geq 2$, T_k is a tree with nodes $V_k = E_{k-1}$ and edges E_k.

*(iii) (*Proximity condition*) Whenever two nodes a, b of T_{k+1} are joined by an edge, the edges a, b in tree T_k must share a common node.*

An example of an R-vine tree sequence for $d = 5$ is given in Figure 2.5. For the annotation of the edges in each tree we use a specific scheme. For each $e \in E_k$, $k = 1, \dots, d-1$ we define the *complete union* of e as

$$Q(e) := \Big\{ i \in \mathbb{N} | \exists e_1 \in E_1, \dots e_{k-1} \in E_{k-1} : i \in e_1 \in \dots, \in e_{k-1} \in e \Big\}.$$

We define the *conditioning set* of an edge $e = \{a, b\}$ as $D_e := Q_a \cap Q_b$ and the *conditioned sets* as $a_e := Q(a) \setminus D_e, b_e := Q(b) \setminus D_e$. Finally, we annotate each edge by $\{a_e, b_e; D_e\}$. Note that a_e and b_e consist of just a single element each.

In layman's terms this can be understood as follows: To annotate an edge $e \in T_k$, $k \geq 2$, we look at all the numbers appearing in the annotation of the two nodes in T_k that are joined by e. All numbers that appear in both nodes will be the conditioning set and the two numbers that appear in just one of the nodes will give the two conditioned sets.

Regular vine copulas

To relate such a sequence of trees to a PCC, we identify each edge in the sequence with a bivariate copula.

Definition 2.10. *A copula C corresponding to a random vector $(U_1, \dots, U_d)$, where $U_j \sim U[0,1]$, $j = 1, \dots, d$, is called a* ***regular vine (R-vine) copula****, if there is a tuple $(\mathcal{V}, \mathcal{C})$ such that*

(i) $\mathcal{V}$ is a regular vine tree sequence on d elements.

(ii) $\mathcal{C} = \Big\{ C_e | e \in E_k, k = 1, \dots, d-1 \Big\}$, where C_e is a bivariate copula.

(iii) Each $e \in E_k$, $k = 1, \dots, d-1$, can be identified as $\{a_e, b_e; D_e\}$, and C_e is the copula corresponding to $(U_{a_e}, U_{b_e}) | (U_l)_{l \in D_e} = (u_l)_{l \in D_e}$.

Note that the set notation $e = \{a, b\}$ does not induce any ordering of its elements. Therefore, the order of the indices a_e, b_e is not uniquely determined and can be chosen arbitrarily. We should, however, pay attention to this order when the copula $C_{a_e,b_e;D_e}$ is not symmetric in its arguments. In this case it would be more appropriate to use directed graphs, but we will stay with our notation for simplicity.

Proposition 2.3. *Let C be an R-vine copula corresponding to the tuple $(\mathcal{V}, \mathcal{C})$. If all copulas in $\mathcal{C}$ admit a density, the density of C can be written as*

$$c(\boldsymbol{u}) = \prod_{k=}^{d-1} \prod_{e \in E_k} c_{a_e,b_e;D_e}\Big(C_{a_e|D_e}(u_{a_e} | \boldsymbol{u}_{D_e}), C_{b_e|D_e}(u_{b_e} | \boldsymbol{u}_{D_e}) \Big),$$

where $\boldsymbol{u}_{D_e} = (u_j)_{j \in D_e}$ is a subvector of $\boldsymbol{u} = (u_1, \dots, u_d)$ and $C_{j_e|D_e}$ is the conditional distribution of $U_{j_e} | \boldsymbol{U}_{D_e} = \boldsymbol{u}_{D_e}$, for $j_e \in \{1, \dots, d\}$.

In order to explicitly write down the full density of a regular vine copula it is often convenient to introduce the short notation

$$u_{j_e|D_e} := C_{j_e|D_e}(u_{j_e} | \boldsymbol{u}_{D_e}).$$

Example 2.12. *The density of an R-vine copula corresponding to the tree sequence in Figure 2.5 is*

$$\begin{aligned} c(u_1, \dots, u_5) = {} & c_{1,2}(u_1, u_2) \cdot c_{1,3}(u_1, u_3) \cdot c_{3,4}(u_3, u_4) \cdot c_{3,5}(u_3, u_5) \\ & \cdot c_{2,3;1}(u_{2|1}, u_{3|1}) \cdot c_{1,4;3}(u_{1|3}, u_{4|3}) \cdot c_{1,5;3}(u_{1|3}, u_{5|3}) \\ & \cdot c_{2,4;1,3}(u_{2|1,3}, u_{4|1,3}) \cdot c_{4,5;1,3}(u_{4|1,3}, u_{5|1,3}) \\ & \cdot c_{2,5;1,3,4}(u_{2|1,3,4}, u_{5|1,3,4}). \end{aligned}$$

The density of a regular vine copula involves conditional distributions of the form $C_{j_e|D_e}$, where $j_e \in \{a_e, b_e\}$. When D_e has more than one element, it is not immediately clear how to get those functions. Fortunately, they can be expressed as a recursive application of conditional distributions corresponding to bivariate copulas contained in $\mathcal{C}$. Because of the frequent appearance of such functions, they are given an own name.

Definition 2.11. *Let $U_1, U_2 \sim U[0,1]$ and C be the copula of (U_1, U_2). The* ***h-functions*** *corresponding to C are defined as*

$$\begin{aligned} h_{1|2}(u_1|u_2) = C_{1|2}(u_1|u_2) = \frac{\partial C(u_1,u_2)}{\partial u_2} = P(U_1 \le u_1 | U_2 = u_2), \\ h_{2|1}(u_2|u_1) = C_{2|1}(u_2|u_1) = \frac{\partial C(u_1,u_2)}{\partial u_1} = P(U_2 \le u_2 | U_1 = u_1). \end{aligned}$$

Now let $j'_e \in D_e$ be another index such that $C_{j_e,j'_e;D_e \setminus j'_e} \in \mathcal{C}$ and define $D'_e := D_e \setminus j'_e$. Then, we can write

$$\begin{aligned} C_{j_e|D_e}(u_{j_e}|\boldsymbol{u}_{D_e}) &= C_{j_e|j'_e;D'_e}\Big(C_{j_e|D'_e}(u_{j_e}|\boldsymbol{u}_{D'_e})|C_{j'_e|D'_e}(u_{j'_e}|\boldsymbol{u}_{D'_e})\Big) \\ &= h_{j_e|j'_e;D'_e}\Big(C_{j_e|D'_e}(u_{j_e}|\boldsymbol{u}_{D'_e})|C_{j'_e|D'_e}(u_{j'_e}|\boldsymbol{u}_{D'_e})\Big), \end{aligned} \tag{2.2}$$

where $h_{j_e|j'_e;D'_e}$ is the h-function corresponding to the random vector

$$\Big(C_{j_e|D'_e}(U_{j_e}|\boldsymbol{u}_{D'_e})|C_{j'_e|D'_e}(U_{j'_e}|\boldsymbol{u}_{D'_e})\Big). \tag{2.3}$$

Subsequently, the conditional distributions $C_{j_e|D'_e}$ and $C_{j'_e|D'_e}$ appearing in the arguments can be rewritten in an equal manner, then their arguments, and so on. Eventually, we end up with a chain of h-functions.

Example 2.13. *Consider an R-vine copula corresponding to the R-vine tree sequence given in Figure 2.5. We have*

$$C_{3|1,2}(u_3|u_1,u_2) = h_{3|2;1}\Big(h_{3|1}(u_3|u_1)\Big|h_{2|1}(u_2|u_1)\Big).$$

Estimation

Next, we want to discuss the estimation of a regular vine copula. In parametric models, estimates can be obtained by maximization of the full likelihood. As the number of parameters increases drastically with dimension, a sequential estimation approach is often used to find good starting values for the optimization. In this approach, only bivariate estimations are conducted. This method will later prove valuable for nonparametric estimation as well.

Definition 2.12. *Let C be an R-vine copula corresponding to the tuple $(\mathcal{V}, \mathcal{C})$ and assume we are given iid samples $(u_1^{(i)}, \dots, u_d^{(i)})_{i=1,\dots,n}$ from C. A* ***sequential estimate of an R-vine copula density*** *is obtained as follows:*

1. *For all $e \in E_1$, obtain estimates for c_{a_e,b_e}.*
2. *For $k = 2, \dots, d-1$:*

   For all $e \in E_k$ and $j = a_e, b_e$:

   (i) Let $j' \in D_e$ be another index such that $C_{j,j';D_e \setminus j'} \in \mathcal{C}$ and define $D'_e := D_e \setminus j'$.

   (ii) Based on the sample $\left(u_{j|D'_e}^{(i)}, u_{j'|D'_e}^{(i)}\right)_{i=1,\dots,n}$, obtain an estimate of the h-function $h_{j|j';D'_e}$ which we will denote $\widehat{h}_{j|j';D'_e}$.

   (iii) Define $u_{j|D_e}^{(i)} := \widehat{h}_{j|j';D'_e}\left(u_{j|D'_e}^{(i)} \middle| u_{j'|D'_e}^{(i)}\right), \qquad i = 1, \dots, n.$

   (iv) Based on $\left(u_{a_e|D_e}^{(i)}, u_{b_e|D_e}^{(i)}\right)_{i=1,\dots,n}$ obtain an estimate of the copula density $c_{a_e,b_e;D_e}$.

In plain words, sequential estimation works as follows: In the first tree, each node corresponds to one random variable. We use samples of these random variables to obtain estimates of all pair-copulas that correspond to the edges of the tree. The nodes of the second tree can be identified with random variables (c.f. equation (2.3)), but we are not directly provided with samples from them. Thus, we first estimate the h-functions involved in (2.3) and apply them to obtain pseudo-samples corresponding to each node. Based on the pseudo-samples, we can then estimate the copulas corresponding to edges in the second tree, and so on. At the end of the procedure we have estimates for all copula densities and all h-functions that are required to evaluate the density of the full R-vine copula.

Structure selection

In order to tell an estimation algorithm which pair-copulas and h-functions have to be estimated, we always have to specify an R-vine tree sequence in advance. The tree sequence corresponding to an R-vine copula model is also called the *structure* of the vine. The structure can have a notable influence on the estimator's performance. Sometimes *a priori* expert knowledge in the field of application is available and gives some intuition about a good structure. In other cases, automatic structure selection procedures are available. Czado et al. (2013) give a review over recent developments.

The most popular procedure is Dißmann et al. (2013)'s heuristic which sequentially selects the tree with the highest cumulative strength of dependence between the corresponding variables. To do that, all possible h-functions of a selected tree have to be estimated, before we can advance to the next higher level. Although a variety of dependence measures can be used, we will focus on Kendall's τ for convenience. The heuristic is summarized in the following definition.

Definition 2.13. *Assume we are given iid samples* $(u_1^{(i)}, \dots, u_d^{(i)})_{i=1,\dots,n}$ *from a random vector* $(U_1, \dots, U_d) \sim C$. *The* ***sequential structure selection method*** *works as follows:*

1. *For all possible pairs* $\{a, b\}$, $1 \leq a < b \leq d$, *calculate* $\widehat{\tau}_{a,b}$ *as the empirical Kendall's* τ *of the corresponding variables.*
2. *Define* E_1 *as the edges of the spanning tree that maximizes*
$$\sum_{e=\{a_e,b_e\}\in E_1} \widehat{\tau}_{a_e,b_e}.$$
3. *For all* $e \in E_1$, *obtain estimates* $\widehat{h}_{a_e|b_e}$ *and* $\widehat{h}_{b_e|a_e}$. *Then define pseudo-observations*
$$u_{a_e|b_e}^{(i)} := \widehat{h}_{a_e|b_e}\Big(u_{a_e}^{(i)}\Big|u_{b_e}^{(i)}\Big), \quad u_{b_e|a_e}^{(i)} := \widehat{h}_{b_e|a_e}\Big(u_{b_e}^{(i)}\Big|u_{a_e}^{(i)}\Big), \qquad i = 1, \dots, n.$$
4. *For all* $k = 2, \dots, d-1$:
   - *(i) For all conditional pairs* $\{a, b; D\}$ *that fulfill the proximity condition, calculate* $\widehat{\tau}_{a,b;D}$ *as the empirical Kendall's* τ *of the corresponding pseudo observations.*
   - *(ii) Define* E_k *as the edges of the spanning tree that maximizes*
   $$\sum_{e=\{a_e,b_e;D_e\}\in E_k} \widehat{\tau}_{a_e,b_e;D_e}.$$
   - *(iii) For all* $e \in E_k$, *obtain estimates* $\widehat{h}_{a_e|b_e;D_e}$ *and* $\widehat{h}_{b_e|a_e;D_e}$. *Then define pseudo-observations*
   $$u_{a_e|b_e,D_e}^{(i)} := \widehat{h}_{a_e|b_e;D_e}\Big(u_{a_e|D_e}^{(i)}\Big|u_{b_e|D_e}^{(i)}\Big), \quad u_{b_e|a_e,D_e}^{(i)} := \widehat{h}_{b_e|a_e;D_e}\Big(u_{b_e|D_e}^{(i)}\Big|u_{a_e|D_e}^{(i)}\Big),$$
   for $i = 1, \dots, n$.

2.2 Bivariate kernel density estimation

In the remainder of this chapter, we will introduce *kernel density estimation (KDE)* as a non-parametric method for the estimation of probability densities. Univariate KDE was introduced by Rosenblatt (1956) and Parzen (1962). Wand (1992) and Wand and Jones (1993) elaborately discuss a natural extension to the multivariate case. For a more extensive introduction, the reader is referred to Wand and Jones (1994) and Simonoff (1996). We will put our focus on the bivariate case directly, as this will be our main interest in the next chapter.

2.2.1 The estimator

Let $(X, Y) \in \mathbb{R}^2$ be a random vector with density f and assume we are given *iid* copies $(X_i, Y_i)_{i=1,\dots,n}$.[1] Recall that the density can be defined as

$$f(x, y) = \frac{\partial^2 F(x, y)}{\partial x \partial y} = \lim_{\epsilon_x \to 0} \lim_{\epsilon_y \to 0} \frac{F(x + \epsilon_x, y + \epsilon_y) - F(x - \epsilon_x, y - \epsilon_y)}{4\epsilon_x \epsilon_y}, \tag{2.4}$$

where F is the *cdf* corresponding to f. A natural estimator of the density could be obtained by fixing values for ϵ_x, ϵ_y in (2.4) and using the empirical *cdf*, $\widehat{F}_n$, as an estimator for F. For simplicity, take $\epsilon_x = \epsilon_y = b$, for some small $b > 0$. The resulting estimator is

$$\begin{aligned} \widehat{f}_n(x, y) &= \frac{\widehat{F}_n(x + b, y + b) - \widehat{F}_n(x - b, y - b)}{4b^2} \\ &= \frac{\#\big\{(X_i, Y_i) \in [x - b, x + b] \times [y - b, y + b]\big\}/n}{4b^2}. \end{aligned} \tag{2.5}$$

The estimator calculates the fraction of all (X_i, Y_i) that lie in a (rectangular) neighborhood around the point (x, y) and divides it by the neighborhood's area. The parameter b controls the size of the neighborhood and is usually called the *bandwidth* of the estimator.

Note that the estimator in (2.5) can also be rewritten as

$$\widehat{f}_n(x, y) = \frac{1}{nb^2} \sum_{i=1}^{n} K\left(\frac{x - X_i}{b}\right) K\left(\frac{x - Y_i}{b}\right), \tag{2.6}$$

where the *kernel* K is defined as

$$K(z) := \begin{cases} 1/2 & \text{if } -1 \leq z \leq 1 \\ 0 & \text{else.} \end{cases}$$

An estimator of the form (2.6) is called a bivariate *kernel density estimator*. In the above case, the kernel K corresponds to the uniform probability density on $[-1, 1]$. In general, one could use any probability density function as the kernel K and the resulting estimator will be a proper probability density function. However, it is usually assumed that the kernel is bounded, i.e. $K(z) < \infty$ for $z \in \mathbb{R}$, and symmetric. This gives the resulting estimator nice properties and facilitates its theoretical analysis.

Let us subsume our considerations with a formal definition. Take K as any symmetric, bounded probability density function. For $b > 0$, we will use the short notation $K_b(x) = K(x/b)/b$.

[1] In the literature on nonparametric estimation, often *iid* copies are considered rather than *iid* observations. This way, one wants to emphasize that the estimator is a random variable itself. We will follow this convention in the larger part of this thesis. It is, however, a mere technicality and should not cause any confusion.

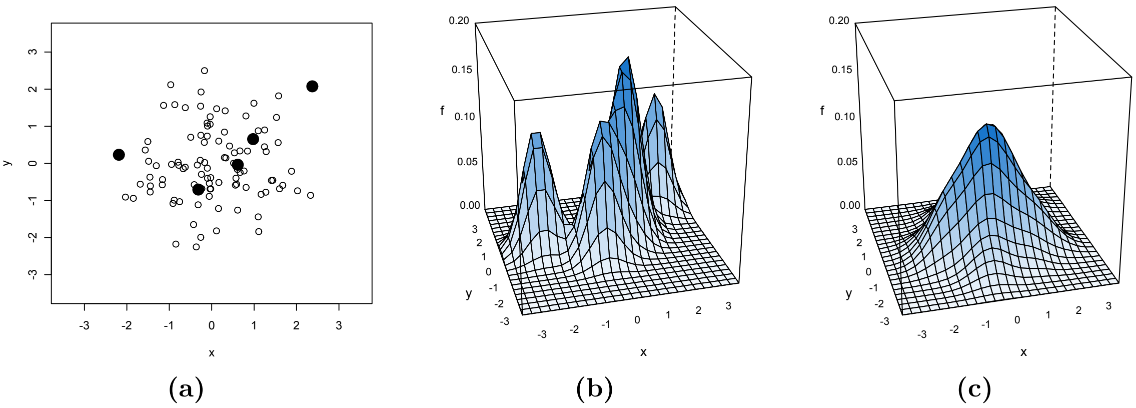


Figure 2.6: Illustration of the kernel density estimator. (a) 100 samples of two independent standard normal random variables. Five observations are marked as filled circles. (b) Kernel density estimator applied to the marked samples. (c) Kernel density estimator applied to the full sample. Bandwidths were set equally ($b_n = 2$) in both cases.

Definition 2.14. *The* ***kernel density estimator*** *with bandwidth parameter* $b_n > 0$ *is given by*

$$\widehat{f}_n(x, y) = \frac{1}{n} \sum_{i=1}^{n} K_{b_n}\Big(x - X_i\Big) K_{b_n}\Big(y - Y_i\Big), \quad \textit{for all } (x, y) \in \mathbb{R}^2.$$

There are two ways to interpret the kernel density estimator — from a local and global point of view. The local view essentially corresponds to our previous considerations. Here, we see a point estimate as a weighted average of frequencies in a neighborhood of that point. The weighting is conducted according to the kernel function K and the size of the neighborhood is controlled by the bandwidth b_n.

From a global point of view, an estimate of the density is constructed as follows: Centered upon each observation, a ‘bump’ in the shape of a scaled kernel, $K(\cdot/b_n)/b_n$, is placed and all the bumps are averaged to give the whole surface of the density. This is illustrated in Figure 2.6. In (a) we see simulated data of two independent standard normal random variables. Five observations are marked as thick, filled circles. In (b) the kernel density estimator was applied to the marked samples only. We can clearly see distinct bumps. They have the shape of a Gaussian density, since it was used as the kernel function. One of the bumps appears to be higher which is caused by an overlap between two bumps that add up to give a higher peak. In the (c) the kernel density estimator was applied to the full sample. The single bumps are not distinguishable any longer and the estimate looks approximately like the bivariate density of independent standard normal random variables.

Note that the bandwidth b_n in the above definition is annotated to depend on n. This is done in order to facilitate asymptotic analysis of the estimator to which we turn next.

2.2.2 Properties

An important tool to analyze the behavior of a kernel estimator is to consider asymptotic approximations of bias and variance. In the following, we state and prove those for the kernel density estimator. We will not need this particular result in the remainder of the thesis, but the proof may be instructive. It makes use of the basic techniques that are also a key tool for the derivation of asymptotic approximations for more complicated estimators.

Proposition 2.4. *Let f be supported on $\mathbb{R}^2$, twice continuously differentiable and let $b_n \to 0$ and $nb_n^2 \to \infty$ as $n \to \infty$. Denote further, $f_x := \partial f / \partial x$, $f_{xx} := \partial^2 f / \partial x^2$, and so forth. Then for all $(x, y) \in \mathbb{R}^2$,*

$$\begin{aligned}\mathrm{Bias}\Big[\widehat{f}_n(x,y)\Big] &= \frac{\sigma_K^2 b_n^2}{2}\Big[f_{xx}(x,y) + f_{yy}(x,y)\Big] + o\Big(b_n^2\Big)\\ \mathrm{Var}\Big[\widehat{f}_n(x,y)\Big] &= \frac{d_K^2}{nb_n^2} f(x,y) + o\bigg(\frac{1}{nb_n^2}\bigg),\end{aligned}$$

where

$$\sigma_K^2 = \int_{\mathbb{R}} s^2 K(s) ds \qquad \text{and } d_K = \int_{\mathbb{R}} K^2(s) ds.$$

Proof. Let us start with derivation the of the bias. First, we calculate the expectation of the estimator:

$$\begin{aligned}\mathrm{E}\Big[\widehat{f}_n(x,y)\Big] &= \mathrm{E}\bigg[\frac{1}{n}\sum_{i=1}^n K_{b_n}\Big(x - X_i\Big) K_{b_n}\Big(y - Y_i\Big)\bigg]\\ &= \mathrm{E}\bigg[K_{b_n}\Big(x - X\Big) K_{b_n}\Big(y - Y\Big)\bigg]\\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} K_{b_n}\Big(x - s\Big) K_{b_n}\Big(y - t\Big) f(s,t) ds dt\\ &= \frac{1}{b_n^2}\int_{\mathbb{R}}\int_{\mathbb{R}} K\bigg(\frac{x-s}{b_n}\bigg) K\bigg(\frac{y-t}{b_n}\bigg) f(s,t) ds dt, \qquad (2.7)\end{aligned}$$

where the second equality holds because $(X_i, Y_i)_{i=1,\ldots,n}$ are *iid* copies of the random vector (X, Y). Next, we will use the change of variables $s = x - b_n w, t = y - b_n z$ and a second-order Taylor approximation for f in the point (x, y):

$$\begin{aligned}(2.7) &= \int_{\mathbb{R}}\int_{\mathbb{R}} K\Big(w\Big) K\Big(z\Big) f(x - b_n w, y - b_n z) dw dz\\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} K\Big(w\Big) K\Big(z\Big)\bigg[f(x,y) - f_x(x,y) b_n w - f_y(x,y) b_n z\\ &\qquad + \frac{1}{2} f_{xx}(x,y) b_n^2 w^2 + \frac{1}{2} f_{yy}(x,y) b_n^2 z^2 + o\Big(b_n^2\Big)\bigg] dw dz. \qquad (2.8)\end{aligned}$$

Now recall that K is a symmetric probability density on $\mathbb{R}$. In particular,

$$\int_{\mathbb{R}} K(w) dw = 1, \qquad \int_{\mathbb{R}} w K(w) dw = 0,$$

and the second and third terms in brackets in (2.8) vanish. This gives

$$\begin{aligned}(2.8) &= f(x,y) + \frac{b_n^2}{2}\Big[f_{xx}(x,y)\int_{\mathbb{R}} w^2K(w)dw + f_{yy}(x,y)\int_{\mathbb{R}} z^2K(z)dz\Big] + o\Big(b_n^2\Big)\\ &= f(x,y) + \frac{\sigma_K^2 b_n^2}{2}\Big[f_{xx}(x,y) + f_{yy}(x,y)\Big] + o\Big(b_n^2\Big), \qquad (2.9)\end{aligned}$$

and finally

$$\begin{aligned}\mathrm{Bias}\Big[\widehat{f}_n(x,y)\Big] &= \mathrm{E}\Big[\widehat{f}_n(x,y)\Big] - f(x,y)\\ &= \frac{\sigma_K^2 b_n^2}{2}\Big[f_{xx}(x,y) + f_{yy}(x,y)\Big] + o\Big(b_n^2\Big)\end{aligned}$$

as claimed.

For the variance we have that

$$\mathrm{Var}\Big[\widehat{f}_n(x,y)\Big] = \mathrm{E}\Big[\widehat{f}_n^2(x,y)\Big] - \mathrm{E}\Big[\widehat{f}_n(x,y)\Big]^2,$$

Using independence of the copies, the first part can be calculated as

$$\begin{aligned}\mathrm{E}\Big[\widehat{f}_n^2(x,y)\Big] &= \mathrm{E}\Bigg[\frac{1}{n^2}\sum_{i=1}^n\sum_{j=1}^n K_{b_n}(x-X_i)K_{b_n}(y-Y_i)K_{b_n}(x-X_j)K_{b_n}(y-Y_j)\Bigg]\\ &= \frac{1}{n^2}\Bigg[\sum_{i=j}\mathrm{E}\Big[K_{b_n}^2(x-X)K_{b_n}^2(y-Y)\Big] + \sum_{i\neq j}\mathrm{E}\Big[K_{b_n}(x-X)K_{b_n}(y-Y)\Big]^2\Bigg]\\ &= \frac{1}{n^2}\Bigg[n\mathrm{E}\Big[K_{b_n}^2(x-X)K_{b_n}^2(y-Y)\Big] + n(n-1)\mathrm{E}\Big[K_{b_n}(x-X)K_{b_n}(y-Y)\Big]^2\Bigg].\end{aligned}$$

We have already seen that $\mathrm{E}\Big[\widehat{f}_n(x,y)\Big] = \mathrm{E}\Big[K_{b_n}(x-X)K_{b_n}(y-Y)\Big]$. Together, this gives

$$\begin{aligned}\mathrm{Var}\Big[\widehat{f}_n(x,y)\Big] &= \mathrm{E}\Big[\widehat{f}_n^2(x,y)\Big] - \mathrm{E}\Big[\widehat{f}_n(x,y)\Big]^2\\ &= \underbrace{\frac{1}{n}\mathrm{E}\Big[K_{b_n}(x-X)^2K_{b_n}(y-Y)^2\Big]}_{=:V_1} - \underbrace{\frac{1}{n}\mathrm{E}\Big[K_{b_n}(x-X)K_{b_n}(y-Y)\Big]^2}_{=:V_2}.\end{aligned} \tag{2.10}$$

Now put again $s = x - b_nw, t = y - b_nz$ and expand the first part of (2.10) by a first-order Taylor approximation.

$$\begin{aligned}
V_1 &= \frac{1}{nb_n^2}\int_{\mathbb{R}}\int_{\mathbb{R}} K^2\Big(w\Big)K^2\Big(z\Big)f(x-b_nw,y-b_nz)dwdz \\
&= \frac{1}{nb_n^2}\int_{\mathbb{R}}\int_{\mathbb{R}} K^2\Big(w\Big)K^2\Big(z\Big)\Big[f(x,y)-f_x(x,y)b_nw-f_y(x,y)b_nz+o(b_n)\Big]dwdz \\
&= \frac{1}{nb_n^2}\int_{\mathbb{R}}\int_{\mathbb{R}} K^2\Big(w\Big)K^2\Big(z\Big)f(x,y)dwdz \\
&\quad \underbrace{-\frac{1}{nb_n}\int_{\mathbb{R}}\int_{\mathbb{R}} K^2\Big(w\Big)K^2\Big(z\Big)\Big[f_x(x,y)w+f_y(x,y)z+o\Big(\frac{1}{nb_n}\Big)\Big]dwdz}_{=o\left(\frac{1}{nb_n^2}\right)} \\
&= \frac{1}{nb_n^2}\int_{\mathbb{R}}\int_{\mathbb{R}} K^2\Big(w\Big)K^2\Big(z\Big)f(x,y)dwdz+o\Big(\frac{1}{nb_n^2}\Big).
\end{aligned}$$

The second part of (2.10) can be easily approximated by recalling that we already calculated the expectation to be of order $O(b_n^2)$, see (2.9). Thus,

$$V_2 = \frac{1}{n}\mathrm{E}\Big[K_{b_n}(x-X)K_{b_n}(y-Y)\Big]^2 \overset{(2.9)}{=} \frac{1}{n}\Big(O(b_n^2)\Big)^2 = O\Big(\frac{b_n^4}{n}\Big) = o\Big(\frac{1}{nb_n^2}\Big),$$

where the last equality holds due to

$$\frac{b_n^4/n}{1/(nb_n^2)} = b_n^6 \to 0.$$

Taking both things together we finally get

$$\begin{aligned}
\mathrm{Var}\Big[\widehat{f}_n(x,y)\Big] &= V_1+V_2 \\
&= \frac{1}{nb_n^2}\int_{\mathbb{R}}\int_{\mathbb{R}} K^2\Big(w\Big)K^2\Big(z\Big)f(x,y)dwdz+o\Big(\frac{1}{nb_n^2}\Big) \\
&= \frac{d_K^2}{nb_n^2}f(x,y)+o\Big(\frac{1}{nb_n^2}\Big).
\end{aligned}$$

□

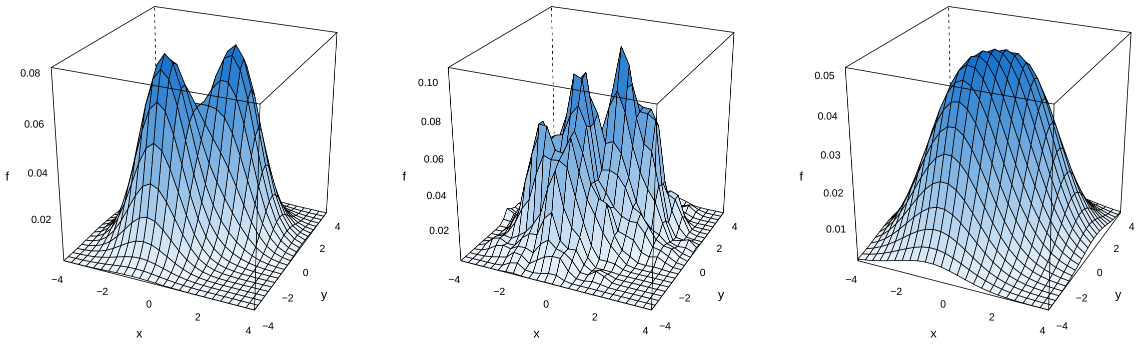
The *mean integrated squared error (MISE)* is a measure of the estimator's accuracy and is given by the sum of squared bias and variance, integrated over all $(x,y)\in\mathbb{R}^2$. An asymptotic approximation can easily be given with the above results.

Corollary 2.5. *Under the assumptions of Proposition 2.4, an asymptotic approximation of the mean integrated squared error can be given as*

$$\mathrm{MISE}\Big[\widehat{f}_n(x,y)\Big] = \frac{\sigma_K^4 b_n^4}{4}\int_{\mathbb{R}}\int_{\mathbb{R}}\Big[f_{xx}(x,y)+f_{yy}(x,y)\Big]^2dxdy+\frac{d_K^2}{nb_n^2}+o\Big(\frac{1}{nb_n^2}\Big)+o\Big(b_n^2\Big),$$

where

$$\sigma_K^2=\int_{\mathbb{R}} s^2K(s)ds \qquad \textit{and } d_K=\int_{\mathbb{R}} K^2(s)ds.$$



(a) true **(b)** $b_n = 1$ **(c)** $b_n = 4$

Figure 2.7: The effect of the bandwidth parameter in kernel density estimation. (a) True density of a bimodal distribution. (b) Undersmoothed kernel density estimate on simulated data ($n = 100$). (c) Oversmoothed kernel density estimate on simulated data ($n = 100$).

All derived expressions depend on the bandwidth b_n as well as the kernel K. Thus, their choice affects the properties of the estimator. Their role will be discussed in more detail in the next sections.

2.2.3 The bandwidth

In this section we want to shed some light on the role of the bandwidth in kernel density estimation. To do so, let us assume that we have fixed a kernel function K. Recall the asymptotic approximations of bias and variance were given as

$$\begin{aligned}\operatorname{Bias}\left[\widehat{f}_n(x,y)\right] &= \frac{\sigma_K^2 b_n^2}{2}\left[f_{xx}(x,y) + f_{yy}(x,y)\right] + o\left(b_n^2\right)\\ \operatorname{Var}\left[\widehat{f}_n(x,y)\right] &= \frac{d_K^2}{nb_n^2} f(x,y) + o\left(\frac{1}{nb_n^2}\right),\end{aligned}$$

with some constants σ_K and d_K. In general, it is desirable to have a small bias as well as a small variance. However, the bias is decreasing in b_n whereas the variance is increasing in b_n.

This phenomenon is usually called the *bias-variance trade-off.* and is illustrated in Figure 2.7. Figure 2.7a shows the density of a Gaussian mixture distribution with equal mixing probabilities, identity covariance matrices and means at $(-1,-1)$ and $(1,1)$. It has two distinct modes at the means of the two parts of the mixture. Figure 2.7b shows a kernel density estimate on simulated data ($n = 100$) from this distribution, where we used $b_n = 1$ as the bandwidth. We can recognize the two true modes of the distribution, but it actually seems as if there were more than just the two. Overall, the estimated density is very wiggly and appears to have overfit the data, i.e. that it shows features of the sample that are due to random variation. Such an estimate is called *undersmoothed* and is a consequence of a too small bandwidth. In this case, the estimator has high variance and small bias. Figure 2.7c shows

Kernel name	$K(x)$	σ_K^2	d_K
Uniform	$\frac{1}{2}\mathbb{1}_{[-1,1]}$	$\frac{1}{2}$	$\frac{1}{3}$
Gaussian	$\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$	1	$\frac{1}{2\pi}$
Epanechnikov	$\frac{3}{4}(1-x^2)\mathbb{1}_{[-1,1]}$	$\frac{1}{5}$	$\frac{3}{5}$
Biweight	$\frac{15}{16}(1-x^2)^2\mathbb{1}_{[-1,1]}$	$\frac{1}{7}$	$\frac{5}{7}$
Triweight	$\frac{35}{32}(1-x^2)^3\mathbb{1}_{[-1,1]}$	$\frac{1}{9}$	$\frac{350}{429}$
Cosine	$\frac{\pi}{4}\cos(\pi x/2)\mathbb{1}_{[-1,1]}$	$1-\frac{8}{\pi^2}$	$\frac{\pi^2}{16}$

Table 2.3: Commonly used kernel functions and the resulting constants σ_K^2 and d_K.

another kernel density estimate, where this time we used $b_n = 4$. The estimate is extremely smooth but important features of the density were ‘smoothed away’. In particular, the two modes are not distinguishable any longer. Such an estimate is called *oversmoothed* and is a consequence of a too large bandwidth. In this case, the estimator has small variance and large bias.

A good choice of the bandwidth should balance the two opposing forces. In theory, it is often possible to derive an asymptotically optimal bandwidth by minimizing the leading terms of the asymptotic approximation of the MISE, commonly referred to as *asymptotic mean integrated squared error (AMISE)*. However, this involves a priori knowledge (or approximation) of the true density. Popular data-driven selection strategies are based on cross-validation, but are computationally intensive. The most popular instances are least-squares cross-validation (Rudemo, 1982) and biased cross validation (Scott and Terrell, 1987).

2.2.4 The kernel

In Proposition 2.4 we found that bias and variance also depend on the kernel function through the constants σ_K^2 and d_K respectively. Epanechnikov (1969) derived an optimal kernel in the AMISE sense, which was subsequently named for him. This and other commonly used kernel functions as well as their respective values for σ_K^2 and d_K are listed in Table 2.3. All of these kernels are symmetric and bounded probability density functions.

In univariate kernel density estimation one can explicitly calculate the asymptotic efficiency of the estimator for a given kernel (assuming optimal choice of the bandwidth). It turns out that the relative loss in efficiency compared with the optimal Epanechnikov kernel is less than 2% for Biweight, Triweight and Cosine kernels, and less than 8% for Uniform and Gaussian kernels (c.f. Silverman, 1986). Hence, the choice of kernel has a rather small impact on the overall accuracy of kernel density estimators, especially compared with the choice of bandwidth. As a consequence, kernel selection is usually ignored, as each of the presented kernel functions constitutes a viable choice. In this thesis, we used Epanechnikov or Gaussian kernels which are the two most popular choices.

Chapter 3

Kernel estimation of bivariate copula densities

This chapter deals with the estimation of an unknown bivariate copula density using kernel techniques. We will see that the boundedness of the support of a copula density creates the need for more advanced techniques than the one considered in Section 2.2. In particular, we will consider three different ways to cope with this problem: With data augmentation in the *mirror-reflection technique*, with the use of *beta kernels* to directly match the bounded support, and by a *transformation technique*.

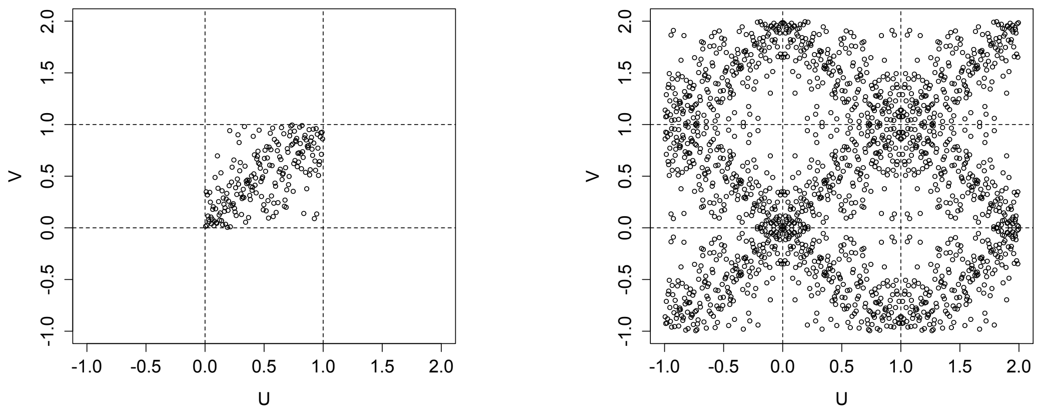
In the remainder of this chapter, we consider the following general setup. Let $U, V \sim U[0,1]$ be random variables with joint distribution C and corresponding density $c : [0,1]^2 \to \mathbb{R}$. We assume to have *iid* copies $(U_i, V_i)_{i=1,\ldots,n}$ from the copula C and our interest is in estimating the density c.

3.1 A naive estimator

Recall the bivariate kernel density estimator (see Definition 2.14) which when applied to a copula sample, we will denote by $\widehat{c}_n$:

$$\widehat{c}_n(u,v) = \frac{1}{n}\sum_{i=1}^{n} K_{b_n}\Big(u - U_i\Big) K_{b_n}\Big(v - V_i\Big), \quad \text{for all } (u,v) \in [0,1]^2,$$

where we again used the notation $K_b(\cdot) = 1/b\, K(\cdot/b)$, K is a symmetric, bounded probability density function on $\mathbb{R}^2$ and $b_n > 0$. Now consider a data point close to a boundary of the unit square. For such a point, the estimator will put a considerable amount of probability mass outside the unit square. This in turn implies that $\widehat{c}_n$ is not a density function on $[0,1]^2$, since it no longer integrates to one over the unit square. In addition, the estimator will have a severe bias on the boundaries. Due to these unsatisfying properties there is a need for other, more advanced estimation techniques that overcome these problems.



(a) Original data **(b)** Augmented data

Figure 3.1: The data augmentation process. The set in (b) is obtained by reflecting all original data points w.r.t to all corners and edges.

3.2 Mirror-reflection estimator

There is an intuitive way of adapting $\widehat{c}_n$ to make sure that it is a density on $[0,1]^2$. Just gather all the probability mass that was put outside of the unit square, and redistribute it back to $[0,1]^2$. Following this idea, the so called *mirror-reflection technique* was introduced to the estimation of copula densities by Gijbels and Mielniczuk (1990). It requires the reflection of all data points w.r.t. to all corners and edges of the unit square. When $\widehat{c}_n$ is applied to this augmented data set, all probability mass that was initially put outside of the square is reflected back in.

3.2.1 The basic estimator

Formally, the augmented data is defined by

$$\left(\tilde{U}_{ik}, \tilde{V}_{ik}\right)_{k=1,\ldots,9} = \Big\{(U_i, V_i), (-U_i, V_i), (U_i, -V_i), (-U_i, -V_i), (U_i, 2-V_i),$$
$$(-U_i, 2-V_i), (2-U_i, V_i), (2-U_i, -V_i), (2-U_i, 2-V_i)\Big\},$$

for all $i = 1, \ldots, n$. This set contains the original observation itself, and all new data points that were generated by reflecting the original observation w.r.t to all four corners and all four edges. A visualization of the augmented data set is given in Figure 3.1. A kernel will now be placed on top of each of these points and all the probability mass inside the unit square is collected. This procedure is illustrated in Figure 3.2. In the left picture, we see a data point with a contour line of the kernel that is placed on top of it. Some of the probability mass in the interior of the contour line is not inside the unit square. On the right, the point is reflected w.r.t. to all corners and edges and a kernel is placed on top of each point (some of the new points

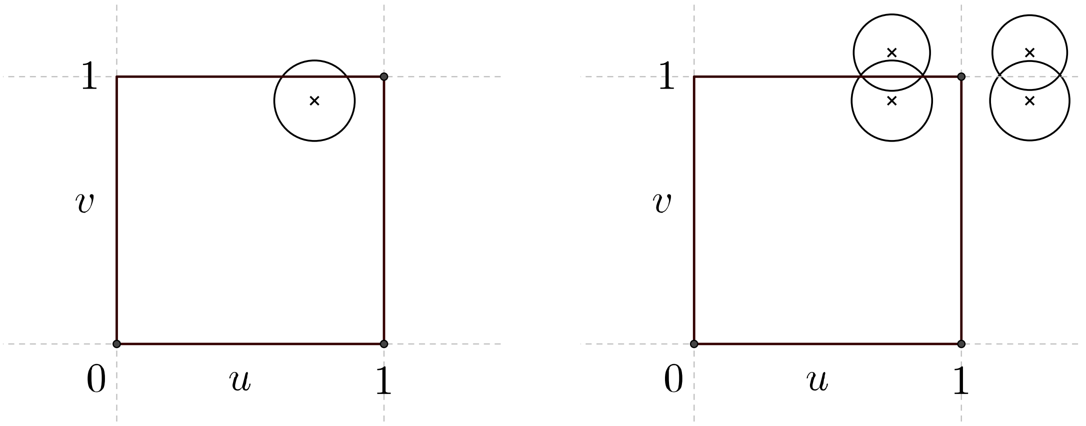


Figure 3.2: Left: A data point and a contour of its kernel. Some of the probability mass is outside the unit square. Right: The data point is reflected w.r.t to all edges and corners. The missing probability mass is reflected back inside.

are not in this picture, because they are too far away). Thereby, also the probability mass that was missing on the left is reflected back into the unit square.

The described estimator can be written formally as follows.

Definition 3.1. *The* ***mirror-reflection estimator*** *of a copula density $c(u, v)$ with bandwidth parameter $b_n > 0$ is given by*

$$\widehat{c}_n^{(MR)}(u, v) = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{9} K_{b_n}\Big(u - \tilde{U}_{ik}\Big) K_{b_n}\Big(v - \tilde{V}_{ik}\Big), \qquad \textit{for all } (u, v) \in [0, 1]^2.$$

Gijbels and Mielniczuk (1990) proved strong consistency and asymptotic normality of this estimator. We can see that due to the boundedness of K, the estimate $\widehat{c}_n^{(MR)}(u, v)$ will always be bounded. This is not a very pleasant feature, since most of the popular copula families are unbounded near some of the corners. Another interesting issue is revealed when looking at the asymptotic bias of this estimator. Later on, we will see that

$$\text{ABias}[\widehat{c}_n^{(MR)}(u, v)] = \frac{\sigma_K^2}{2} b_n^2 \Big[c_{uu}(u, v) - c_{vv}(u, v)\Big],$$

where $c_{ww} = \partial^2 C / \partial^2 w$ denotes the second order partial derivative of the copula density and $\sigma_K^2 = \int_{-1}^{1} t^2 K(t) dt$ is a constant depending on the kernel. As soon we have chosen a kernel K and a bandwidth b_n the magnitude of this quantity will only depend on the second order partial derivatives of the true copula density c. Again, for many popular parametric copula families, the second order partial derivatives c_{uu} and c_{vv} are unbounded near some of the corners of the unit square, leading to an unbounded bias in these regions.

Omelka et al. (2009) faced a similar issue when looking at a mirror-reflection estimate of the copula C. The bias term in this case is similar to the one above, but depending on the second order partial derivatives of C instead.

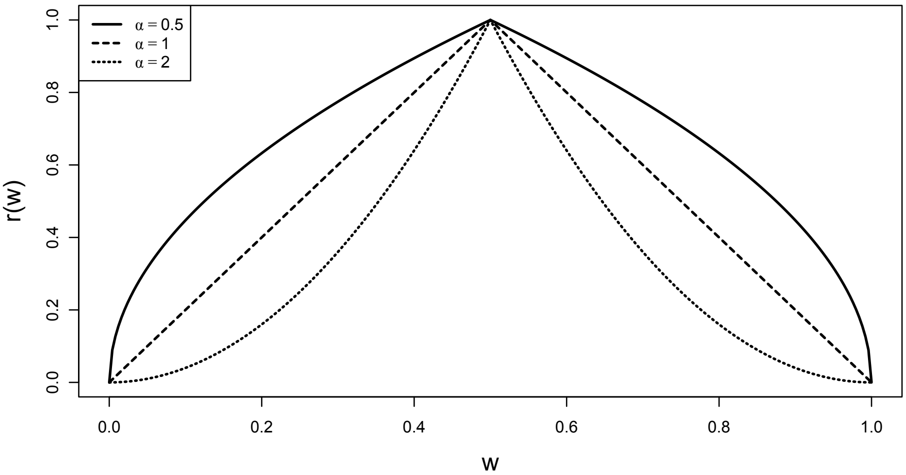


Figure 3.3: The shrinkage factor $r(w)$ for different values of α. The functions were normalized so that their maximum is one.

3.2.2 An improved estimator

Motivated by this issue, Omelka et al. (2009) suggest an improved version of the presented mirror-reflection estimator. The idea is to 'shrink' the bandwidth b_n when coming closer to the corners. If the shrinking is fast enough, the procedure will result in a bounded asymptotic bias of $\widehat{C}_n^{(MR)}(u,v)$. We will now use the same technique for a density estimate.

Definition 3.2. *The* ***improved mirror-reflection estimator*** *of a copula density $c(u,v)$ with bandwidth parameter $b_n > 0$ and shrinkage function $r : [0,1] \to \mathbb{R}$ is given by*

$$\widehat{c}_n^{(MRS)}(u,v) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{9} K_{r(u)b_n}\Big(u - \tilde{U}_{ik}\Big) K_{r(v)b_n}\Big(v - \tilde{V}_{ik}\Big), \quad \textit{for all } (u,v) \in [0,1]^2.$$

We can see that the usual bandwidth parameter b_n is multiplied by factors $r(u)$ and $r(v)$ respectively. These factors contrive that the effective bandwidths $r(u)b_n, r(v)b_n$ adapt to the point (u,v) at which the estimation takes place. Omelka et al. (2009) showed that for their copula estimator, using

$$r(w) = \min(w^\alpha, (1-w)^\alpha), \quad \alpha \geq 1/2,$$

is sufficient for Gaussian, Gumbel, Clayton and Student copulas. The behavior of $r(w)$ is illustrated in Figure 3.3. Note that the actual maximum of the functions is decreasing in α. In practice, one would compensate this by using a bigger bandwidth parameter b_n. We therefore normalized the functions $r(w)$ to make them comparable. We see that the higher the value of α the faster the function $r(w)$ tends to zero towards the boundaries. This also means that the effective bandwidth $r(w)b_n$ will approach zero more quickly.

3.2.3 Properties

In this section, we will discuss the asymptotic properties of the estimator. For this and the following estimators, we will focus on the interior of the unit square, since the edges have zero measure, but complicate analysis. It should be noted, however, that asymptotic behavior may be different in these cases.

In the following proposition, we give asymptotic expressions for bias and variance. For simplicity, assume that K is supported on $[-1, 1]$, sufficiently smooth and a symmetric probability density function.

Proposition 3.1. *Let $c(u, v)$ be twice continuously differentiable on $(0, 1)^2$, $b_n \to 0$ and $nb_n^2 \to \infty$ as $n \to \infty$. Then for all $(u, v) \in (0, 1)^2$,*

$$\operatorname{Bias}[\widehat{c}_n^{(MRS)}(u, v)] = \frac{\sigma_K^2}{2} b_n^2 \Big[r^2(u)c_{uu}(u, v) - r^2(v)c_{vv}(u, v)\Big] + o\Big(b_n^2\Big),$$

$$\operatorname{Var}[\widehat{c}_n^{(MRS)}(u, v)] = \frac{d_K^2}{r(u)r(v)nb_n^2} c(u, v) + o\left(\frac{1}{nb_n^2}\right),$$

where

$$\sigma_K^2 = \int_0^1 s^2 K(s) ds \qquad \text{and } d_K = \int_0^1 K^2(s) ds.$$

Proof. Note that for a fixed point (u, v) the bandwidth is fixed, too. The fact that, we use different bandwidths in each direction does not complicate the analysis. We can therefore rely on results obtained for the basic mirror-reflection estimator obtained by (Gijbels and Mielniczuk, 1990, Theorem 3.2). The variance is then equal to the expression given above. They also showed that the bias is asymptotically equivalent to

$$\int_{-1}^1 \int_{-1}^1 K_{h_n r(u)}(u - x) K_{h_n r(v)}(u - y) c(x, y) dx dy - c(u, v),$$

which is the same expression that would appear, when no reflection is conducted. This is due to the fact that for large n the bandwidth gets so small that the probability mass put outside the unit square goes to zero. The expression can be further approximated via a Taylor expansion. Denote $c_u = \partial c / \partial u$,$c_{uu} = \partial^2 c / \partial u^2$ and so on. Assuming $w \in [b_n r(w), 1 - b_n r(w)]$, for $w = u, v$, and using the change of variables $x = u - r(u)b_n s$, $y = v - r(v)b_n t$ gives

$$\begin{aligned}
&\int_{-1}^{1}\int_{-1}^{1} K_{b_n r(u)}(u-x)K_{b_n r(v)}(u-y)c(x,y)dxdy - c(u,v)\\
=&\int_{-1}^{1}\int_{-1}^{1} K(s)K(t)\Big[c\Big(u-r(u)b_n s, v-r(v)b_n t\Big) - c(u,v)\Big]dsdt\\
=&\int_{-1}^{1}\int_{-1}^{1} K(s)K(t)\Bigg[-c_u(u,v)r(u)b_n s - c_v(u,v)r(v)b_n t\\
&\qquad\qquad + \frac{1}{2}c_{uu}(u,v)r^2(u)b_n^2 s^2 + \frac{1}{2}c_{vv}(u,v)r^2(v)b_n^2 t^2\\
&\qquad\qquad + c_{uv}(u,v)r(u)r(v)b_n^2 st + o(b_n^2)\Bigg]dsdt.
\end{aligned}$$

When we expand the brackets we obtain six terms of the form $\int_0^1\int_0^1 K(s)K(t)\dots dsdt$. Recall that we assumed that K is a symmetric probability density. Thus, we have that

$$\int_{-1}^{1}\int_{-1}^{1} K(s)K(t)s\,dsdt = \int_{-1}^{1}\int_{-1}^{1} K(s)K(t)t\,dsdt = \int_{-1}^{1}\int_{-1}^{1} K(s)K(t)st\,dsdt = 0,$$

and the corresponding terms in the approximation of the bias vanish. Note also that

$$\int_{-1}^{1}\int_{-1}^{1} K(s)K(t)s^2\,dsdt = \int_{-1}^{1}\int_{-1}^{1} K(s)K(t)t^2\,dsdt = \sigma_K^2.$$

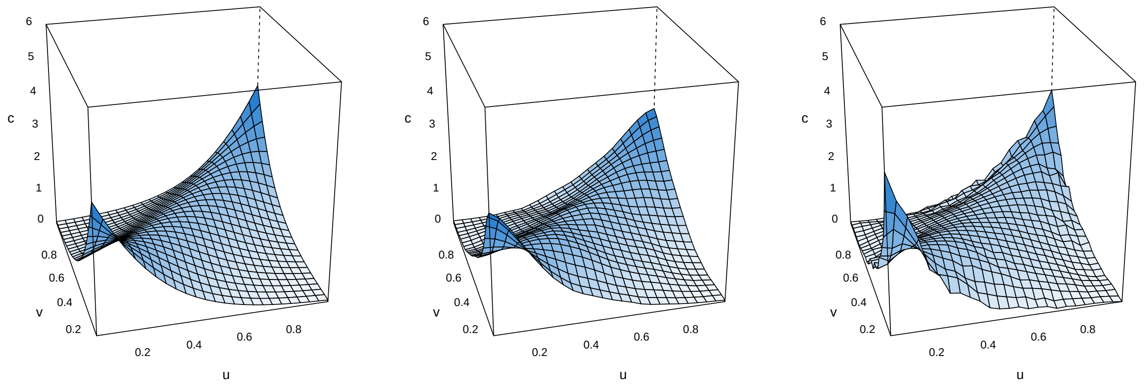
So we are left with

$$\frac{\sigma_K^2}{2}b_n^2\Big[r^2(u)c_{uu}(u,v) - r^2(v)c_{vv}(u,v)\Big] + o(b_n^2).$$

□

Clearly, the bias and variance terms explicitly depend on the true copula density c. In particular, we see that the bias is only bounded when b decreases fast enough towards the boundaries in order to offset a possible increase in c_{uu} and c_{vv}. From the findings in Figure 3.3 we can conclude that increasing the value of α will decrease the bias near the boundaries. When we do not shrink the bandwidth, i.e. $b \equiv 1$, there will be no offsetting effect and the bias will only be small if the second order partial derivatives are. The asymptotic variance on the other hand will certainly explode towards the boundaries if c does. This is true for both the simple and improved mirror-reflection estimators. It is also revealed that reducing the bias by shrinking the bandwidth towards the boundaries comes at a cost. Since b tends to zero at the boundaries, it will even further inflate the variance. This inconvenience will be even more pronounced when α is big.

An exemplary finite sample comparison of the mirror-reflection estimators was conducted on simulated data from a Frank copula with parameter $\theta = 5$. Figure 3.4 provides perspective plots of the resulting density estimates $\widehat{c}_n^{(\cdot)}$and the true density c. The basic mirror-reflection estimator notably underestimates the density close to



(a) True density c

(b) Basic mirror-reflection estimator (MR)

(c) Improved mirror-reflection estimator (MRS) ($\alpha = 1/2$)

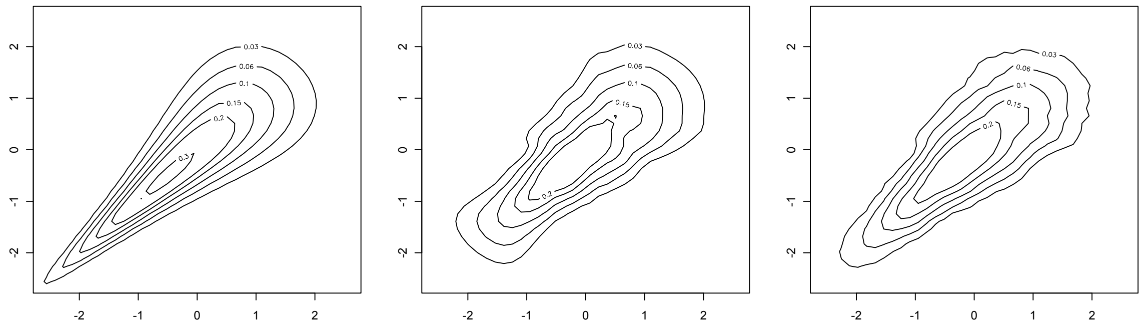
Figure 3.4: Perspective plots of the true density and estimates on simulated data ($n = 1\,000$) of a Frank copula with parameter $\theta = 5$ (Kendall's $\tau \approx 0.46$). Bandwidths were selected based on AMISE-optimality.

the corners due to the discussed bias issues. Finally, the improved mirror-reflection estimator quite accurately estimates the whole density. It can also be observed that the shrinking of the bandwidth makes the estimate more wiggly near the borders.

Note that the Frank copula model provides a bounded density. The simple mirror-reflection estimator will fail to imitate the tail behavior of unbounded densities such as for the Gaussian copula or any copula with tail dependence. We can see this by estimates on simulated data from the Clayton copula. Since unbounded functions are hard to visualize in a 3d-plot, we will make use of another exploratory tool. Marginal normal contour plots of the true density and the two estimates are given in Figure 3.5. It is known that the Clayton density grows very fast towards the corner $(0, 0)$, which gives the spiky shape in the bottom left. The basic mirror reflection estimator is not able to adapt to this feature. The contour lines stay quite broad and have no spike at all. If anything, the contours end very flatly, which is a direct effect of the boundedness of the estimate. The improved version does a little better. The contours get narrower towards the corner, although not as strong as the true density.

3.2.4 Bandwidth selection

A direct approach to find a good bandwidth for a given kernel and sample size is to consider one of the error measures *mean squared error* (MSE) or *mean integrated squared error* (MISE) for a local or global bandwidth choice respectively. In most cases, asymptotic approximations of these quantities are available to make things easier. However, they depend on the unknown density, so we can not minimize them directly. A popular resolution is to create a *rule-of-thumb* by making reference to a particular parametric family. In what follows, we will derive such a rule of thumb for the mirror-reflection density estimators.



(a) True contours

(b) Basic mirror-reflection estimator (MR)

(c) Improved mirror-reflection estimator (MRS) ($\alpha = 1/2$)

Figure 3.5: Normal contour plots of the true density and mirror-reflection estimates on simulated data ($n = 1\,000$) of a Clayton copula with parameter $\theta = 3$ (Kendall's $\tau = 0.6$). Bandwidths were selected based on AMISE-optimality w.r.t to the Frank copula.

Since we are interested in density estimates on the whole unit square, we want to focus on a global bandwidth choice. Hence, we will follow the strategy to minimize an asymptotic approximation of the MISE. If the leading terms in the asymptotic expression of squared bias and variance are integrable, the AMISE is just the integral over the sum of them. We have

$$\begin{aligned}
\text{AMISE}[\widehat{c}_n^{(MRS)}] &= \frac{\sigma_K^4}{4} b_n^4 \underbrace{\int_0^1 \int_0^1 \Big[b^2(u) c_{uu}(u,v) - b^2(v) c_{vv}(u,v)\Big]^2 dudv}_{=:\beta(b,c)} \\
&\quad + \; d_K^2 n^{-1} b_n^{-2} \underbrace{\int_0^1 \int_0^1 \frac{c(u,v)}{b(u)b(v)} dudv}_{=:\gamma(b,c)} \\
&= \frac{\sigma_K^4}{4} \beta(b,c) b_n^4 + d_K^2 \gamma(b,c) n^{-1} b_n^{-2},
\end{aligned}$$

provided that both $\beta(b,c)$ and $\gamma(b,c)$ are finite. From this, we can easily deduce that the AMISE-optimal bandwidth is given by

$$b_n^{opt} = \left(\frac{2d_K^2}{\sigma_K^4} \frac{\gamma(b,c)}{\beta(b,c)} \right)^{1/6} n^{-1/6}.$$

This expression still depends on the unknown density c. In practice we will choose a parametric reference copula family instead, and properly adjust the parameter to the strength of dependence apparent in the data, e.g. by inversion of Kendall's τ. For this reference model, the optimal bandwidth can then be computed numerically. We emphasize that this is only possible when the values for $\beta(b,c)$ and $\gamma(b,c)$ are finite. Note that $r(w) = w^\alpha$ for $w \leq 1/2$ and $r(w) = (1-w)^\alpha$ otherwise. We have that

$$\int_0^{1/2} \frac{1}{w^\alpha} dw = \int_{1/2}^{1} \frac{1}{(1-w)^\alpha} dw = \left. \frac{w^{1-\alpha}}{1-\alpha} \right|_0^{1/2},$$

which is finite if and only if $\alpha < 1$. Therefore a necessary condition for the finiteness of γ is that $\alpha < 1$. Provided this holds true, we can easily ensure a finite value for γ by choosing a bounded reference density (e.g. the Frank copula density). In practice $\alpha = 1/2$ proved to be a good choice.

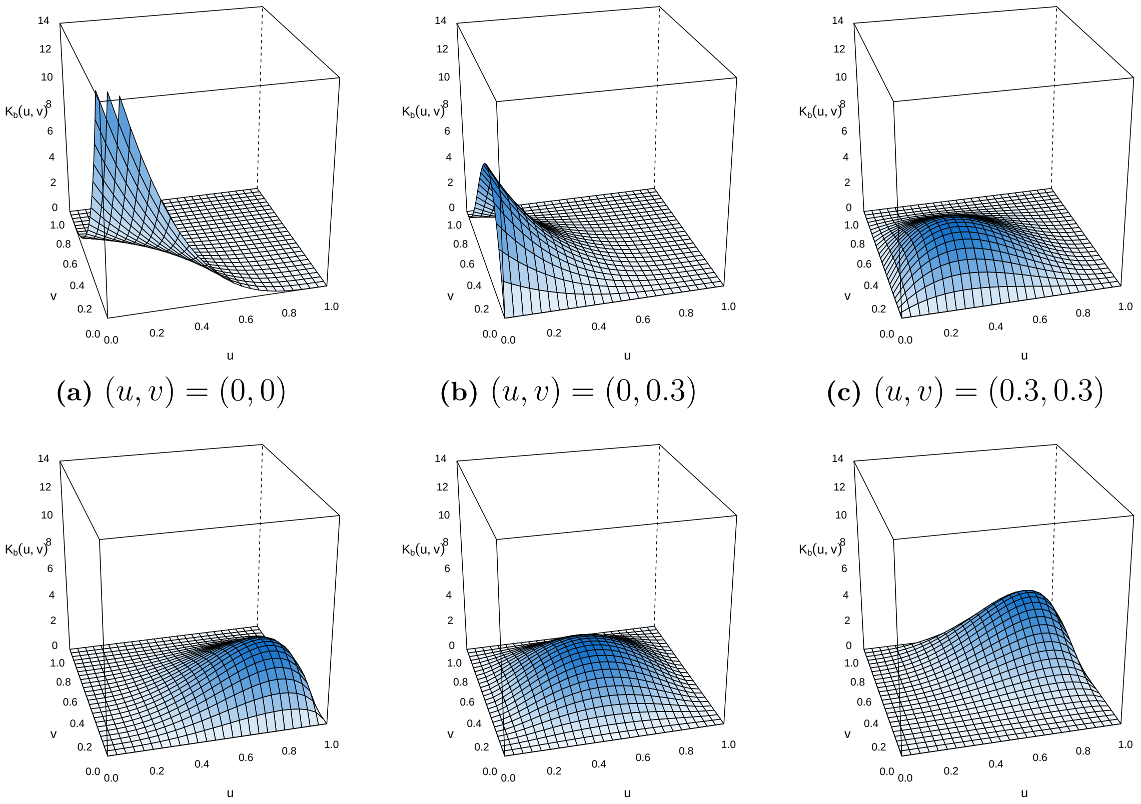
By looking at the perspective and contour plots in Figures 3.4 and 3.5, we can conclude that the bandwidth selection rules are appropriately functioning on our finite samples. If we would use smaller bandwidths, the estimates would get even more wiggly, which is quite unlikely to be the case for the true density. In addition, it makes a visualization very unpleasant. Since in the upper right part the contour plots are a little wiggly, one could also argue to make the estimates more smooth by increasing the bandwidth. This however will immediately increase the bias. Furthermore, we can see that in the lower left part the estimates already seem to be oversmoothed. This discrepancy is caused by the asymmetry of the Clayton density and can not be solved by a global bandwidth rule. Here, it seems as if a good balance between all these factors was found.

3.3 Beta kernels

An alternative approach follows the ideas of Chen (1999) for kernel density estimation on the unit line. The intuition is to use kernels whose support matches the bounded support of the density we want to estimate. Consequently, we will not use a different location for each data point, but rather vary the kernel shape for each point we want to estimate. An estimator of the copula density based on this idea (see also Charpentier et al., 2006) will be presented and a rule-of-thumb for bandwidth selection based on AMISE-optimality will be derived.

3.3.1 The estimator

For estimation on the unit square, we can simply use a product of such kernels. Chen (1999) suggests to use a family of densities corresponding to $Beta(p, q)$-distributed random variables as kernels, where the shape parameters vary with each data point. The resulting estimator of the copula density c can then be written as follows.



(a) $(u, v) = (0, 0)$ **(b)** $(u, v) = (0, 0.3)$ **(c)** $(u, v) = (0.3, 0.3)$

(d) $(u, v) = (0.8, 0.3)$ **(e)** $(u, v) = (0.5, 0.5)$ **(f)** $(u, v) = (0.8, 0.8)$

Figure 3.6: The shape of beta product kernels for different estimation points (u, v) for a fixed bandwidth.

Definition 3.3. *The **beta kernel estimator** of copula density $c(u, v)$ with bandwidth parameter b_n is given by*

$$c^{(\beta)}(u, v) = \frac{1}{n} \sum_{i=1}^{n} K\left(U_i, \frac{u}{b_n} + 1, \frac{1-u}{b_n} + 1\right) K\left(V_i, \frac{v}{b_n} + 1, \frac{1-v}{b_n} + 1\right),$$

where $K(x, p, q)$ is the density of a Beta(p, q)-distributed random variable evaluated at x, for all $(u, v) \in [0, 1]^2$.

Charpentier et al. (2006) claim (without proof) consistency and asymptotic normality for this type of estimator. From the boundedness of the beta density for this shape parameterization, it follows that this estimator will also produce bounded estimates.

Shapes of the product beta kernel for various points of estimation (u, v) are plotted in Figure 3.6. One can clearly see, how the shape adapts to the point, where the copula density should be estimated. For example, in panel (a) the density in the corner $(0, 0)$ should be estimated. Accordingly, observations that are not near this corner will have (almost) no contribution to the estimate, since the product kernel takes values close to zero in other regions apart from this corner. Similar conclusions can be drawn from the other panels. It is interesting to note that the size of the contribution of a particular data point to the estimate $\widehat{c}_n^{(\beta)}(u,v)$ does not only depend on the distance between the data point to (u,v), but also on the point (u,v) itself. This effect is similar to a decrease in bandwidth and leads to an increasing variance as the point of estimation comes closer to the boundaries. We will see this effect more clearly in the asymptotic expressions for bias and variance below.

3.3.2 Properties

First, we give asymptotic expressions for the bias and variance.

Proposition 3.2. *Let $c(u,v)$ be twice continuously differentiable on $(0,1)^2$, $b_n \to 0$ and $nb_n \to \infty$ as $n \to \infty$. Then for all $(u,v) \in (0,1)^2$,*

$$\begin{aligned}\mathrm{Bias}\big[\widehat{c}_n^{(\beta)}(u,v)\big] &= b_n\bigg[(1-2u)c_u(u,v) + (1-2v)c_v(u,v)\\ &\qquad + \frac{1}{2}u(1-u)c_{uu}(u,v) + \frac{1}{2}v(1-v)c_{vv}(u,v)\bigg] + o\big(b_n\big)\\ \mathrm{Var}\big[\widehat{c}_n^{(\beta)}(u,v)\big] &= \frac{1}{4nb_n\pi}\frac{c(u,v)}{\sqrt{u(1-u)v(1-v)}} + o\bigg(\frac{1}{nb_n}\bigg).\end{aligned}$$

Proof. We will follow some ideas developed by Chen (2000) for the analysis of univariate beta kernel smoothers for non-parametric regression. For sake of brevity we will not make the shape specifications of the kernel (which depend on u and v) explicit.

First, we consider the bias. We start by noting that due to independence of the samples the expectation of the beta kernel estimator can be written as

$$\begin{aligned}\mathrm{E}\big[\widehat{c}_n^{(\beta)}(u,v)\big] &= \mathrm{E}\bigg[\frac{1}{n}\sum_{i=1}^{n} K(U_i)K(V_i)\bigg]\\ &= \mathrm{E}\big[K(U)K(V)\big]\\ &= \int_0^1\int_0^1 K(s)K(t)c(s,t)dsdt\\ &= \mathrm{E}\big[c(M_u,M_v)\big],\end{aligned}$$

where $M_x \sim Beta(x/b_n+1,(1-x)/b_n+1)$, for $x=u,v$, are independent random variables. Next we take the Taylor series expansions of the mean and variance of M_x from (Chen, 2000, Appendix I) stating that uniformly in $(u,v)\in(0,1)^2$

$$\mathrm{E}[M_x] = x + b_n(1-2x) + O(b_n^2) \tag{3.1}$$

$$\mathrm{Var}[M_x] = b_n x(1-x) + O(b_n^2). \tag{3.2}$$

Denote $c_u = \partial c/\partial u$,$c_{uu} = \partial^2 c/\partial u^2$ and so on. Another Taylor expansion of the term $c(M_u, M_v)$ around (u, v) leads us to

$$\begin{aligned} c(M_u, M_v) = c(u, v) + c_u(u, v)(M_u - u) + c_v(u, v)(M_v - v)+ \\ + \frac{1}{2}\Big[c_{uu}(u, v)(M_u - u)^2 + c_{vv}(u, v)(M_v - v)^2\Big] + R, \end{aligned} \tag{3.3}$$

where, as usual, the remainder term is given by

$$\begin{aligned} R = \int_0^{M_u-u}\int_0^{M_v-v} \Bigg[&(M_u - u - s)\Big[c_{uu}(u + s, v + t) - c_{uu}(u, v)\Big] \\ &+ (M_v - v - s)\Big[c_{vv}(u + s, v + t) - c_{vv}(u, v)\Big]\Bigg] dsdt. \end{aligned}$$

Now define ψ_x as the density of the random variable $(M_x - x)/\sqrt{b_n}$. This gives

$$\begin{aligned} \mathrm{E}[R] = &\int\int \psi_u(p)\psi_v(q) \\ &\cdot \int_0^{\sqrt{b_n}p}\int_0^{\sqrt{b_n}q} \Bigg[(\sqrt{b_n}p - s)\Big[c_{uu}(u + s, v + t) - c_{uu}(u, v)\Big] \\ &\qquad + (\sqrt{b_n}q - s)\Big[c_{uu}(u + s, v + t) - c_{uu}(u, v)\Big]\Bigg] dsdt\, dpdq \\ = b_n &\int\int \psi_u(p)\psi_v(q) \\ &\cdot \int_0^{p}\int_0^{q} \Bigg[(p - s)\Big[c_{uu}(u + \sqrt{b_n}s, v + \sqrt{b_n}t) - c_{uu}(u, v)\Big] \\ &\qquad + (q - s)\Big[c_{uu}(u + \sqrt{b_n}s, v + \sqrt{b_n}t) - c_{uu}(u, v)\Big]\Bigg] dsdt\, dpdq \\ =: b_n&\epsilon_n(u, v). \end{aligned}$$

When c_{uu} and c_{vv} are uniformly continuous on $(0, 1)^2$, the differences in small brackets tend to zero uniformly, and $\epsilon_n(u, v)$ converges to zero uniformly. Thus, $E[R] = o(b_n)$ uniformly. Further, note that using the approximations (3.1) and (3.2), we have

$$\begin{aligned} \mathrm{E}\Big[(M_u - u)\Big] &= b_n(1 - 2u) + O(b_n^2) \\ &= b_n(1 - 2u) + o(b_n), \\ \mathrm{E}\Big[(M_u - u)^2\Big] &= \mathrm{E}\Big[\Big(M_u - \mathrm{E}[M_u] + \mathrm{E}[M_u] - u\Big)^2\Big] \\ &= \mathrm{E}\Big[(M_u - \mathrm{E}[M_u])^2\Big] + \mathrm{E}\Big[(\mathrm{E}[M_u] - u)^2\Big] \\ &\quad + 2\,\underbrace{\mathrm{E}\Big[M_u - \mathrm{E}[M_u]\Big]}_{=0}\,\mathrm{E}\Big[\mathrm{E}[M_u] - u\Big] \\ &\overset{(3.1)}{=} \mathrm{Var}[M_u] + b_n^2(1 - 2u)^2 + O(b_n^4) \\ &\overset{(3.2)}{=} b_n u(1 - u) + o(b_n). \end{aligned}$$

These expressions can now be plugged into the expectation of (3.3), yielding

$$\begin{aligned}
\mathrm{E}[c(M_u, M_v)] &= c(u,v) + c_u(u,v)\mathrm{E}\Big[(M_u - u)\Big] + c_v(u,v)\mathrm{E}\Big[(M_v - v)\Big] \\
&\quad + \frac{1}{2}\Big[c_{uu}(u,v)\mathrm{E}\Big[(M_u - u)^2\Big] + c_{vv}(u,v)\mathrm{E}\Big[(M_v - v)^2\Big]\Big] + E[R] \\
&= c(u,v) + c_u(u,v)b_n(1-2u) + c_v(u,v)b_n(1-2v) \\
&\quad + \frac{1}{2}\Big[c_{uu}(u,v)b_n u(1-u) + c_{vv}(u,v)b_n v(1-v)\Big] + o(b_n)
\end{aligned}$$

uniformly on $(0,1)$. By recalling that $\mathrm{E}[\widehat{c}_n^{(\beta)}(u,v)\Big] = \mathrm{E}[c(M_u, M_v)\Big]$, this concludes the proof for the bias.

Let us turn to the proof for the variance. In the following we write $K_w(s)$ shorthand for $K\Big(s, w/b_n + 1, (1-w)/b_n + 1\Big)$ and denote $\mathcal{B}$ as the beta function. First note that by the definition of the beta density it holds

$$\begin{aligned}
K_w^2(s) &= \left(\frac{1}{\mathcal{B}\Big(w/b_n + 1, (1-w)/b_n + 1\Big)} x^{w/b_n}(1-x)^{(1-w)/b_n}\right)^2 \\
&= \frac{1}{\mathcal{B}^2\Big(w/b_n + 1, (1-w)/b_n + 1\Big)} x^{2w/b_n}(1-x)^{2(1-w)/b_n} \\
&= \underbrace{\frac{\mathcal{B}\Big(2w/b_n + 1, 2(1-w)/b_n + 1\Big)}{\mathcal{B}^2\Big(w/b_n + 1, (1-w)/b_n + 1\Big)}}_{=:A(w)} K\Big(s, 2w/b_n + 1, 2(1-w)/b_n + 1\Big).
\end{aligned}$$

Let us introduce the notation $K_w^*(s) := K\Big(s, 2w/b_n + 1, 2(1-w)/b_n + 1\Big)$ and write M_u^*, M_v^* for two independent random variables with distributions K_u^* and K_v^* respectively. Then

$$\begin{aligned}
\mathrm{Var}[\widehat{c}_n^{(\beta)}(u,v)\Big] &= \mathrm{E}\Big[K_w^2(U)K_v^2(V)\Big] - \mathrm{E}\Big[K_w(U)K_v(V)\Big]^2 \\
&= A(u)A(v)\mathrm{E}\Big[K_u^*(U)K_v^*(V)\Big] + o(b_n) \\
&= A(u)A(v)\mathrm{E}\Big[c(M_u^*, M_V^*)\Big] + o(b_n) \\
&= A(u)A(v)\Big[c(u,v) + O(b_n)\Big] + o(b_n) \\
&= A(u)A(v)c(u,v) + O(b_n)
\end{aligned}$$

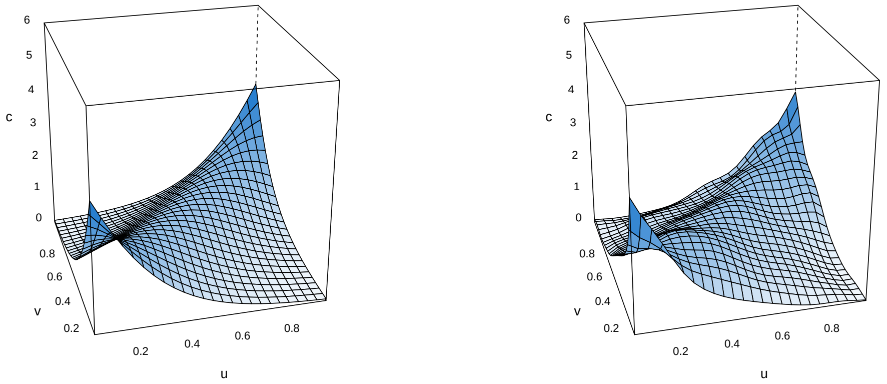
Lastly, we use the following result derived in Chen (2000): If $w/b_n \to \infty$ and $(1-w)/b_n \to \infty$ for $w = u, v$, it holds that

$$A(u)A(v) = \frac{1}{4b_n\pi}\frac{1}{\sqrt{u(1-u)v(1-v)}} + o\Big(b_n^{-1}\Big) \qquad \text{for } (u,v) \in (0,1)^2.$$

Then the variance can be written as

$$\begin{aligned}
\mathrm{Var}\Big[\widehat{c}_n^{(\beta)}(u,v)\Big] &= \frac{n}{n^2}\mathrm{Var}\Big[K_u(U)K_v(V)\Big] \\
&= \frac{1}{4nb_n\pi}\frac{c(u,v)}{\sqrt{u(1-u)v(1-v)}} + o\left(\frac{1}{nb_n}\right).
\end{aligned}$$

□



(a) True density **(b)** Beta kernel estimator

Figure 3.7: Perspective plots of the true density and a beta kernel estimate on simulated data ($n = 1\,000$) of a Frank copula with parameter $\theta = 5$ (Kendall's $\tau \approx 0.46$). The AMISE optimal bandwidth was used.

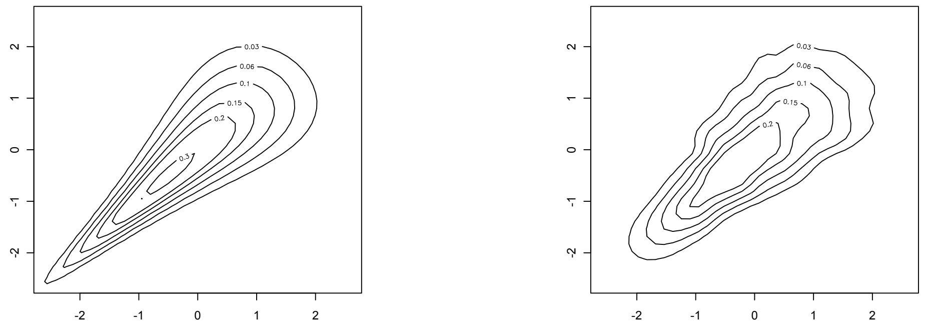
For the beta kernel estimator, the bias depends on the first and second order partial derivatives of the copula density. We can see that if all of them are bounded, the asymptotic bias will be bounded and actually tend to zero near the boundaries. As we already pointed out, the variance increases when u or v get closer to zero or one. The behavior here is very similar to the one we saw for the improved mirror-reflection estimator, c.f. Section 3.2.3.

In Figure 3.7 we compare the beta kernel estimator against the true density for the same sample from a Frank copula as in the previous section. The estimator is competitive with the estimators seen in Figure 3.4. It underestimates the upper tail a little and overestimates the upper tail a little. But since bias and variance are symmetric this is just a single sample effect and not a systematic flaw.

One should however be mindful of the fact that the Frank copula model is in a sense the best case scenario for the beta kernel estimator. This is because the true density in this case is bounded and has bounded first and second order partial derivatives, which is always the case for beta kernel estimates. Just as the simple mirror-reflection estimator, it will not adequately imitate the tail behavior in many other cases. A contour plot of the beta kernel estimate for the simulated Clayton data is shown in Figure 3.8. The shape of the estimate in the lower tail is too broad and the end too flat, although it notably improves over the basic mirror-reflection estimator.

3.3.3 Bandwidth selection

Our goal is again to find a rule-of-thumb for the choice of the bandwidth parameter b_n. Just as before, we will do this by minimizing the AMISE for a reference copula.



(a) True contours **(b)** Beta kernel estimator

Figure 3.8: Normal contour plots of the true density and the beta kernel estimate on simulated data ($n = 1\,000$) of a Clayton copula with parameter $\theta = 3$ (Kendall's $\tau = 0.6$). The bandwidth was selected based on AMISE-optimality w.r.t to the Frank copula.

Define

$$\begin{aligned} g(u, v) &:= (1 - 2u)c_u(u, v) + (1 - 2v)c_v(u, v) \\ &\quad + \frac{1}{2}u(1 - u)c_{uu}(u, v) + \frac{1}{2}v(1 - v)c_{vv}(u, v). \end{aligned}$$

Then the expressions of the previous section lead to

$$\text{AMISE}[\widehat{c}_n^{(\beta)}] = b_n^2 \underbrace{\int_0^1 \int_0^1 g_c(u, v)^2 dudv}_{=:\xi(c)} + \frac{1}{4nb_n\pi} \underbrace{\int_0^1 \int_0^1 \frac{c(u, v)}{\sqrt{u(1 - u)v(1 - v)}} dudv}_{=:\zeta(c)},$$

whenever the integrals exist. To ensure integrability of asymptotic squared bias and variance, we have to fulfill severals conditions. These are met in particular when the density c is bounded. In practice, a rule-of-thumb for the beta kernel estimator $\widehat{c}_n^{(\beta)}$ can therefore be given by

$$b_n = \left(\frac{1}{8\pi} \frac{\zeta(c)}{\xi(c)}\right)^{1/3} n^{-1/3},$$

where we take c as the Frank copula. Also here it seems as if this rule provides a good balance between the factors discussed at the end of Section 3.2.4.

3.4 Transformation estimator

Another approach is inspired by the early work of Devroye and Györfi (1985) and was used in the context of copula density estimation in Charpentier et al. (2006). It tackles the problems rising from the boundedness of the support by transforming the data so that its distribution is supported on the full $\mathbb{R}^2$. On the transformed data we can apply standard kernel estimation methods and then have to adequately back-transform the estimate to the original support. The most popular choice for the transformation is to apply the inverse of standard normal *cdf*, as it is established that the standard kernel estimator works well for approximately normally distributed random variables. It will therefore, be used from here on. In the following, we will systematically develop this idea in the simplest case and discuss extensions as well as the choice of smoothing parametrization.

3.4.1 The basic estimator

Let Φ be the standard normal *cdf* and ϕ its density. Then the random vector $(X, Y) = \big(\Phi^{-1}(U), \Phi^{-1}(V)\big)$ has normally distributed margins and is, in particular, supported on the full $\mathbb{R}^2$. By Sklar's Theorem (Theorem 2.1) its density f can be written as

$$f(x, y) = c\Big(\Phi(x), \Phi(x)\Big)\phi(x)\phi(y). \tag{3.4}$$

To estimate this density, we need a sample of the random vector (X, Y). Hence, put $(X_i, Y_i) = \big(\Phi^{-1}(U_i), \Phi^{-1}(V_i)\big)$ for $i = 1, \ldots, n$. To this transformed sample, we can now apply the standard estimator described in Section 2.2, which is

$$\widehat{f}_n(x, y) = \frac{1}{n}\sum_{i=1}^{n} K_{b_n}(x - X_i)K_{b_n}(y - Y_i), \qquad \text{for all } (x, y) \in \mathbb{R}^2.$$

By isolating c in (3.4) and using the estimator $\widehat{f}_n$ instead of f, we can then define an estimator for the copula density c.

Definition 3.4. *The* ***transformation estimator*** *of a copula density $c(u, v)$ with bandwidth parameter b_n is given by*

$$\widehat{c}_n^{(T)}(u, v) = \frac{\sum_{i=1}^{n} K_{b_n}\Big(\Phi^{-1}(u) - \Phi^{-1}(U_i)\Big)K_{b_n}\Big(\Phi^{-1}(v) - \Phi^{-1}(V_i)\Big)}{n\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))},$$

for all $(u, v) \in [0, 1]^2$.

By construction, this estimator of a copula density inherits all the pleasant properties of the usual kernel density estimator $\widehat{f}_n$.

An illustration of the process is given in Figure 3.9. We begin in the outer left panel with the sample of the copula, $(U_i, V_i)_{i=1,\ldots,n}$. In the next step the data was transformed by the standard normal quantile function Φ^{-1}. From the transformed samples, we obtain the usual kernel density estimate $\widehat{f}$, which is then divided by $\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))$ to obtain the copula density estimate $\widehat{c}_n^{(T)}$.

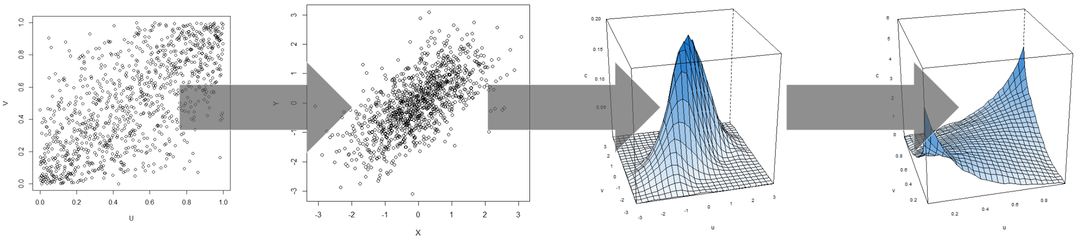


Figure 3.9: The transformation estimator. From left to right: Data sample, transformed sample, kernel density estimate for transformed sample and kernel density estimate for data sample.

3.4.2 Full bandwidth parameterization

Up until now, we just used the simplest form of bandwidth parameterization, namely one smoothing parameter for both components of the random vector (U, V). Of course, we can also specify two different bandwidths $b_{1n}, b_{2n} > 0$, one for each dimension. Now, define B_n as the diagonal matrix having b_{1n} as the upper and b_{2n} as the lower diagonal entries. By noting that $\det(B_n) = b_{1n}b_{2n}$ and putting $\boldsymbol{w} = (u, v)$ and $\boldsymbol{W}_i = (U_i, V_i)$ for all $i = 1, \ldots, n$, we can write this extension of the transformation estimator as

$$\widehat{c}_n^{(T')}(\boldsymbol{w}) = \frac{\sum_{i=1}^n K_{B_n}\Big(\Phi^{-1}(\boldsymbol{w}) - \Phi^{-1}(\boldsymbol{W}_i)\Big)}{n\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))},$$

where Φ^{-1} is applied componentwise to vectors and we used the short notation

$$K_{B_n}(\boldsymbol{w}) = \frac{K\bigg(\Big(B_n^{-1}[\Phi^{-1}(\boldsymbol{w}) - \Phi^{-1}(\boldsymbol{W}_i)]\Big)_1\bigg)K\bigg(\Big(B_n^{-1}[\ \Phi^{-1}(\boldsymbol{w}) - \Phi^{-1}(\boldsymbol{W}_i)]\Big)_2\bigg)}{\det(B_n)}$$

Now there is no reason to restrict ourselves to diagonal bandwidth matrices B_n. In fact, we will use any 2×2 matrix such that $B_n B_n^\top$ is a symmetric and positive definite matrix. This gives us a third parameter $b_{3n} \geq 0$, which we put below the diagonal B_n, yielding a fully parameterized version of the transformed estimator.

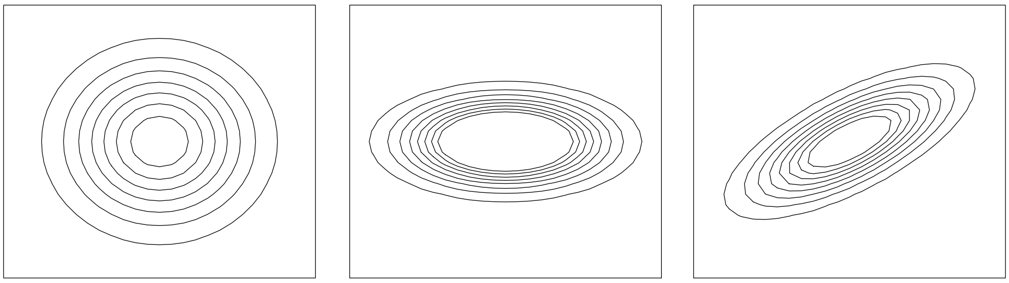
Definition 3.5. *The **fully parametrized transformation estimator** of a copula density $c(u, v)$ with bandwidth matrix*

$$B_n = \begin{pmatrix} b_{1n} & 0 \\ b_{3n} & b_{2n} \end{pmatrix}$$

is given by

$$c^{(TB)}(\boldsymbol{w}) = \frac{\sum_{i=1}^n K_{B_n}\Big(\Phi^{-1}(\boldsymbol{w}) - \Phi^{-1}(\boldsymbol{W}_i)\Big)}{n\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))},$$

for all $\boldsymbol{w} = (u, v) \in [0, 1]^2$.



(a) $B_n = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ (b) $B_n = \begin{pmatrix} 1 & 0 \\ 0 & 0.5 \end{pmatrix}$ (c) $B_n = \begin{pmatrix} 1 & 0 \\ 0.5 & 0.5 \end{pmatrix}$

Figure 3.10: The effect of using different bandwidth parameterizations for the Gaussian product kernel.

The effect of using more than one bandwidth parameter is most easily studied (following Wand and Jones, 1993) by reference to the Gaussian product kernel

$$\begin{aligned} K^p(\boldsymbol{x}) &= \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x_1^2\right\} \cdot \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x_2^2\right\} \\ &= \frac{1}{2\pi} \exp\left\{-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{x}\right\} \\ \Rightarrow K_{B_n}(\boldsymbol{x}) &= \frac{1}{2\pi \det(B_n)} \exp\left\{-\frac{1}{2}\boldsymbol{x}^\top (B_n B_n^\top)^{-1} \boldsymbol{x}\right\}. \end{aligned}$$

So the kernel K_{B_n} with bandwidth matrix B_n will simply be a bivariate Gaussian distribution with covariance matrix $C = B_n B_n^\top$. Plots of this kernel for the use of one, two and three different smoothing parameters respectively are given in Figure 3.10. Just one parameter will always give a kernel that is spherically symmetric; two different bandwidths, one for each dimension, will result in elliptical contours, but with elliptical axes parallel to the coordinate axes. Lastly, the inclusion of a third parameter will allow for elliptical kernels with arbitrary orientation.

3.4.3 Properties

Since the fully parameterized transformation estimator $\widehat{c}_n^{(TB)}$ is a generalization of the basic transformation estimator, let us state the more general expressions first.

Proposition 3.3. *Let $c(u, v)$ be twice continuously differentiable on $(0, 1)^2$ and $\det(B_n) \to 0$, $n \det(B_n) \to \infty$ as $n \to \infty$. Then for all $(u, v) \in (0, 1)^2$,*

$$\begin{aligned} \mathrm{Bias}\big[\widehat{c}_n^{(TB)}(u, v)\big] &= \frac{\sigma_K^2}{2} T\Big(\Phi^{-1}(u), \Phi^{-1}(v)\Big) + o\Big(\det(B_n)\Big), \\ \mathrm{Var}[\widehat{c}_n^{(TB)}(u, v)] &= \frac{d_K^2}{n \det(B_n)} \frac{c(u, v)}{\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))} + o\left(\frac{1}{n \det(B_n)}\right), \end{aligned}$$

where

$$\begin{aligned}
&T\Big(\Phi^{-1}(u), \Phi^{-1}(b)\Big) = \\
&\quad b_{1n}^2\Big[c_{uu}\Big(u,v\Big)\phi^2\Big(\Phi^{-1}(u)\Big) - 3c_u\Big(u,v\Big)\Phi^{-1}(u)\phi\Big(\Phi^{-1}(u)\Big) + c\Big(u,v\Big)(\Phi^{-1}(u)^2-1)\Big] \\
&+ (b_{2n}^2 + b_{3n}^2)\Big[c_{vv}\Big(u,v\Big)\phi^2\Big(\Phi^{-1}(b)\Big) - 3c_v\Big(u,v\Big)\Phi^{-1}(b)\phi\Big(\Phi^{-1}(b)\Big) + c\Big(u,v\Big)(\Phi^{-1}(b)^2-1)\Big] \\
&+ 2b_{1n}b_{3n}\Big[c\Big(u,v\Big)\Phi^{-1}(u)\Phi^{-1}(b) - c_u\Big(u,v\Big)\phi\Big(\Phi^{-1}(u)\Big)\Phi^{-1}(b) - c_v\Big(u,v\Big)\phi\Big(\Phi^{-1}(b)\Big)\Phi^{-1}(u) \\
&\qquad\qquad + c_{uv}\Big(u,v\Big)\phi\Big(\Phi^{-1}(u)\Big)\phi\Big(\Phi^{-1}(b)\Big)\Big],
\end{aligned}$$

and

$$\sigma_K^2 = \int_0^1 s^2K(s)ds \qquad \text{and } d_K = \int_0^1 K^2(s)ds,$$

Proof. We will use results obtained by Wand (1992) for the asymptotic bias and variance of the standard kernel estimator with full parameterization

$$\widehat{f}_n^{(H)}(\boldsymbol{x}) = \frac{1}{n\det B_n}\sum_{i=1}^n K^p(B_n^{-1}(\boldsymbol{x} - \boldsymbol{X}_i)),$$

with K^p being the product kernel of K. It states that if f is the true density,

$$\begin{aligned}
\operatorname{Bias}\Big[\widehat{f}_n^{(H)}(x,y)\Big] &= \frac{\sigma_K^2}{2}\operatorname{tr}\{B_nB_n^\top \operatorname{Hess}[f(x,y)]\} + o\Big(\det(B_n)\Big) \qquad \text{and} \\
\operatorname{Var}\Big[\widehat{f}_n^{(H)}(x,y)\Big] &= \frac{d_K^2}{n\det(B_n)}f(x,y) + o\bigg(\frac{1}{n\det(B_n)}\bigg).
\end{aligned}$$

By construction of the transformation estimator, we have

$$\widehat{c}_n^{(TB)}\Big(\Phi(x),\Phi(y)\Big) = \frac{\widehat{f}_n^{(H)}(x,y)}{\phi(x)\phi(y)}.$$

We will exploit this relationship in the following. The variance can easily be derived as

$$\begin{aligned}
\operatorname{Var}\Big[\widehat{c}_n^{(TB)}\Big(\Phi(x),\Phi(y)\Big)\Big] &= \frac{1}{\phi^2(x)\phi^2(y)}\operatorname{Var}[\widehat{f}(x,y)] \\
&= \frac{1}{\phi^2(x)\phi^2(y)}\bigg[\frac{d_K^2}{n\det(B_n)}f(x,y) + o\bigg(\frac{1}{n\det(B_n)}\bigg)\bigg] \\
&= \frac{d_K^2}{n\det(B_n)}\frac{c\Big(\Phi(x),\Phi(y)\Big)}{\phi(x)\phi(y)} + o\bigg(\frac{1}{n\det(B_n)}\bigg).
\end{aligned}$$

The bias is more complicated. Similarly, we start with

$$\begin{aligned}
\mathrm{E}\Big[\widehat{c}_n^{(TB)}\Big(\Phi(x),\Phi(y)\Big)\Big] &= \frac{1}{\phi(x)\phi(y)}\mathrm{E}\Big[\widehat{f}_n^{(H)}(x,y)\Big] \\
&= \frac{f\Big(x,y\Big)}{\phi(x)\phi(y)} + \frac{\sigma_K^2}{2}\frac{\mathrm{tr}\{B_n B_n^\top \mathrm{Hess}[f(x,y)]\}}{\phi(x)\phi(y)} + o\Big(\det(B_n)\Big) \\
&\approx c\Big(\Phi(x),\Phi(y)\Big) + \underbrace{\frac{\sigma_K^2}{2}\frac{\mathrm{tr}\{B_n B_n^\top \mathrm{Hess}[f(x,y)]\}}{\phi(x)\phi(y)}}_{T(x,y)}.
\end{aligned}$$

Next, we want to make the expression for the trace more explicit. This will be done in several steps.

We start with the calculation of the terms in the Hessian matrix of $f(x,y)$. We define $c_u = \partial c(u,v)/\partial u$, $c_v = \partial c(u,v)/\partial v$, $c_{uu} = \partial^2 c(u,v)/\partial^2 u, \ldots$ as the partial derivatives of the copula density c w.r.t. to its arguments and denote the derivative of ϕ by ϕ'. Note that

$$\begin{aligned}
\phi'(x) &= -x\phi(x), \\
\phi''(x) &= (x^2-1)\phi(x), \\
\frac{\partial c\Big(\Phi(x),\Phi(y)\Big)}{\partial x} &= c_u\Big(\Phi(x),\Phi(y)\Big)\phi(x), \\
\frac{\partial^2 c\Big(\Phi(x),\Phi(y)\Big)}{\partial^2 x} &= c_{uu}\Big(\Phi(x),\Phi(y)\Big)\phi^2(x) + c_u\Big(\Phi(x),\Phi(y)\Big)\phi'(x) \\
&= c_{uu}\Big(\Phi(x),\Phi(y)\Big)\phi^2(x) - c_u\Big(\Phi(x),\Phi(y)\Big)x\phi(x).
\end{aligned}$$

and recall that $f(x,y) = c\Big(\Phi(x),\Phi(y)\Big)\phi(x)\phi(y)$. By the product rule for differentiation we can then obtain the first order derivative

$$\begin{aligned}
\frac{\partial f(x,y)}{\partial x} &= \phi(y)\Big[c_u\Big(\Phi(x),\Phi(y)\Big)\phi^2(x) + c\Big(\Phi(x),\Phi(y)\Big)\phi'(x)\Big] \\
&= \phi(y)\Big[c_u\Big(\Phi(x),\Phi(y)\Big)\phi^2(x) - c\Big(\Phi(x),\Phi(y)\Big)x\phi(x)\Big].
\end{aligned}$$

and the second order partial derivative

$$\begin{aligned}
\frac{\partial^2 f(x,y)}{\partial x^2} &= \phi(y)\Big[c_{uu}\Big(\Phi(x),\Phi(y)\Big)\phi^3(x) + 3c_u\Big(\Phi(x),\Phi(y)\Big)\phi(x)\phi'(x) \\
&\qquad + c\Big(\Phi(x),\Phi(y)\Big)\phi''(x)\Big] \\
&= \phi(y)\Big[c_{uu}\Big(\Phi(x),\Phi(y)\Big)\phi^3(x) - 3c_u\Big(\Phi(x),\Phi(y)\Big)x\phi^2(x)x \\
&\qquad + c\Big(\Phi(x),\Phi(y)\Big)(x^2-1)\phi(x)\Big].
\end{aligned}$$

The corresponding derivatives w.r.t the second argument, y, are analogue. The mixed partial derivatives can be calculated as

$$\begin{aligned}\frac{\partial^2 f(x,y)}{\partial x \partial y} &= c\Big(\Phi(x),\Phi(y)\Big)\phi'(x)\phi'(y) + c_{uv}\Big(\Phi(x),\Phi(y)\Big)\phi^2(x)\phi^2(y)\\ &\quad + c_u\Big(\Phi(x),\Phi(y)\Big)\phi^2(x)\phi'(y) + c_v\Big(\Phi(x),\Phi(y)\Big)\phi^2(y)\phi'(x)\\ &= c\Big(\Phi(x),\Phi(y)\Big)xy\phi(x)\phi(y) + c_{uv}\Big(\Phi(x),\Phi(y)\Big)\phi^2(x)\phi^2(y)\\ &\quad - c_u\Big(\Phi(x),\Phi(y)\Big)\phi^2(x)y\phi(y) - c_v\Big(\Phi(x),\Phi(y)\Big)\phi^2(y)x\phi(x)\end{aligned}$$

As a next step we calculate

$$B_n B_n^\top = \begin{pmatrix} b_{1n} & 0 \\ b_{3n} & b_{2n}\end{pmatrix}\begin{pmatrix} b_{1n} & b_{3n} \\ 0 & b_{2n}\end{pmatrix} = \begin{pmatrix} b_{1n}^2 & b_{1n}b_{3n} \\ b_{1n}b_{3n} & b_{2n}^2 + h_{3n}^2\end{pmatrix}.$$

Altogether this results in

$$\begin{aligned}T(x,y) &= \frac{\operatorname{tr}\{B_n B_n^\top \operatorname{Hess}[f(x,y)]\}}{\phi(x)\phi(y)}\\ &= \frac{1}{\phi(x)\phi(y)}\left[b_{1n}^2\frac{\partial^2 f(x,y)}{\partial^2 x} + (b_{2n}^2 + b_{3n}^2)\frac{\partial^2 f(x,y)}{\partial^2 y} + 2b_{1n}b_{3n}\frac{\partial^2 f(x,y)}{\partial x\partial y}\right]\\ &= b_{1n}^2\left[c_{uu}\Big(\Phi(x),\Phi(y)\Big)\phi^2(x) - 3c_u\Big(\Phi(x),\Phi(y)\Big)x\phi(x) + c\Big(\Phi(x),\Phi(y)\Big)(x^2-1)\right]\\ &\quad + (b_{2n}^2 + b_{3n}^2)\left[c_{vv}\Big(\Phi(x),\Phi(y)\Big)\phi^2(y) - 3c_v\Big(\Phi(x),\Phi(y)\Big)y\phi(y)\right.\\ &\qquad \left. + c\Big(\Phi(x),\Phi(y)\Big)(y^2-1)\right]\\ &\quad + 2b_{1n}b_{3n}\left[c\Big(\Phi(x),\Phi(y)\Big)xy - c_u\Big(\Phi(x),\Phi(y)\Big)\phi(x)y - c_v\Big(\Phi(x),\Phi(y)\Big)\phi(y)x\right.\\ &\qquad \left. + c_{uv}\Big(\Phi(x),\Phi(y)\Big)\phi(x)\phi(y)\right].\end{aligned}$$

The change of variables $u = \Phi(x), v = \Phi(y)$ finally gives the result. □

As a direct consequence we also get asymptotic expressions for the one-parameter transformation estimator.

Corollary 3.4. *Let $c(u,v)$ be twice continuously differentiable on $(0,1)^2$ and $b_n \to 0$, $nb_n^2 \to \infty$ as $n \to \infty$. Then for all $(u,v) \in (0,1)^2$,*

$$\begin{aligned}\mathrm{Bias}\big[\widehat{c}_n^{(T)}(u,v)\big] = \frac{\sigma_K^2}{2}b_n^2\Big[& c_{uu}\big(u,v\big)\phi^2\big(\Phi^{-1}(u)\big) + c_{vv}\big(u,v\big)\phi^2\big(\Phi^{-1}(v)\big) \\ & - 3c_u\big(u,v\big)\phi\big(\Phi^{-1}(u)\big)\Phi^{-1}(u) - 3c_v\big(u,v\big)\phi\big(\Phi^{-1}(v)\big)\Phi^{-1}(v) \\ & + c\big(u,v\big)\Big[\big(\Phi^{-1}(u)\big)^2 + \big(\Phi^{-1}(v)\big)^2 - 2\Big]\Big] + o\big(b_n^2\big), \\ \mathrm{Var}[\widehat{c}_n^{(T)}(u,v)] = \frac{d_K^2}{nb_n^2}&\frac{c(u,v)}{\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))} + o\left(\frac{1}{nb_n^2}\right),\end{aligned}$$

where

$$\sigma_K^2 = \int_0^1 s^2K(s)ds \qquad \text{and } d_K = \int_0^1 K^2(s)ds.$$

Proof. Follows directly from Proposition 3.3 by noting that for the case of just a single bandwidth parameter, $b_n = b_{1n} = b_{2n}$, $b_{3n} = 0$ and $\det(B_n) = b_n^2$. □

In this case the bias expression is much simpler and transparent. Hence, we will illustrate the asymptotic behavior on the basis of the simplest transformation-estimator with just a single bandwidth parameter b_n. The effects we discuss will carry over to the fully parameterized estimator.

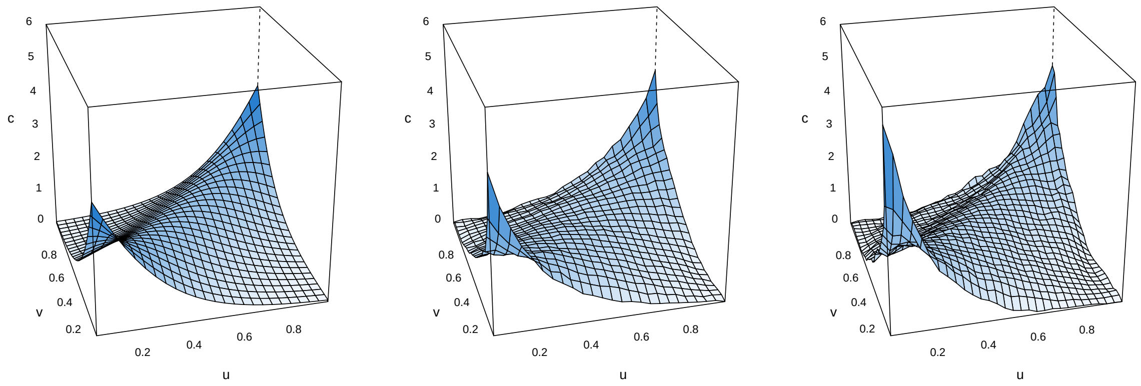
Let us first look at an asymptotic approximation of the bias, which is still quite complex. To simplify matters, let us assume that c and its second order partial derivatives are bounded, which is one of the more convenient situations in practice. Note that, if u tends to zero or one, $\phi(\Phi^{-1}(u)) \to 0$ and $\phi(\Phi^{-1}(u))\Phi^{-1}(u) \to 0$. This will make all terms in the first and second line of the brackets vanish near the boundaries. The term in the last line, however, will explode towards the boundaries even if c is bounded. This reveals that the bias is always unbounded near the boundaries of the unit square.

The asymptotic variance will also grow unboundedly, even when c is bounded, since $\phi(\Phi^{-1}(u)) \to 0$ at the boundaries. Actually, it is not even integrable making bandwidth selection based on AMISE-optimality infeasible. This can easily be seen by substituting $u = \Phi(x)$ in the following integral:

$$\int_0^1 \frac{1}{\phi(\Phi^{-1}(u))}du = \int_{-\infty}^{\infty} \frac{\phi(x)}{\phi(\Phi^{-1}(\Phi(x)))}dx = \int_{-\infty}^{\infty} \frac{\phi(x)}{\phi(x)}dx = \infty.$$

In contrast, when estimating the density of the transformed sample by $\widehat{f}_n^{(H)}$, the asymptotic variance tends to zero when moving to the outer regions. This indicates that the transformation is to be blamed for the exploding variances. To see why, let us consider the Taylor approximation of $\Phi^{-1}(U_i)$ in u:

$$\begin{aligned}\Phi^{-1}(U_i) &\approx \Phi^{-1}(u) + (U_i - u)(\Phi^{-1}(u))' \\ &= \Phi^{-1}(u) + \frac{(U_i - u)}{\phi(\Phi^{-1}(u))}.\end{aligned}$$



(a) True density

(b) Transformation estimator (T)

(c) Fully parameterized transformation estimator (TB)

Figure 3.11: Perspective plots of the true density and the transformation estimator on simulated data ($n = 1\,000$) of a Frank copula with parameter $\theta = 5$ (Kendall's $\tau \approx 0.46$). The bandwidth was selected based on the reference-rules discussed in Section 3.4.4.

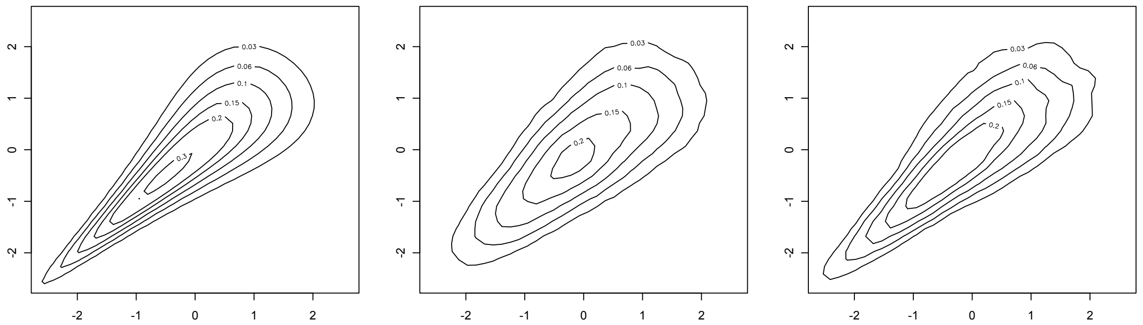
Using this approximation in the formula of the estimator yields

$$\begin{aligned}\widehat{c}_n^{(T)}(u,v) &\approx \frac{1}{n\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))}\sum_{i=1}^{n} K_{b_n}\left(\frac{U_i-u}{\phi(\Phi^{-1}(u))}\right)K_{b_n}\left(\frac{V_i-v}{\phi(\Phi^{-1}(v))}\right)\\ &= \frac{1}{n}\sum_{i=1}^{n} K_{b_n\phi(\Phi^{-1}(u))}\Big(U_i-u\Big)K_{b_n\phi(\Phi^{-1}(v))}\Big(V_i-v\Big).\end{aligned}$$

The transformation is therefore approximately equivalent to improving the naive estimator by imposing a bandwidth that adapts to the location of the estimate. This effective bandwidth will decrease towards the boundaries, since here $\phi(\Phi^{-1}(w))$ tends to zero. This is similar to the improvement of the mirror-reflection estimator.

Evidence of the discussed effects can be observed on the simulated sample of the Frank copula (see Figure 3.11). Both transformation estimators tend to overestimate the true density in the corners $(0,0)$ and $(1,1)$. Actually, the plot only contains estimates for $(1/26, 1/26)$ and $(25/26, 25/26)$ as the corner points. In fact, both estimators grow unboundedly when coming closer to $(0,0)$ and $(1,1)$. There is a hint that the use of a full bandwidth matrix helps the estimator adapting to the true shape in the inner regions, although this effect is quite small.

More insight can be gained from marginal normal contour plots corresponding to estimates on the Clayton sample (see Figure 3.12). The estimate with just a single bandwidth parameter is not able to properly imitate the narrow shape of the true contours. The fully parameterized version on the other hand is strikingly good. It imitates the shape of the true model very well — even in the lower tail. In both types of plots, we can also observe that both estimators show a little more spurious fluctuation on the boundaries. This is because the effective bandwidth in



(a) True contours

(b) Transformation estimator (T)

(c) Fully parameterized transformation estimator (TB)

Figure 3.12: Marginal normal contour of the true density and the beta kernel estimate on simulated data ($n = 1\,000$) of a Clayton copula with parameter $\theta = 3$ (Kendall's $\tau = 0.6$). The bandwidth was selected based on a scale rule for the bivariate normal distribution (see next section for details).

these regions is very small, which gives the estimates a higher variance due to the high peak of the kernel.

3.4.4 Bandwidth selection

Since the asymptotic expressions do not allow for bandwidth choice by minimization of the AMISE, a practical approach is to consider the AMISE on the level of the transformed data instead. Elegant results can be obtained considering the Gaussian copula as a reference copula. For using just one bandwidth parameter b_n, we can then use the optimal bandwidth for estimating the bivariate normal distribution with no correlation derived in Henderson and Parmeter (2011). In our setting this results in

$$b_n = c_K n^{-1/6}, \qquad \text{with } c_K := \left[\frac{4\pi d_K^2}{\sigma_K^4}\right]^{1/6}.$$

This rule does not adapt to the strength of dependence in any way. As we have seen in Figure 3.12, this may lead to an oversmoothed estimate when the data is dependent. Note that for the Gaussian kernel the constant c_K equals one. Hence, it can be interpreted as the ratio of optimal bandwidths for the kernel K and the Gaussian kernel.

When we want to use a fully parameterized bandwidth matrix instead, a very appealing and elegant result can be found in Wand and Jones (1993). It states that, if K is a Gaussian kernel and f is a bivariate Gaussian density with covariance matrix Σ, the AMISE-optimal bandwidth matrix is $B_n = \Sigma^{1/2} n^{-1/6}$, where $\Sigma^{1/2}$ involves the matrix root defined via $\Sigma^{1/2}(\Sigma^{1/2})^\top = \Sigma$. This is very intuitive, as it simply states that the covariance matrix of the kernel should be the same as the covariance matrix of the Gaussian density we want to estimate. Since we transform the margins to be standard normal, the 'true' variance of the data will be exactly one in each direction and the covariance matrix coincides with the correlation matrix Γ. For the use of other kernels, we suggest to multiply the resulting bandwidth matrix by the constants c_K defined above. This results in the rule-of-thumb

$$B_n = c_K \Gamma^{1/2} n^{-1/6}.$$

We see that, in case Γ is not the identity matrix, $\det(B_n^{opt}) < (b_n^{opt})^2$ and notice that these two quantities play the same role in their respective estimators. Equality between them implies that for a fixed kernel K, also the height of the 'bumps' placed on a particular sample point as well as the area of their contours will be equal for both estimators. A smaller value on the left hand side implies that there is less smoothing for the fully parameterized version. This is because the bandwidth rule adapts to the strength of dependence in the data and makes the estimates less smooth when the dependence in the data is stronger.

It should be stressed that, although B_n implicitly made reference to a Gaussian copula, it does not mean that it gives an AMISE-optimal bandwidth for the Gaussian copula density. The optimization is carried out on a density with normally distributed margins. Nonetheless, it worked very well for the two exemplary samples discussed in the last section (see Figures 3.11 and 3.12).

3.5 Local likelihood transformation estimator

An extension of the transformation estimator was recently suggested by Geenens et al. (2014). Instead of applying the standard kernel estimator, they locally fit a polynomial to the log-density of the transformed sample. This is in fact a generalization of the standard kernel estimator, since the latter emerges when fitting a polynomial of order zero. Furthermore, a nearest-neighbor type bandwidth will be used.

3.5.1 The estimator

The idea behind the local likelihood method (c.f. Loader, 1999) is to assume that the log-density $\log f(x,y)$ of the random vector $\boldsymbol{Z} = (X,Y) = \big(\Phi^{-1}(U), \Phi^{-1}(V)\big)$ can locally be approximated by a polynomial $P_{\boldsymbol{a}(x,y),p}$ of some order p. Here, $\boldsymbol{a}(x,y) \in \mathbb{R}^{(p+1)(p+2)/2}$ is the coefficient vector of the polynomial, where $(p+1)(p+2)/2$ is just the number of terms (including a constant) of a two-dimensional polynomial of order p. We can then write

$$\begin{aligned}\log f(x',y') \approx P&_{\boldsymbol{a}(x,y),p}\\ &= a_1(x,y) + a_2(x,y)(x-x') + a_3(x,y)(y-y')\\ &\quad + a_4(x,y)(x-x')^2 + a_5(x,y)(x-x')(y-y') + a_6(x,y)(y-y')^2\\ &\quad + \dots\\ &\quad + a_{(p+1)(p+2)/2}(x,y)(y-y')^p,\end{aligned}$$

for (x', y') in the neighborhood of (x, y). In order to fit the local coefficients to the log-density around $\boldsymbol{z} = (x, y)$, we can solve the weighted maximum likelihood problem

$$\widehat{\boldsymbol{a}}(x, y) = \arg \max_{\boldsymbol{a} \in \mathbb{R}^{(p+1)(p+2)/2}} \left\{ \sum_{i=1}^{n} K_B\Big(\boldsymbol{z} - \boldsymbol{Z}_i\Big) P_{\boldsymbol{a},p}(\boldsymbol{z} - \boldsymbol{Z}_i) \right.$$
$$\left. - n \int_{\mathbb{R}^2} K_B\Big(\boldsymbol{z} - \boldsymbol{s}\Big) \exp\Big(P_{\boldsymbol{a},p}(\boldsymbol{z} - \boldsymbol{s})\Big) d\boldsymbol{s} \right\},$$

where similar to the last section,

$$K_B(\boldsymbol{x}) := K\Big(\big(B^{-1}(\boldsymbol{x})\big)_1\Big) K\Big(\big(B^{-1}(\boldsymbol{x})\big)_2\Big),$$

with some symmetric probability density K and bandwidth matrix B. As a result, we obtain $\widehat{a}_1(x, y)$ as an estimate for $\log f(x, y)$ and, consequently, $\exp\Big(\widehat{a}_1(x, y)\Big)$ as an estimate for $f(x, y)$.

The definition of the likelihood function above is a bit unusual. While the kernel functions enter the formula in order to localize the estimation problem, the integral term serves as a penalty which becomes necessary due to the fact that, in general, the polynomial is not a density (see Loader, 1999, for a more thorough motivation). Since we known that the distributions of X and Y are standard normal, we might want our density estimates to behave locally like the bivariate Gaussian density $\phi(\boldsymbol{x}) = (2\pi)^{-1} \exp(-\frac{1}{2}\boldsymbol{x}^\top \Sigma^{-1} \boldsymbol{x})$. But this is equivalent to letting the log-density behave like a (constrained) polynomial of order two. Therefore, we will always use the order $p = 2$ from here on.

An estimate of the copula density can be obtained by rescaling the estimates just as we did in the previous section. As we have seen, this back-transformation causes the variance to explode near the boundaries. Geenens et al. (2014) suggest to use a variable bandwidth of nearest-neighbor type to stabilize the variance. The only difference here is that a constant bandwidth matrix B is multiplied by a quantity $\Delta_{k_n}(x, y)$, which is defined as the Euclidean distance between (x, y) and its k_nth closest sample point (X_i, Y_i) (w.r.t. the Euclidean norm). More formally, define $d_i(x, y) := \|(x, y) - (X_i, Y_i)\|$ and let $d_{(k)}(x, y)$ be the k-smallest amongst all $d_i(x, y)$, $i = 1, \ldots, n$. Then,

$$\Delta_{k_n}(x, y) = d_{(k_n)}(x, y).$$

When the kernel function is supported on $[-1, 1]$ and $B = I_d$, this means that exactly the k_n closest samples (X_i, Y_i) are used for estimation. Note that $\Delta_{k_n}(x, y)$ is random and gives a large bandwidth in areas of low density and a small bandwidth in areas of high density. Recall that $X_i, Y_i \sim \mathcal{N}(0, 1)$ and, thus, the density of the underlying samples (X_i, Y_i) looks roughly like a bivariate normal density. As a consequence, the bandwidth will increase when moving away from $(0, 0)$. When back-transformed to the unit square this corresponds to an increased bandwidth when approaching the boundaries. This will help to stabilize the variance in these regions, as we will see in the next section.

Remark 3.1. *It is reasonable to ask why we did not use nearest-neighbor bandwidths for the previous estimators. It is a well known fact that nearest-neighbor bandwidths do not work well with regular kernel estimators. The resulting estimates are usually too rough and the bandwidth does not adapt properly to the local conditions. When imposing a nearest-neighbor bandwidth in the context of local likelihood fitting however, these pitfalls disappear as the bandwidth is incorporated only indirectly through the local likelihood (see e.g. Simonoff, 1996, Section 3.4).*

Let us subsume the previously developed ideas in a formal definition.

Definition 3.6. *Define $\boldsymbol{W}_i = (U_i, V_i)$ and $\Delta_{k_n}(x, y)$ as the euclidean distance between (x, y) and its k_nth closest observation amongst all $\big(\Phi^{-1}(\boldsymbol{W}_i)\big)_{i=1,\dots,n} := \big(\Phi^{-1}(U_i), \Phi^{-1}(V_i))\big)_{i=1,\dots,n}$. For all $(u, v) \in [0, 1]^2$, the* ***local likelihood transformation estimator*** *of a copula density $c(u, v)$ with nearest-neighbor factor Δ_{k_n} and bandwidth matrix B is given by*

$$\widehat{c}_n^{(TLL)}(u, v) = \frac{\exp\Big\{\widehat{a}_1\Big(\Phi^{-1}(u), \Phi^{-1}(v)\Big)\Big\}}{\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))},$$

where $\widehat{a}_1\Big(\Phi^{-1}(u), \Phi^{-1}(v)\Big)$ can be found via

$$\begin{aligned}
&\widehat{\boldsymbol{a}}(x, y) = \\
&\operatorname*{arg\,max}_{\boldsymbol{a}\in\mathbb{R}^6}\Bigg\{\sum_{i=1}^{n} K_{\Delta_{k_n}(x,y)B}\Big((x, y) - \Phi^{-1}(\boldsymbol{W}_i)\Big)P_{\boldsymbol{a},2}\Big((x, y) - \Phi^{-1}(\boldsymbol{W}_i)\Big) \\
&\qquad - n\int_{\mathbb{R}^2} K_{\Delta_{k_n}(x,y)B}\Big((x, y)) - \Phi^{-1}(\boldsymbol{z})\Big) \exp\Big\{P_{\boldsymbol{a},2}\Big((x, y) - \Phi^{-1}(\boldsymbol{z})\Big)\Big\}d\boldsymbol{z}\Bigg\},
\end{aligned}$$

with

$$\begin{aligned}
P_{\boldsymbol{a}(x,y),2}(x', y') &= a_1(x, y) + a_2(x, y)(x - x') + a_3(y - y') \\
&\quad + a_4(x - x')^2 + a_5(x - x')(y - y') + a_6(y - y')^2.
\end{aligned}$$

Note that we used a constant bandwidth matrix in this case, since the ultimate amount of smoothing is determined by the nearest-neighbor term $\Delta_{k_n}(x, y)$, which already depends on n.

3.5.2 Properties

For the local likelihood transformation estimator the bias gets even more complicated, so we will restrict ourselves directly to the case of just one bandwidth parameter.

Proposition 3.5. *Let $c(u, v)$ be four times continuously differentiable on $(0, 1)^2$ and let $k_n \to \infty$, $k_n/n \to 0$ and, more specifically, $k_n = O(n^{4/5})$ as $n \to \infty$. Then for all*

$(u, v) \in (0,1)^2$,

$$\begin{aligned}
\operatorname{Bias}\big[\widehat{c}_n^{(TLL)}(u,v)\big] = &-\frac{\sigma_K^2 (k_n/n)^2}{8\pi^2 c\Big(u,v\Big)\phi^2\Big(\Phi^{-1}(u)\Big)\phi^2\Big(\Phi^{-1}(v)\Big)} \\
&\cdot \left\{ \frac{\partial^4 g}{\partial x^4} + \frac{\partial^4 g}{\partial y^4} + 2\frac{\partial^4 g}{\partial x^2 \partial y^2} \right. \\
&\left. + 4\left(\frac{\partial^3 g}{\partial x^3}\frac{\partial g}{\partial x} + \frac{\partial^3 g}{\partial y^3}\frac{\partial g}{\partial y} + \frac{\partial^3 g}{\partial x^2 \partial y}\frac{\partial g}{\partial y} + \frac{\partial^3 g}{\partial x \partial y^2}\frac{\partial g}{\partial x} \right) \right\}(x,y) \\
&+ o\left(\frac{k_n^2}{n^2}\right), \\
\operatorname{Var}\Big[\widehat{c}_n^{(TLL)}(u,v)\Big] = &\frac{5 d_K^2 \pi}{2k_n} c^2(u,v) + o\Big(k_n^{-1}\Big),
\end{aligned}$$

where

$$\sigma_K^2 = \int_0^1 s^2 K(s) ds \qquad \textit{and } d_K = \int_0^1 K^2(s) ds,$$

$x = \Phi^{-1}(u)$, $y = \Phi^{-1}(v)$ *and* $g(x,y) = \log c\Big(\Phi(x), \Phi(y)\Big) + \log\phi(x) + \log\phi(y)$.

For the proof, we will need a result from (Geenens et al., 2014, Theorem 3.3). Denote $\widehat{c}_n^{(TLL^*)}$ as the local likelihood transformation estimator without nearest-neighbor bandwidth and let us again assume that we work with just one parameter b_n, i.e. $\Delta_{k_n} \equiv 1$ and $B = b_n$.

Proposition 3.6. *Let $c(u,v)$ be be four times continuously differentiable on $(0,1)^2$ and let $b_n \to 0$, $nb_n^6 \to \infty$ as $n \to \infty$. Then for all $(u,v) \in (0,1)^2$,*

$$\begin{aligned}
\operatorname{Bias}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big] = &-\frac{\sigma_K^2 b_n^4}{8} c\Big(u,v\Big) \\
&\cdot \left\{ \frac{\partial^4 g}{\partial x^4} + \frac{\partial^4 g}{\partial y^4} + 2\frac{\partial^4 g}{\partial x^2 \partial y^2} \right. \\
&\left. + 4\left(\frac{\partial^3 g}{\partial x^3}\frac{\partial g}{\partial x} + \frac{\partial^3 g}{\partial y^3}\frac{\partial g}{\partial y} + \frac{\partial^3 g}{\partial x^2 \partial y}\frac{\partial g}{\partial y} + \frac{\partial^3 g}{\partial x \partial y^2}\frac{\partial g}{\partial x} \right) \right\}(x,y) \\
&+ o\Big(b_n^4\Big),
\end{aligned} \tag{3.5}$$

$$\operatorname{Var}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big] = \frac{5 d_K^2}{2nb_n^2} \frac{c(u,v)}{\phi\Big(\Phi^{-1}(u)\Big)\phi\Big(\Phi^{-1}(v)\Big)} + o\left(\frac{1}{nb_n^2}\right), \tag{3.6}$$

where

$$\sigma_K^2 = \int_0^1 s^2 K(s) ds \qquad \textit{and } d_K = \int_0^1 K^2(s) ds,$$

$x = \Phi^{-1}(u)$, $y = \Phi^{-1}(v)$ *and* $g(x,y) = \log c\Big(\Phi(x), \Phi(y)\Big) + \log\phi(x) + \log\phi(y)$.

Under additional conditions, (Geenens et al., 2014) show that the same expressions hold even if we are not provided with *iid* copies (U_i, V_i) of the copula and they have to be estimated in the sense of a transformation to pseudo-samples (c.f. Definition 2.8). This can be achieved by making of use of asymptotic theory for the empirical copula process (see e.g. Segers, 2012).

Proof of Proposition 3.5. Thanks to Proposition 3.6, it only remains to show how the use of a nearest-neighbor bandwidth affects the expressions (3.5) and (3.6). We utilize a version of the asymptotic approximation of $\Delta_{k_n}(x,y)$'s moments derived in Mack and Rosenblatt (1979). In our setting we can write

$$\mathrm{E}\Big[\Delta_{k_n}^{\lambda}(x,y)\Big] = \left(\frac{k_n/n}{\pi c\Big(\Phi(x),\Phi(y)\Big)\phi(x)\phi(y)}\right)^{\lambda/2} + o\Big((k_n/n)^{\lambda/2}\Big), \tag{3.7}$$

for all $\lambda \in \mathbb{Z}$. In the following we will omit the arguments of Δ_{k_n} for notational convenience.

Let us start with the bias. First, note that

$$\begin{aligned}\mathrm{Bias}\Big[\widehat{c}_n^{(TLL)}(u,v)\Big] &= \mathrm{E}\Big[\widehat{c}_n^{(TLL)}(u,v) - c(u,v)\Big]\\ &= \mathrm{E}\left[\mathrm{E}\Big[\widehat{c}_n^{(TLL)(u,v)}\Big|\Delta_{k_n}\Big] - c(u,v)\right]\\ &= \mathrm{E}\left[\mathrm{Bias}\Big[\widehat{c}_n^{(TLL)(u,v)}\Big|\Delta_{k_n}\Big]\right]\end{aligned}$$

and that $\mathrm{Bias}\Big[\widehat{c}_n^{(TLL)}(u,v)\Big|\Delta_{k_n}\Big]$ is exactly (3.5) where we replace b_n by Δ_{k_n}. Using formula (3.7) we can therefore write,

$$\begin{aligned}\mathrm{Bias}\Big[\widehat{c}_n^{(TLL)}(u,v)\Big] &= -\frac{\sigma_K^2\mathrm{E}\Big[\Delta_{k_n}^4\Big]}{8}c\Big(u,v\Big)\\ &\quad\cdot\left\{\frac{\partial^4 g}{\partial x^4} + \frac{\partial^4 g}{\partial y^4} + 2\frac{\partial^4 g}{\partial x^2\partial y^2}\right.\\ &\qquad\left. + 4\left(\frac{\partial^3 g}{\partial x^3}\frac{\partial g}{\partial x} + \frac{\partial^3 g}{\partial y^3}\frac{\partial g}{\partial y} + \frac{\partial^3 g}{\partial x^2\partial y}\frac{\partial g}{\partial y} + \frac{\partial^3 g}{\partial x\partial y^2}\frac{\partial g}{\partial x}\right)\right\}(x,y)\\ &\quad + o\Big(\mathrm{E}\Big[\Delta_{k_n}^4\Big]\Big)\\ &\overset{(3.7)}{=} -\frac{\sigma_K^2(k_n/n)^2}{8\pi^2 c\Big(u,v\Big)\phi^2\Big(\Phi^{-1}(u)\Big)\phi^2\Big(\Phi^{-1}(v)\Big)}\\ &\quad\cdot\left\{\frac{\partial^4 g}{\partial x^4} + \frac{\partial^4 g}{\partial y^4} + 2\frac{\partial^4 g}{\partial x^2\partial y^2}\right.\\ &\qquad\left. + 4\left(\frac{\partial^3 g}{\partial x^3}\frac{\partial g}{\partial x} + \frac{\partial^3 g}{\partial y^3}\frac{\partial g}{\partial y} + \frac{\partial^3 g}{\partial x^2\partial y}\frac{\partial g}{\partial y} + \frac{\partial^3 g}{\partial x\partial y^2}\frac{\partial g}{\partial x}\right)\right\}(x,y)\\ &\quad + o\left(\frac{k_n^2}{n^2}\right).\end{aligned}$$

For the variance we make use of the formula

$$\operatorname{Var}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big] = \mathrm{E}\Big[\operatorname{Var}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big|\Delta_{k_n}\Big]\Big] + \operatorname{Var}\Big[\mathrm{E}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big|\Delta_{k_n}\Big]\Big].$$

Now, $\operatorname{Var}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big|\Delta_{k_n}\Big]$ is exactly (3.6), where we replace b_n by Δ_{k_n}. Hence, we can write the first summand in the above formula as

$$\begin{aligned}\mathrm{E}\Big[\operatorname{Var}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big|\Delta_{k_n}\Big]\Big] &= \frac{5d_K^2}{2n\mathrm{E}\big[\Delta_{k_n}^2\big]}\frac{c(u,v)}{\phi\big(\Phi^{-1}(u)\big)\phi\big(\Phi^{-1}(v)\big)} + o\Big(n^{-1}\mathrm{E}\big[\Delta_k^2\big]\Big)\\ &\overset{(3.7)}{=} \frac{5d_K^2\pi}{2k_n}c^2(u,v) + o\Big(k_n^{-1}\Big).\end{aligned}$$

The second summand can be written as

$$\begin{aligned}\operatorname{Var}\Big[\mathrm{E}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big|\Delta_{k_n}\Big]\Big] &= \operatorname{Var}\Big[\operatorname{Bias}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big|\Delta_{k_n}\Big] + c(u,v)\Big]\\ &= \operatorname{Var}\Big[\operatorname{Bias}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big|\Delta_{k_n}\Big]\Big].\end{aligned}$$

Recall that the only random quantity in the expression for $\operatorname{Bias}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big|\Delta_{k_n}\Big]$ is $\Delta_{k_n}^4$. By application of formula (3.7) we can infer

$$\begin{aligned}\operatorname{Var}\Big[\Delta_{k_n}^4\Big] &= \mathrm{E}\Big[\Delta_{k_n}^8\Big] - \mathrm{E}\Big[\Delta_{k_n}^4\Big]^2\\ &= o\Big((k_n/n)^4\Big)\\ &= o\Big(k_n^{-1}\Big),\end{aligned}$$

where the last equality follows from the assumption that $k_n = O(n^{4/5})$. This gives

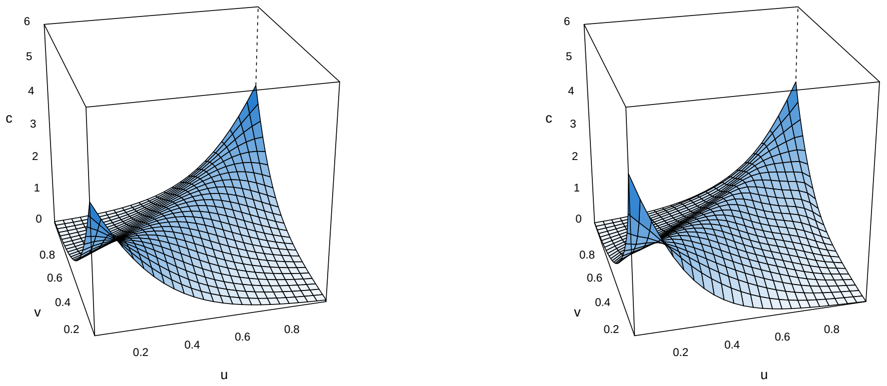
$$\operatorname{Var}\Big[\operatorname{Bias}\Big[\widehat{c}_n^{(TLL^*)}(u,v)\Big|\Delta_{k_n}\Big]\Big] = o\Big(k_n^{-1}\Big)$$

and, altogether,

$$\operatorname{Var}\Big[\widehat{c}_n^{(TLL)}(u,v)\Big] = \frac{5d_K^2\pi}{2k_n}c^2(u,v) + o\Big(k_n^{-1}\Big).$$

□

Of course, we could further expand the above expressions of $\operatorname{Bias}[\widehat{c}_n^{(TLL)}(u,v)]$ and $\operatorname{Bias}[\widehat{c}_n^{(TLL^*)}(u,v)]$, but $\partial^4 g/\partial x^4$ alone consists of more than ten different terms involving the first four partial derivatives of c as well as multiple powers of ϕ. Alas, this makes any further effort for interpretation in the general case hopeless. We should notice, though, that compared with the regular transformation estimator $\widehat{c}_n^{(T)}$, the order of the bias in Proposition 3.6 is reduced from $O(b_n^2)$ to $O(b_n^4)$. This



(a) True density

(b) Local Likelihood transformation estimator (TLL)

Figure 3.13: Perspective plots of the true density and the local likelihood transformation estimator on simulated data ($n = 1\,000$) of a Frank copula with parameter $\theta = 5$ (Kendall's $\tau \approx 0.46$). The bandwidth was selected based on the method described in Section 3.5.3.

fortunate effect is common when transitioning from standard kernel estimates to quadratic local likelihood estimation (see e.g. Loader, 1999).

An interesting special case is to take c as a Gaussian copula with association parameter ρ. In this case we have that

$$g(x, y) = \log \varphi_\rho(x, y) = -\log(2\pi) - \frac{1}{2(1-\rho^2)}\Big(x^2 + y^2 - 2\rho xy\Big),$$

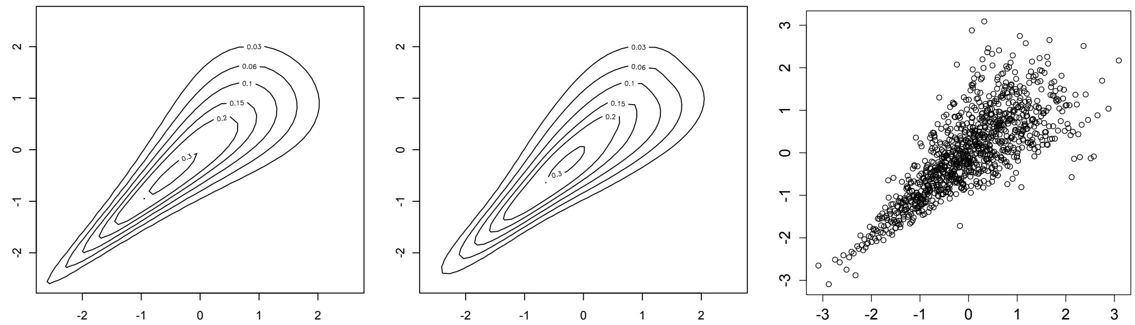
where φ_ρ is a bivariate normal distribution with zero means, unit variances and correlation ρ. We immediately see that all the derivatives needed in the above formula of the bias are zero and, hence, the asymptotic bias is zero. This is not surprising when we recall why we chose quadratic polynomials to approximate the log-density. These polynomials enable us to perfectly resemble the log-density of a Gaussian distribution. Hence, the estimator $\widehat{c}^{TLL}$ is even more flexible than local parametric (Gaussian) fitting, which, as usual, is unbiased when the model is correctly specified (for more on local parametric density estimation see Hjort and Jones, 1996).

By contrast, asymptotic analysis of the variance is more transparent. We have that

$$\mathrm{AVar}\Big[\widehat{c}_n^{(TLL^*)}(u, v)\Big] = \frac{5d_K^2}{2nb_n^2}\frac{c(u, v)}{\phi\Big(\Phi^{-1}(u)\Big)\phi\Big(\Phi^{-1}(v)\Big)},$$

which is the same as $\mathrm{AVar}\Big[\widehat{c}_n^{(T)}(u, v)\Big]$, except that it was inflated by the factor 5/2. Also this effect is common in quadratic local likelihood estimation. By contrast,

$$\mathrm{AVar}\Big[\widehat{c}_n^{(TLL)}(u, v)\Big] = \frac{5d_K^2\pi}{2k_n}c^2(u, v).$$



(a) True contours

(b) Local Likelihood transformation estimator (TLL)

(c) Transformed sample

Figure 3.14: Marginal normal contour plot of the true density and the local likelihood transformation estimate on simulated data ($n = 1\,000$) of a Clayton copula with parameter $\theta = 3$ (Kendall's $\tau = 0.6$). The bandwidth was selected based on the method described in Section 3.5.3.

Compared with $\mathrm{AVar}\Big[\widehat{c}_n^{(TLL^*)}(u, v)\Big]$ the destabilizing factors $\phi\Big(\Phi^{-1}(u)\Big)\phi\Big(\Phi^{-1}(v)\Big)$ in the denominator of the variance disappear. This does not ensure boundedness due to the influence of $c^2(u, v)$ and the variance will be unbounded wherever $c(u, v)$ is. At a first glance, we don't even know if the variance is reduced near the boundaries after all, so let us investigate the effect in more detail. By comparing the two asymptotic variances, we can conclude that, asymptotically, imposing a nearest-neighbor bandwidth has the effect of replacing $1/b_n^2$ by a factor

$$m(u, v) = \frac{\pi n}{k_n} c(u, v)\phi\Big(\Phi^{-1}(u)\Big)\phi\Big(\Phi^{-1}(v)\Big).$$

Note that with the change of variables $x = \Phi^{-1}(u), y = \Phi^{-1}(u)$, we have

$$m(x, y) = \frac{\pi n}{k_n} c(\Phi(x), \Phi(y))\phi(x)\phi(y) = \frac{\pi n}{k_n} f(x, y),$$

where $f(x, y)$ is a density with standard normal margins. It follows that $m(u, v) \to 0$ when (u, v) approaches the boundaries of the unit square. This argument shows that we actually decrease the variance near the boundaries.

In Figures 3.13 and 3.14 two things are striking. First, the estimates seem to be very accurate and even the sharp tail of the Clayton copula (Figure 3.14) is almost perfectly resembled in the estimated contours. Second, the estimates are very smooth and pleasant to look at. This can be explained by the reduced order of the bias, which allows us to make the estimates more smooth compared with the other estimators. Clearly, we could have made the estimates in the previous sections equally smooth by increasing the bandwidth, but this would have come at the cost of 'smoothing away' important features of the density. For example, increasing the bandwidth would make the estimated contours in the lower tail of the Clayton density less sharp.

Moreover, we do not see the effect of an oversmoothed lower and undermoothed upper tail of the Clayton density. This stands in contrast to the other estimation methods and is due to the adaptive bandwidth. To see how it works, we also included a scatterplot of the transformed sample in Figure 3.14c. In the lower left part, there are many observations in a small neighborhood of a point and, therefore, the distance to the k_nth closest observation is very small. This leads to a small bandwidth which allows the estimator to appropriately resemble the spiky shape. In the upper right region on the other hand, the distance to the k_nth closest observation and, as a result, the bandwidth become much larger. This leads to very smooth contours, which all the previous estimators failed to provide. Overall, the local likelihood transformation estimator seems to give the best results for the two exemplary models.

3.5.3 Bandwidth selection

A bandwidth selection rule for the local likelihood transformation estimator calls for a bit of creativity. In general, optimization of the AMISE is impracticable due to the very complex asymptotic representation of bias. We could simplify matters by considering the Gaussian as a reference copula, but also this is inadequate. The reason can be found in Section 3.5.2 where we saw that in this scenario the asymptotic bias will be zero. Therefore, one of the opposing forces in the AMISE is missing and minimization of the remaining variance-term would always suggest to use an infinitely large bandwidth. This is equivalent to a full maximum likelihood fit and we lose the local character of the estimator. A practicable alternative was suggested in Geenens et al. (2014) who use univariate *least-squares cross-validation* on the transformed domain. We will use an approach very similar to theirs and give a heuristic motivation in the following.

Consider the transformed samples $(X_i, Y_i)_{i=1,\ldots,n} = (\Phi^{-1}(U_i), \Phi^{-1}(UV_i))_{i=1,\ldots,n}$. In the analysis of the regular transformation estimator, we already noticed that using a bandwidth proportional to the correlation matrix of the transformed sample, Γ, is advisable. Recall that the distribution of the samples (X_i, Y_i) is approximately bivariate normal with covariance matrix $\Sigma = \Gamma$. For the moment, let us assume that this holds exactly. By another transformation of the data, namely $(Q_i, R_i) = \Gamma^{-1}(X_i, Y_i)$, we obtain samples from two *iid* standard normal random variables. This situation is very convenient and would allow us to optimize the parameter k_n in a univariate setting and then rescale the result to the optimal order in the bivariate case. It should be stressed that this argument is only heuristic and, theoretically, we are walking on thin ice. In reality the distribution of the transformed samples (X_i, Y_i) deviates from being bivariate normal precisely through the copula c. But this is what we wanted to estimate in the first place. Nevertheless, it provides a convenient practical approach to select the bandwidth.

We still have to provide a method to select the parameter k_n in a univariate setting. A conceptually easy and popular method is the *least-squares cross-validation* bandwidth selector and will be explained in the following. Assume $\widehat{f}_n$ is an estimator for the density f_Q of a random variable Q supported on the real line. We can write the mean integrated squared error of this estimator as

$$\begin{aligned}\mathrm{MISE}\big[\widehat{f}_n\big] &= \int_{\mathbb{R}} \mathrm{E}\bigg[\Big(\widehat{f}_n(x) - f_Q(x)\Big)^2\bigg]dx \\ &= \mathrm{E}\bigg[\int_{\mathbb{R}} \Big(\widehat{f}_n(x) - f_Q(x)\Big)^2 dx\bigg] \\ &= \mathrm{E}\bigg[\int_{\mathbb{R}} \widehat{f}_n^2(x)dx\bigg] - 2\mathrm{E}\bigg[\int_{\mathbb{R}} \widehat{f}_n(x)f_Q(x)dx\bigg] + \int_R f_Q^2(x)dx \\ &= \mathrm{E}\bigg[\int_{\mathbb{R}} \widehat{f}_n^2(x)dx\bigg] - 2\mathrm{E}\big[\widehat{f}_n(Q)\big] + \int_R f_Q^2(x)dx.\end{aligned}$$

The goal is now to estimate the terms in the last line. Since the third term does not depend on the estimate f_n it can be ignored when optimizing the MISE. An obvious (and unbiased) estimator for the first term is $\int_{\mathbb{R}} \widehat{f}_n^2(x)dx$. The second term can be estimated via *leave-one-out cross-validation* (e.g. Rizzo, 2008) by

$$-\frac{2}{n}\sum_{i=1}^{n} \widehat{f}_{-i}(X_i),$$

where $\widehat{f}_{-i}$ is the estimator $\widehat{f}_{n-1}$ applied to the data set $(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$. Altogether minimizing the MISE is therefore approximately equivalent to choosing

$$k_{Q,n}^{LSCV} = \arg\min_{k_n \in \mathbb{N}} \bigg\{\int_{\mathbb{R}} \widehat{f}_n^2(x)dx - \frac{2}{n}\sum_{i=1}^{n} \widehat{f}_{-i}(X_i)\bigg\}.$$

The result will be a good estimate for the optimal bandwidth in the univariate case. The optimal orders of k_n are different for one- and two-dimensional kernel estimation, though. They can be derived from optimizing AMISE expressions of the corresponding estimators. In the case of quadratic local likelihood density estimation, the difference in orders is a factor $n^{-4/45}$ and we should therefore multiply k_n^{LSCV} with $n^{-4/45}$ to take account of that fact (see Geenens et al., 2014).

In practice, the distributions of the samples Q_i and R_i will not be exactly equal. It turned out as a good solution to apply least-squares cross-validation on both samples and take the minimum of the two parameters as the final choice. Let us summarize the whole procedure:

1. Choose B_n as the empirical correlation matrix of $(X_i, Y_i)_{i=1,\dots,n}$.

2. Construct samples $(Q_i, R_i)_{i=1,\dots,n} = \Big(B_n^{-1}(X_i, Y_i)\Big)_{i=1,\dots,n}$.

3. Find $k_{Q_n}^{LSCV}$ and $k_{R,n}^{LSCV}$ by univariate least-squares cross-validation on the samples $(Q_i)_{i=1,\dots,n}$ and $(R_i)_{i=1,\dots,n}$ respectively.

4. Choose $k_n = \Big\lfloor \min\Big\{k_{Q_n}^{LSCV}, k_{R,n}^{LSCV}\Big\} n^{-4/45}\Big\rfloor$.

Since this approach is only heuristically motivated, it is important to validate its results. In Figures 3.13 and 3.14 we saw that both estimates are very smooth on the whole unit square. If anything, we could use a smaller value for k_n which would result in a less smooth estimate and a decreased bias. On the other hand, the estimates were also almost perfectly resembling the shape of the true density and we did not see important features being 'smoothed away'. All in all, the bandwidth selection rule seems to work appropriately.

3.6 Simulation study

In the following we will discuss the performance of the presented estimators on the basis of an extensive simulation study and compare the presented estimators with the popular parametric approach.

3.6.1 Methods

For conducting the study, all estimators were implemented in R (R Core Team, 2013). Let us mention the specific settings for each estimation method. In parentheses we give short handles with which each estimator can be identified in plots and tables.

- **Mirror reflection estimators (MR/MRS)** Both the basic (MR) and improved version of this estimator (MRS) are considered. We used an Epanechnikov kernel and the bandwidths were chosen according to the rule-of-thumb discussed in Section 3.2.4. It optimizes the AMISE for a reference copula density, for which we used the Frank family. The parameter of the reference density was set by inversion of the empirical Kendall's τ. The shrinkage intensity of the MRS estimator was set to $\alpha = 1/2$.

- **Beta kernels (beta)** We used an Epanechnikov kernel and the bandwidths were chosen according to the rule-of-thumb discussed in Section 3.3.3. Again, we choose the Frank copula as reference density and its parameter was set by inversion of the empirical Kendall's τ.

- **Transformation estimators (T/TB)** The one-parameter (T) and fully parameterized (TB) versions are considered. We used an Epanechnikov kernel and the bandwidths were chosen according to the rule-of-thumbs discussed in Section 3.4.4. They optimize the AMISE on the transformed level w.r.t to a bivariate normal distribution. The empirical correlation matrix of the transformed sample was used as an estimate for Γ.

- **Local-likelihood transformation estimator (TLL)** The implementation employs the `locfit` R-package (Loader, 2013) for local likelihood estimation. For this estimator, we used a Gaussian kernel due to numerical difficulties with the Epanechnikov kernel. This should be a disadvantage, if there is any notable difference at all. The bandwidth matrix B and nearest-neighbor parameter k_n

were selected according to the cross-validation procedure described in Section 3.5.3.

- **Parametric estimators (par/par-)** Parametric estimation was implemented via the `BiCopSelect` function of the `VineCopula` package (Schepsmeier et al., 2013). The function fits several one- and two- parameter copula families including the Gaussian, t- and Tawn copulas as well as the most popular Archimedean family such as Clayton, Joe, Gumbel, BB1, etc. and rotations thereof (see package description for a full list). The parameter is estimated by maximum likelihood and the model with the highest AIC is selected (par). Since we simulated data from parametric models, but in general there is no reason to assume that a known parametric model is underlying real data, we also included a reduced version, where the estimation procedure is prohibited from choosing a family that contains the true model (par-).

To get a broad view over the performance in different situation, we considered various scenarios:

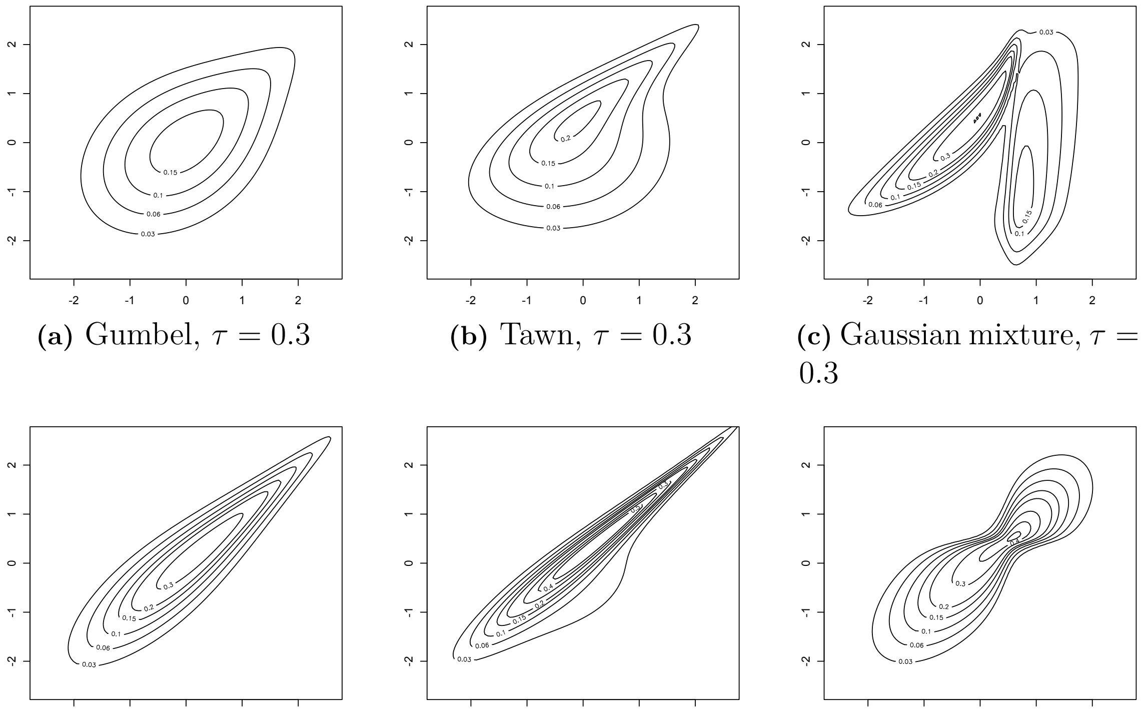
- **Sample size** We simulated data for two different sample sizes, $n = 250$ and $n = 1000$.

- **Copula families** We considered three different copula families to simulate the data: the Gumbel copula, Tawn copula and a Gaussian mixture copula (c.f. Sections 2.1.1 and 2.1.3).

- **Dependence** For each family, there is one scenario with weak and one with strong dependence. For each family this corresponds (approximately) to a Kendall's τ of 0.3 and 0.7 respectively. In the following table, we give the chosen parameters for each scenario.

τ	Gumbel	Tawn	Gaussian mixture
0.3	1.43	$(2.7, 0.4, 1)$	$\boldsymbol{\mu} = \left(\begin{smallmatrix} 2 \\ 2 \end{smallmatrix}\right), \mathring{\boldsymbol{\mu}} = \left(\begin{smallmatrix} 6 \\ 1.6 \end{smallmatrix}\right), \Sigma = \left(\begin{smallmatrix} 1 & 0.9 \\ 0.9 & 1 \end{smallmatrix}\right), \mathring{\Sigma} = \left(\begin{smallmatrix} 4 & 2 \\ 2 & 4 \end{smallmatrix}\right), \alpha = 0.65$
0.7	3.33	$(6.5, 0.8, 1)$	$\boldsymbol{\mu} = \left(\begin{smallmatrix} 2 \\ 2 \end{smallmatrix}\right), \mathring{\boldsymbol{\mu}} = \left(\begin{smallmatrix} 6 \\ 6 \end{smallmatrix}\right), \Sigma = \left(\begin{smallmatrix} 1.35 & 0.8 \\ 0.8 & 1 \end{smallmatrix}\right), \mathring{\Sigma} = \left(\begin{smallmatrix} 2.1 & 1.4 \\ 1.4 & 2.1 \end{smallmatrix}\right), \alpha = 0.65$

In total we have six simulation models which we compare on two sample sizes. Marginal normal contour plots of the true density underlying each simulation model are given in Figure 3.15. The two Gumbel models are the 'most regular' cases, since the copula density is symmetric in its components. They also have a spike in the upper right part of the contours, which may constitute a difficulty for the kernel estimators especially for $\tau = 0.7$. The Tawn copula models give an additional asymmetry w.r.t. to the copula's arguments. Lastly, the two Gaussian mixture copula models (c.f. Examples 2.5 and 2.11) are seemingly 'odd' cases with two modes of different height. This will be impossible to capture for the popular parametric models, but is also difficult for the kernel estimators, since the optimal amount of smoothing varies substantially in different areas of the density. For each scenario



(a) Gumbel, $\tau = 0.3$

(b) Tawn, $\tau = 0.3$

(c) Gaussian mixture, $\tau = 0.3$

(d) Gumbel, $\tau = 0.7$

(e) Tawn, $\tau = 0.7$

(f) Gaussian mixture, $\tau = 0.7$

Figure 3.15: Marginal normal contour plots of the true densities underlying the considered simulation models.

and sample size, $N = 500$ samples were simulated and estimates were obtained for each estimation method. The performance of the estimators on a particular sample is then measured by three performance measures: the integrated squared error (ISE), integrated absolute error (IAE) and the Hellinger distance (HD). They all describe a distance in the space of continuous functions and are defined as

$$\begin{aligned}
\mathrm{ISE}\Big[\widehat{c}_n^{(\cdot)}\Big] &= \int_0^1 \int_0^1 \Big(\widehat{c}_n^{(\cdot)}(u,v) - c(u,v)\Big)^2 dudv \\
\mathrm{IAE}\Big[\widehat{c}_n^{(\cdot)}\Big] &= \int_0^1 \int_0^1 \Big|\widehat{c}_n^{(\cdot)}(u,v) - c(u,v)\Big| dudv \\
\mathrm{HD}\Big[\widehat{c}_n^{(\cdot)}\Big] &= \sqrt{\frac{1}{2}\int_0^1 \int_0^1 \bigg(\sqrt{\widehat{c}_n^{(\cdot)}(u,v)} - \sqrt{c(u,v)}\bigg)^2 dudv}.
\end{aligned}$$

	τ	n	MR	MRS	beta	T	TB	TLL	par-	par
Gumbel	0.3	250	0.12	0.06	0.09	0.06	0.06	**0.04**	0.03	0.02
		1 000	0.10	0.03	0.06	0.03	0.03	**0.02**	0.01	0.00
	0.7	250	0.98	0.53	0.74	1.03	0.25	**0.20**	0.23	0.07
		1 000	0.82	0.35	0.54	0.74	0.11	**0.10**	0.13	0.01
Tawn	0.3	250	0.32	0.24	0.24	0.20	0.15	**0.12**	0.30	0.02
		1 000	0.28	0.19	0.18	0.12	0.08	**0.07**	0.26	0.04
	0.7	250	2.25	1.71	1.90	2.79	1.12	**0.56**	1.91	0.12
		1 000	1.95	1.30	1.49	2.32	0.69	**0.26**	1.71	0.04
Gaussian	0.3	250	1.03	0.96	0.76	0.88	0.75	**0.58**	1.13	1.15
mixture		000	0.86	0.81	0.60	0.67	0.56	**0.41**	1.10	1.11
	0.7	250	**0.24**	0.28	0.25	0.66	0.45	0.26	0.29	0.30
		1 000	**0.17**	0.21	**0.17**	0.46	0.28	0.19	0.26	0.26

Table 3.1: Estimated MISE over $N = 500$ simulations (rounded on two decimals). The best kernel estimator in each scenario is highlighted in bold.

For numerical convenience, all measures were estimated on a grid of 100 × 100 points. More specifically, we used

$$\widehat{\mathrm{ISE}}\big[\widehat{c}_n^{(\cdot)}\big] = \frac{1}{100^2}\sum_{i=1}^{100}\sum_{j=1}^{100}\left(\widehat{c}_n^{(\cdot)}\left(\frac{i}{101},\frac{j}{101}\right) - c\left(\frac{i}{101},\frac{j}{101}\right)\right)^2 dudv$$

$$\widehat{\mathrm{IAE}}\big[\widehat{c}_n^{(\cdot)}\big] = \frac{1}{100^2}\sum_{i=1}^{100}\sum_{j=1}^{100}\left|\widehat{c}_n^{(\cdot)}\left(\frac{i}{101},\frac{j}{101}\right) - c\left(\frac{i}{101},\frac{j}{101}\right)\right| dudv$$

$$\widehat{\mathrm{HD}}\big[\widehat{c}_n^{(\cdot)}\big] = \sqrt{\frac{1}{2}\frac{1}{100^2}\sum_{i=1}^{100}\sum_{j=1}^{100}\left(\sqrt{\widehat{c}_n^{(\cdot)}\left(\frac{i}{101},\frac{j}{101}\right)} - \sqrt{c\left(\frac{i}{101},\frac{j}{101}\right)}\right)^2}.$$

3.6.2 Analysis

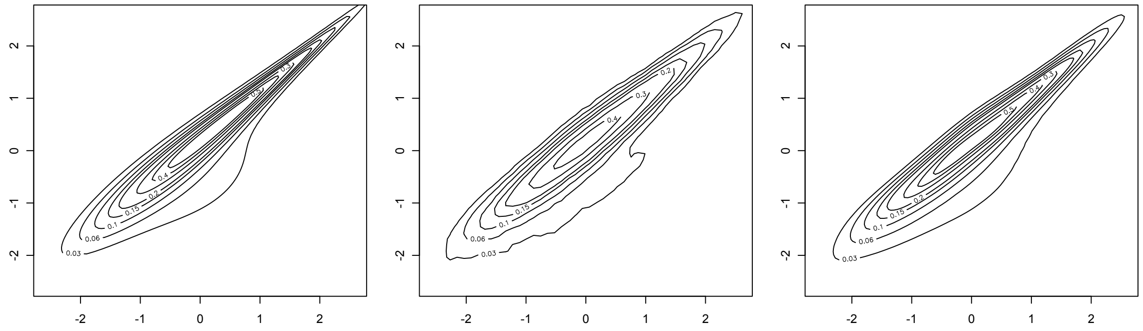
It was found that the results essentially lead to the same conclusions across all three performance measures. For illustrational purposes in this section, we will therefore only use the integrated squared error (ISE). The interested reader can find boxplots of the full results accompanied by marginal normal contour plots of exemplary estimates for each method in Appendix A.

In Table 3.1 the mean of the ISE for a each model and estimator is given and the best kernel estimator is highlighted in bold. We will focus on the Gumbel and Tawn models first. Clearly, the TLL estimator is the best of all kernel estimators in all situations. Mostly, the improvement over the other estimation methods is substantial. It is also apparent, that the kernel estimators have more trouble in case of strong dependence. That is because of the more rapidly exploding tails in the upper right corner in both models. As we have seen in the asymptotic analysis of the estimators, the magnitude of the MISE is related to the magnitude of the copula density and its partial derivatives. When the tails explode more rapidly, these terms get bigger.

The TB and TLL estimators perform much better than the competitors in these situations. The performance of the two estimators is quite comparable in the first six rows of Table 3.1, the margin becomes larger for the Tawn model with strong dependence. Here, the TLL estimator is profiting of the adaptive bandwidth. This is illustrated in Figure 3.16. In the low density regions in the lower right part of the contours, the TB estimates get too wiggly, whereas the TLL estimate is very smooth due to the increased bandwidth. In addition, the high density region in the center of the contours are underestimated by the TB estimator, whereas the decreased bandwidth of the TLL estimator allows for higher peaks.

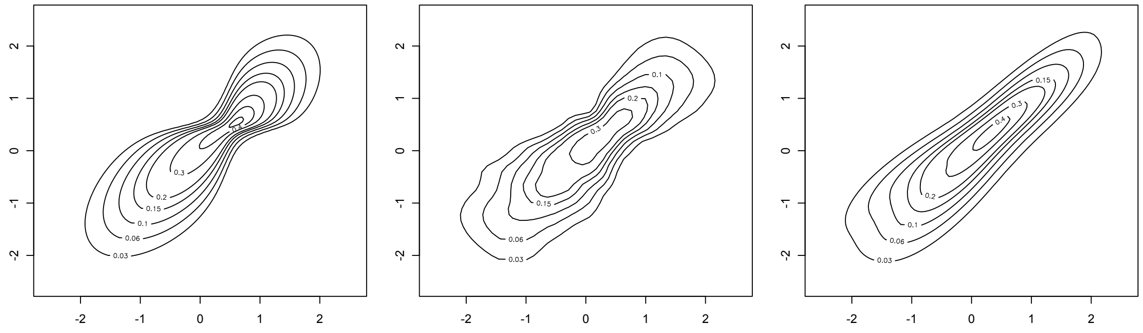
Not surprisingly, the full parametric estimator (par) outperforms all competitors for the Gumbel and Tawn models. In these cases, the true copula model is likely to be selected by the estimation procedure. Then, the estimator is consistent and provides a better convergence rate than non-parametric estimators (see e.g. Wand and Jones, 1994). Note that this is not the case for the reduced estimator (par-), since here the parametric model will always be misspecified. In this case, the estimator is not even consistent and is outperformed by the best nonparametric estimator in all but the Gumbel model with weak dependence. This finding is an excellent illustration of how important it is that a parametric model is correctly specified. Most of the time, parametric assumptions for copulas are made purely for practical convenience, as it allows to reduce an essentially infinite-dimensional problem to just a few parameters. Clearly, there are huge benefits such as simple implementation and further use of the estimate. Still, one should be very careful when making parametric assumptions and it is advisable to at least cross-check against a non-parametric estimate.

For the Gaussian mixture models, the picture is a little different. First of all, the range of parametric families in our implementation does not include Gaussian mixture copulas. In this case, both parametric estimators are always misspecified and almost equivalent. The estimators MR, MRS, beta and TLL outperform the parametric versions in all cases. In case of weak dependence the TLL estimator gives again the best results, whereas the basic mirror-reflection estimator performs best in case of strong dependence. This comes a little surprising judging by the poor performance for the Gumbel and Tawn models. Also the close second's (beta) performance was just average before. There are two features of the density to be held accountable for this phenomenon. One is that the tails of the Gaussian mixture densities are bounded (c.f. Example 2.11). In such cases the beta and MR estimators seem to be reputable competitors. The second issue is the multi-modality of the densities. In such cases Wand and Jones (1993) showed that using the full bandwidth parameterization is likely to decrease the performance of kernel estimators. This fact is underlined by the marginal normal contour plots of estimates on an exemplary sample in Figure 3.17. Recall that the full bandwidth parameterization of the TLL estimator leads to elliptical kernels resembling the covariance structure of the data. As a result, the kernels will stretch over the gap between the two modes and the bimodal feature gets 'smoothed away'. In contrast, the MR and beta estimators (beta not shown here) are able to capture this feature. This explains why these two estimators can compete with the TLL in this scenario while being substantially worse in others. However, even in case of strong dependence the edge over the TLL estimator is very thin. In



(a) true **(b)** TB **(c)** TLL

Figure 3.16: Tawn copula model, $\tau = 0.7$. Contour plots of the true density as well as TB and TLL estimates on a samples of size $n = 1\,000$.



(a) true **(b)** MR **(c)** TLL

Figure 3.17: Gaussian mixture copula model, $\tau = 0.7$. Contour plots of the true density as well as MR and TLL estimates on a samples of size $n = 1\,000$.

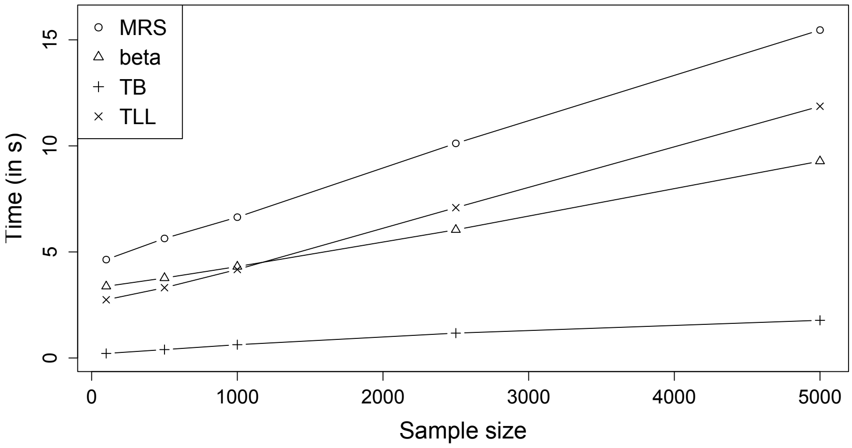


Figure 3.18: Mean computation time of selected estimators for varying sample size.

case of weak dependence the kernel shapes are not drastically different and, hence, the harm caused by the full bandwidth parameterization is not as detrimental.

Sample size did hardly play any role in the ranking of the estimators, so we can expect our findings to be valid over a wide range of sample sizes. Overall, the TLL estimator seems to be the best choice amongst all kernel estimators. It may also improve notably over parametric estimators, when the data is not generated by a known parametric model.

A note on computation time

The whole simulation study was carried out on a computing cluster, but this setup and the computations conducted in the study might not be of much interest in practice. To get a more realistic view of the computation time in practical applications, we compare the aggregate time needed for bandwidth selection and obtaining estimates on a grid of 30×30 points (which was the typical grid for producing the plots in this thesis). The computations are conducted on a customary Windows 7 Lenovo laptop with Intel Core Duo i3-3120 CPU @ 2.50 GHz and 8 GB RAM.

The mean computation times over 100 simulated samples from a Gaussian copula ($\tau = 0.5$) of different size can be found in Figure 3.18. We did not include the MR and T estimators as the computation time is almost identical to the MRS and TB estimators respectively.

The time needed for the bandwidth selection can be observed by the vertical shift across the different estimators for very small sample size. The steepness of the lines correspond to the marginal cost of evaluating the estimator when sample size is increased. First of all, we see that the computation times of the TB estimator are significantly smaller compared with the others. This is facilitated by the much simpler bandwidth selection rule that requires neither numerical integration nor cross-validation. The other estimators are quite comparable whereas the marginal cost of increasing sample size is highest for the MRS and TLL estimators. The reasons are that the size of the augmented data is nine times as large as the actual sample size for the MR estimators, and that we used the computationally more demanding Gaussian kernel for the TLL estimator. An interesting observation is that for all estimators computation time grows approximately linear in sample size. This is a fortunate feature if one is interested in estimation on very large data sets. Even for sample sizes as big as 5 000, all estimators provide results in a matter of seconds. Judging from computation time, all estimators certainly qualify for application in practice.

Chapter 4

Kernel estimation of h-functions

There is a second ingredient we need to estimate a full vine copula model: the h-function. In parametric models, the h-function is readily obtained by using the copula family and parameters of a density estimate. In a nonparametric setting the issue calls for further considerations.

The h-function is defined as the conditional probability $h(u|v) = \mathrm{P}(U \leq u|V = v)$ and can be expressed as

$$h(u|v) = \frac{\partial}{\partial v} C(u, v) = \int_0^u c(s, v) ds, \tag{4.1}$$

provided $c(u, v)$ is a copula density. An obvious estimator $\widehat{h}_n(u|v)$ could be obtained by plugging a density estimate $\widehat{c}_n^{(\cdot)}(u, v)$ into (4.1). This would result in the estimator

$$\widehat{h}_n^{(\cdot)}(u|v) = \int_0^u \widehat{c}_n^{(\cdot)}(s, v) ds.$$

Whenever $\widehat{c}_n^{(\cdot)}(u, v)$ is a consistent estimator of $c(u, v)$, this estimator will be consistent for $h(u|v)$. However, it is not a proper distribution function in general. This is due to the fact that the estimates $\widehat{c}_n^{(\cdot)}(u, v)$ are usually not a proper copula density and formula (4.1) does not hold exactly. This is a serious problem since we cannot assure that the pseudo samples $\widehat{h}_n(U_i|V_i)$ needed for estimation in higher trees of a vine copula, are contained in $[0, 1]$ for all $i = 1, \dots, n$. Consequently, they cannot be considered as samples from a copula and estimation on higher trees may fail.

To overcome this problem we can simply rescale the integral by the estimate of $\widehat{h}_n^{(\cdot)}(1|v)$ as in

$$\widehat{h}_n^{(\cdot),scale}(u|v) = \frac{\int_0^u \widehat{c}_n^{(\cdot)}(s, v) ds}{\int_0^1 \widehat{c}_n^{(\cdot)}(s, v) ds},$$

By consistency of $\widehat{c}_n^{(\cdot)}(u, v)$, the denominator converges to one in probability and the above estimator is consistent for $h(u|v)$. Furthermore, the fact that $\widehat{c}_n^{(\cdot)}(u, v) \geq 0$ implies that

$$0 \leq \int_0^u \widehat{c}_n^{(\cdot)}(s, v) ds \leq \int_0^1 \widehat{c}_n^{(\cdot)}(s, v) ds, \tag{4.2}$$

which is equivalent to $0 \leq \widehat{h}_n^{(\cdot)}(U_i|V_i) \leq 1$.

This estimator has two appealing properties. First of all, there is no bandwidth parameter we need to select. This keeps the estimation process simple and makes implementation very easy. Secondly, the use of the preceding density estimates preserves some level of consistency in view of the estimation process of a whole vine copula. It favors the possibility to compensate errors on lower levels of the tree by corrected estimates in higher trees.

But there is also one severe practical drawback. If the density estimator does not allow for an analytic expression of the integral $\int_0^u \widehat{c}_n^{(\cdot)}(s, v)ds$, we have to perform a substantial number of numerical integrations for the estimation and evaluation of a vine copula. In addition, the precision of the numerical integrations has to be very high to ensure that (4.2) holds in practice. This becomes time consuming very quickly and renders the integration approach almost infeasible. Unfortunately, analytic expressions are only available for the basic mirror-reflection and transformation estimators. In general, it is not a practicable approach and we have to look for alternatives.

In the remainder of this chapter, we will introduce a numerically attractive alternative, discuss its properties and give advice for bandwidth selection.

4.1 A kernel regression estimator

There is no necessity to use the a density estimates in order to estimate an h-function. We could also use an individual estimation approach. One promising way is to relate the estimation problem to a regression equation. First, let us fix a value $u \in [0, 1]$ and note that we can write

$$h(u|v) = \mathrm{P}(U_i \leq u|V_i = v) = \mathrm{E}\Big[\mathbb{1}(U_i \leq u)\Big|V_i = v\Big].$$

By the properties of conditional expectation we get

$$\mathbb{1}\Big(U \leq u\Big|V = v\Big) = \mathrm{E}\Big[\mathbb{1}(U \leq u)\Big|V = v] + \epsilon,$$

where ϵ has zero mean and variance $\gamma^2(u, v)$ conditional on V. The variance of ϵ has two arguments u and v to emphasize that it is different for each $(u, v) \in [0, 1]^2$. More specifically, we have $\gamma^2(u, v) = h(u|v)(1 - h(u|v))$, since $\mathbb{1}\Big(U \leq u\Big|V = v\Big)$ is a Bernoulli random variable with success probability $\mathrm{P}(U \leq u|V = v) = h(u|v)$. The equation

$$\mathbb{1}\Big(U \leq u\Big|V = v\Big) = h(u|v) + \epsilon_u,$$

can thus be interpreted as a classic non-parametric regression with predictor variable V and response $\mathbb{1}(U \leq u|V)$. In this context, $h(u|v)$ serves as the mean regression function and is to be estimated. As a consequence, we can rely on kernel regression techniques to estimate the h-function.

Just as for density estimation, we have to take care that the estimator deals well with the bounded support of the random variable V. One estimator that is known to do particularly well on bounded support is the local-linear estimator of Fan and Gijbels (1992). Let us sketch the basic idea. Assume that $h(u|v)$ is differentiable in v, and u is fixed. Taylor's theorem states that we can locally approximate the regression function $h(u|v)$ by a straight line, i.e.

$$h(u|v') \approx a_1(u,v) + a_2(u,v)\,(v' - v),$$

for all v' in a neighborhood of v. Provided we are given *iid* copies $(U_i, V_i)_{i=1,\ldots,n}$ of the random vector (U, V), the local coefficients can be estimated by minimizing the weighted least squares problem

$$\operatorname*{arg\,min}_{a_1(u,v),a_2(u,v)\in\mathbb{R}} \sum_{i=1}^{n} \Big(\mathbb{1}\Big(U_i \leq u \Big| V_i\Big) - a_1(u,v) - a_2(u,v)\,(v - V_i) \Big)^2 K_{b_n}(v - V_i),$$

where $K_{b_n}(\cdot) = K(\cdot/b_n)/b_n$, K is a symmetric and bounded probability density, and b_n is a sequence of bandwidths tending to zero as $n \to \infty$. An estimate of $h(u|v)$ is then given by the coefficient $\widehat{a}_1(u,v)$ obtained as solution of the above problem. It can be given in closed form (c.f. Fan and Gijbels, 1992) as

$$\widehat{a}_1(u,v) = \sum_{i=1}^{n} \frac{w_i(v)\mathbb{1}(U_i \leq u)}{\sum_{i=1}^{n} w_i(v)},$$

where

$$w_i(v) := K_{b_n}(v - V_i)\Big(s_{n,2} - (v - V_i)s_{n,1}\Big), \qquad i = 1, \ldots, n,$$

with $s_{n,j} := \sum_i^n K_{b_n}(v - V_i)\Big(v - V_i\Big)^j$, for $j = 1, 2$.

There is, however, one further issue we have to take care of. When some of the weights w_i are negative, there is no guarantee that the resulting estimates of $h(u|v)$ are contained in $[0,1]$, nor can we be sure that they are increasing in u. This means that $h(u|v)$ is not a proper distribution function. Hall et al. (1999) proposed a method to guarantee positive weights $w_i(v)$ by modifying the weighting in the least squares problem. Let $p_i(v) \geq 0$, $\sum_{i=1}^n p_i(v) = 1$ and more specifically

$$p_i(v) = \frac{1}{n} \frac{1}{1 + \lambda(v)(v - V_i)K_{b_n}(V_i - v)}, \qquad i = 1, \ldots, n, \tag{4.3}$$

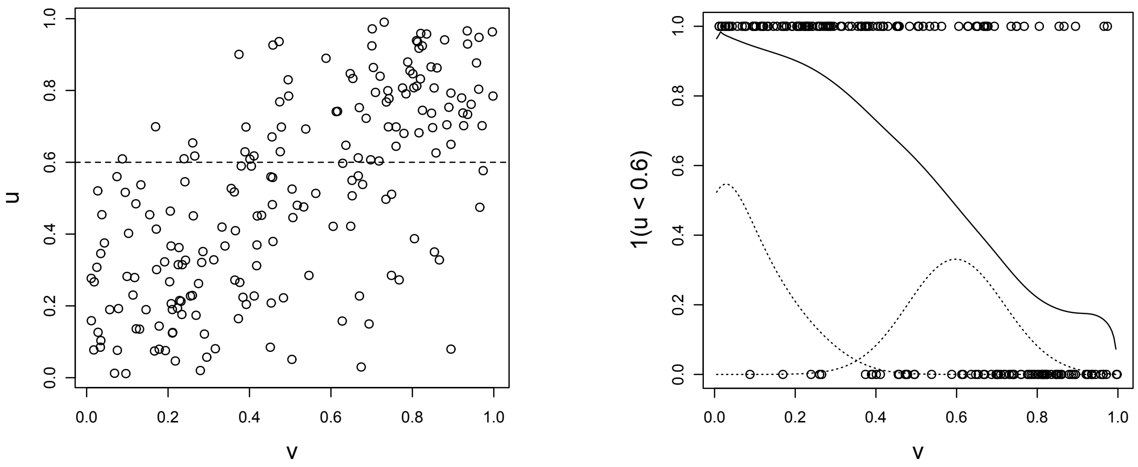
where $\lambda(v)$ solves the equation[1]

$$\sum_{i=1}^{n} p_i(v)(V_i - v)K_{b_n}(V_i - v) = 0, \tag{4.4}$$

We can then obtain an estimate of $h(u|v)$ by solving the modified weighted least squares problem

$$\operatorname*{arg\,min}_{a_1(u,v),a_2(u,v)\in\mathbb{R}} \sum_{i=1}^{n} \Big(\mathbb{1}\Big(U_i \leq u \Big| V_i\Big) - a_1(u,v) - a_2(u,v)\,(v - V_i) \Big)^2 p_i(v)K_{b_n}(v - V_i).$$

[1] In practice the $\lambda(v)$ will have to be computed numerically, but this can be achieved comparably fast with just a few steps of a Newton-Raphson algorithm.



(a) (b)

Figure 4.1: The local linear estimator. (a) Scatterplot of samples from a Frank copula. We fix a value $u = 0.6$ (dashed line). (b) Regression plot of $\mathbb{1}\Big(U_i \leq 0.6\Big|V_i\Big)$ against V_i. Samples are indicated as circles, the estimated regression function as solid line and the weights $w_i(v)$ for $v = 0.1, 0.6$ as dotted lines.

The solution of this modified problem is

$$\widehat{a}_1(u, v) = \sum_{i=1}^{n} \frac{w_i(v)\mathbb{1}(U_i \leq u)}{\sum_{i=1}^{n} w_i(v)},$$

where

$$w_i(v) := p_i(v)K_{b_n}(v - V_i)\Big(s_{n,2} - (v - V_i)s_{n,1}\Big), \qquad i = 1, \dots, n,$$

with $s_{n,j} := \sum_i^n p_i(v)K_{b_n}(v-V_i)\Big(v-V_i\Big)^j$, for $j = 1, 2$. By (4.4), we have that $s_{n,1} = 0$ and the remaining terms in $w_i(v)$ are all non-negative by definition. So overall, $w_i(v)$ will always be non-negative. Let us subsume this in a definition.

Definition 4.1. *For all u, v in $[0, 1]^2$, the **local linear estimator** of an h-function $h(u|v)$ with bandwidth parameter $b_n > 0$ is defined as*

$$\widehat{h}_n^{(LL)}(u|v) = \sum_{i=1}^{n} \frac{w_i(v)\mathbb{1}(U_i \leq u)}{\sum_{i=1}^{n} w_i(v)},$$

where

$$w_i(v) := p_i(v)K_{b_n}(v - V_i)\sum_{j=1}^{n} p_j(v)(v - V_j)^2 K_{b_n}(v - V_j), \qquad i = 1, \dots, n,$$

and the $p_i(v)$ are defined in (4.3) and (4.4).

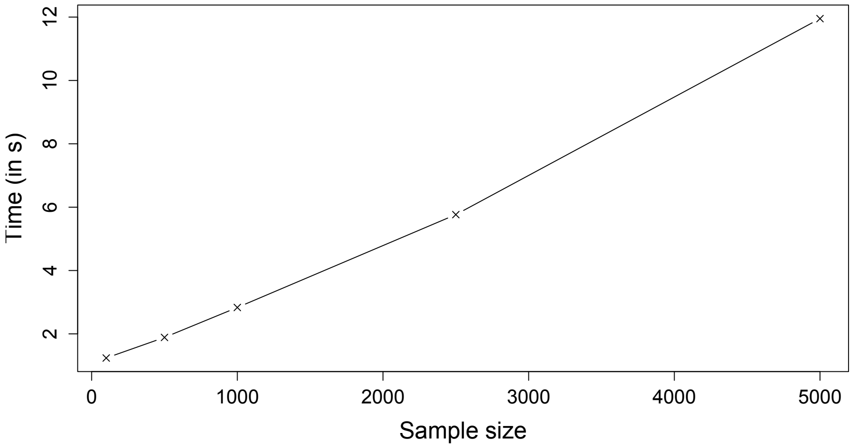


Figure 4.2: Mean computation time of the local linear estimator for varying sample size.

To get a better understanding of what is happening in the estimation process, let us take a look at Figure 4.1. In (a) we see a scatterplot of a sample from a Frank copula (note the interchanged u- and v-axes). We fix a value $u = 0.6$ which is indicated by the dashed horizontal line. In a regression plot of the response variables $\mathbb{1}\Big(U_i \leq 0.6 \Big| V_i\Big)$ against the predictor variables V_i, every sample point that is below the horizontal line in (a) will be represented by a one and every sample point that is above this line will be represented as a zero. This plot can be seen in (b). Note that the lower the value of v, the more points lie below the horizontal line resulting in a large number of ones in (b) (and the other way around). To obtain the regression function $\widehat{h}_n^{(LL)}(0.6|v)$ (solid line), the points in (b) are locally averaged according to the weights defined as $w_i(v)/\sum_{i=1}^n w_i(v)$ (see Definition 4.1). For $v = 0.1, 0.6$ these weights are shown as dotted lines (multiplied by 20 for better visibility). The local weighting scheme adapts to the location of the estimate by getting asymmetric close to the boundary. We also see that the influence of an individual observation V_i on the estimate $h(u|V_i)$ increases when approaching the boundaries.

A note on computation time

Clearly, $h(u|v)$ is not a distribution function in v but in u, but by performing several separate regressions for different values of u we can also obtain an estimate of the distribution function $h(u|v) = P(U \leq u|V = v)$ for fixed v. This might sound time-consuming at first, but actually the estimator is quite fast thanks to the (almost) closed form solution given above. There is a numerical optimization involved in finding the weights p_i (hence the 'almost'). However, as noted earlier, a Newton-Raphson algorithm usually requires just a few steps so that the computation time is negligible. In fact, the computational demands are very much comparable to the kernel density estimators presented in the last chapter. This can be seen in Figure 4.2 where we conducted the same experiment as at the end of the last chapter (see Section 3.6.1 for details on the setup). The time consumed for evaluating the estimator on a grid of 30 × 30 points is in the range of a few seconds for moderate sample sizes. Again, the computation time appears to grow linearly in the sample size making the estimator well scalable to estimation on large samples.

4.2 Properties

We start with discussing asymptotic approximations of bias and variance of the estimator.

Proposition 4.1. *Let $(u, v) \in (0, 1)^2$ such that $h(u|v) \in (0, 1)$ and let $h(u|v)$ be twice continuously differentiable w.r.t. v in a neighborhood of these points. Denote again $h_{vv}(u|v) = \partial^2 h(u|v)/\partial v^2$. For $b_n \to 0$ with $nb_n \to \infty$ as $n \to \infty$ it holds*

$$\begin{aligned}\operatorname{Bias}\Big[\widehat{h}_n^{(LL)}(u|v)\Big] &= \frac{b_n^2\sigma_K^2}{2} h_{vv}(u|v) + o\Big(b_n^2\Big),\\ \operatorname{Var}\Big[\widehat{h}_n^{(LL)}(u|v)\Big] &= \frac{d_K h(u|v)\Big(1 - h(u|v)\Big)}{nb_n} + o\bigg(\frac{1}{nb_n}\bigg),\end{aligned}$$

where

$$\sigma_K^2 = \int_0^1 s^2 K(s)ds \qquad \text{and} \qquad d_K = \int_0^1 K^2(s)ds.$$

Proof. Under the given conditions, Theorem 1 in Hall et al. (1999) gives us the first-order approximation

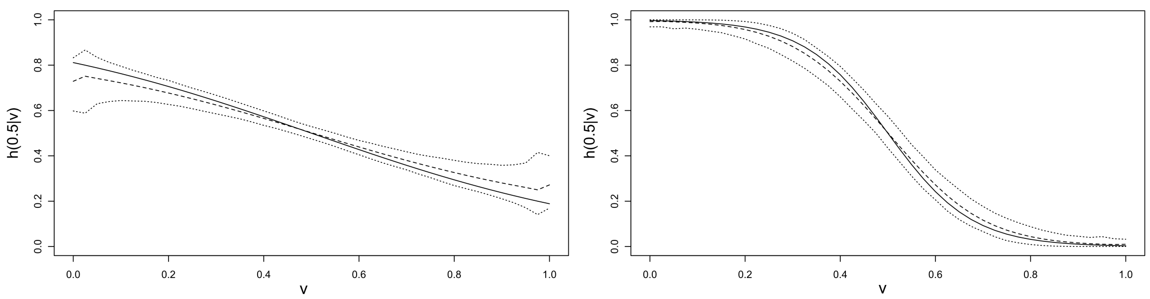
$$\widehat{h}_n^{(LL)}(u|v) - h(u|v) = \frac{b_n^2\sigma_K^2}{2} h_{vv}(u|v) + \frac{\sqrt{d_K}\sqrt{h(u|v)\Big(1 - h(u|v)\Big)}}{\sqrt{nb_n}} Z + o\bigg(b_n^2 + \frac{1}{\sqrt{nb_n}}\bigg),$$

where $Z \sim \mathcal{N}(0, 1)$ and the claim follows immediately. □

The reason why we excluded points where $h(u|v) \in \{0, 1\}$ is simply that in this case asymptotic bias and variance will be zero. It is interesting to note that the bias and variance approximations are equivalent to those of the unmodified local linear estimator of Fan and Gijbels (1992). So at least asymptotically, there is no cost of modifying the weights to ensure that $h(u|v)$ is a distribution function.

The asymptotic approximations already reveal situations in which the estimator could get into trouble. When the absolute value of $h_{vv}(u|v)$ is large, the bias will be large as well. This corresponds to points where the h-function has high curvature in v-direction (the conditional variable). Furthermore, the asymptotic variance is maximal when $h(u|v) = 1/2$ and decreases monotonically until $h(u|v)$ approaches zero or one.

Let us also point out some finite sample phenomena that are not reflected in the asymptotic approximations. We estimated the h-function on simulated data of a Frank copula with sample size $n = 250$ and values for Kendall's τ of 0.3 and 0.7. The



(a) $\tau = 0.3$

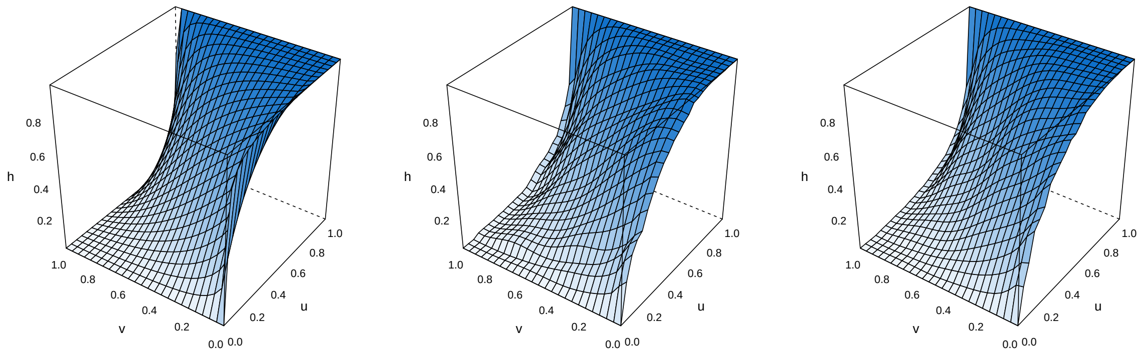
(b) $\tau = 0.7$

Figure 4.3: Estimation of $h(0.5|v)$, $v \in [0, 1]$, for simulated data ($n = 250$) of a Frank copula with Kendall's $\tau = 0.3, 0.7$. True function (solid line), mean of 500 estimates (dashed line) and estimated pointwise 90%-confidence bands (dotted lines). Bandwidths were selected by the reference rule discussed in Section 4.3.

procedure was repeated $N = 500$ times and mean estimates and estimated pointwise 90%-confidence bands were obtained. Since the estimation is conducted for a fixed u, we will also fix the value of u to gain better insights.

In Figure 4.3a we see $h(0.5|v)$, $v \in [0, 1]$, of a Frank copula with Kendall's $\tau = 0.3$ (solid line). The mean estimates are drawn as dashed line and confidence bands as dotted lines. Most notably, we seem to have a strongly increased variance close to the boundaries which is not reflected in the asymptotic approximation. In the discussion of Figure 4.1 we already brought up the cause of this effect: the adaptive weighting increases the influence of an observation to the estimate in this observation. Furthermore, the second derivative w.r.t. v is almost constantly zero in this model, so, asymptotically, there is almost no bias. We can, however, observe a small bias as the mean estimate lies above the true function at the left boundary and below it at the right boundary. This effect is again due to the asymmetric shape of the weightings we mentioned in the discussion of Figure 4.1. Close to the boundary, the proportion of observations entering an estimate is unequally distributed to the right and left of the location of the estimate. At the left boundary for instance, an increased proportion of observations right from the estimation point enter the estimate and lead to a negative bias. Asymptotically, the effect of the weights disappears as the bandwidth vanishes and can therefore not be found in asymptotic approximations. For a larger sample size, these effects will mostly disappear.

In Figure 4.3b we see $h(0.5|v)$ in the case of strong dependence ($\tau = 0.7$). In contrast to the previous situation, we see neither bias nor variance at the boundary regions. Here, the true f-function attains zero and one respectively, so bias and variance tend to zero. We do observe bias and variance in points where the curvature is high, though. Whereas the bias is reflected in the asymptotic approximation by means of the second-order derivative, the effect of the curvature to the variance is hidden in the $o\Big(1/(nb_n^2)\Big)$ term. Overall, we can conclude that the finite-sample effects seem to outweigh the issues detected in the asymptotic analysis of the estimators for moderate sample size.



(a) $h(u|v)$ **(b)** $\widehat{h}_{250}^{(LL)}(u|v)$ **(c)** $\widehat{h}_{1\,000}^{(LL)}(u|v)$

Figure 4.4: Estimation of the h-function for simulated data of a Frank copula with Kendall's $\tau = 0.7$. True function (a) as well as estimates based on samples of size 250 (b) and 1 000 (c). Bandwidths were selected by the reference rule discussed in Section 4.3.

In Figure 4.4 we see exemplary estimates of the whole h-function on simulated data from a Frank copula with Kendall's $\tau = 0.7$ based on samples of size 250 (a) and 1 000 (b). Overall, we can conclude that the estimator is appropriately functioning and gives reasonably accurate results.

4.3 Bandwidth selection

Bandwidth selection for kernel regression estimators is a well studied field. Usually, it is advised to use data-driven criteria such as Akaike's (corrected) information criterion (see Hurvich et al., 1998) or the generalized cross-validation of Craven and Wahba (1978). Our problem is a little different though, since we do not have to perform a single, but a high number of separate regressions. While an averaging approach over criteria computed on several distinct regressions is certainly possible, the computation time is severely increased and the methods provided quite unstable results in numerical experiments.

A stable and computationally appealing alternative can be established by considering an asymptotic approximation of the MISE. With Proposition 4.1 we obtain

$$\begin{aligned}\mathrm{AMISE}\big[\widehat{h}_n^{(LL)}\big] &= \int_0^1\int_0^1 \left[\frac{b_n^4\sigma_K^4}{4}h_{vv}^2(u|v) + \frac{d_K h(u|v)\big(1-h(u|v)\big)}{nb_n}\right]dudv \\ &= \frac{b_n^4\sigma_K^4}{4}\underbrace{\int_0^1\int_0^1 h_{vv}^2(u|v)dudv}_{:=\alpha} + \frac{d_K}{nb_n}\underbrace{\int_0^1\int_0^1 h(u|v)\big(1-h(u|v)\big)dudv}_{:=\beta},\end{aligned}$$

provided the integrals exist. It is minimized by

$$b_n = \left(4\frac{d_K\beta}{\sigma_K^4\alpha}\right)^{1/5} n^{-1/5}.$$

The above expression depends on the unknown h-function $h(u|v)$. In practice we can choose a parametric copula family that ensures integrability (e.g. the Frank copula) and fix its parameter by matching the theoretical and empirical Kendall's τ. The optimal bandwidth for this reference copula can be easily computed by numerical integration which is usually a matter of tenths of a second.

This bandwidth selection rule was applied in all estimates in Figures 4.3 and 4.4. The rule seems to give approximately adequate bandwidths. However, the estimator may have a slight tendency to undersmooth. This is caused by the finite-sample variance effects that are neglected by asymptotic approximations (see discussion in Section 4.2). Improved selection rules based on theoretical considerations or computational attractive data-driven methods may be an interesting topic for further research, but are beyond the scope of this thesis. A practitioner could just increase the bandwidth obtained by the above procedure by a small factor greater than one.

Chapter 5

Kernel estimation of vine copulas

This chapter deals with kernel estimation of a full vine copula density in arbitrary dimensions. Having introduced the necessary bivariate estimation techniques in the previous chapters, it just remains to put the pieces together. After a short description of the estimation procedure, we will demonstrate its abilities by means of two small simulation examples and a real data application. As the close of this chapter, possible directions for future research will be discussed.

5.1 The estimation procedure

In the notation of Section 2.1.5, the density of a d-dimensional R-vine copula C is given as

$$c(\boldsymbol{u}) = \prod_{k=}^{d-1} \prod_{e \in E_k} c_{a_e,b_e;D_e}\Big(C_{a_e|D_e}(u_{a_e}|\boldsymbol{u}_{D_e}), C_{b_e|D_e}(u_{b_e}|\boldsymbol{u}_{D_e})\Big).$$

Recall further that the arguments of the bivariate copula densities can be expressed as a recursive application of h-functions related to pairs in the vine (c.f. equation (2.2)). We can therefore focus solely on bivariate objects in order to estimate all required components of the density.

To obtain estimates of all bivariate densities and h-functions, we will make use of the sequential estimation approach (see Definition 2.12), where we can use any of the kernel estimators for bivariate copula densities and h-functions we have introduced in the previous two chapters. As the result, we get a fully nonparametric kernel estimate of the R-vine copula density.

Example 5.1. *To clarify the estimation procedure, we will go through the steps of the sequential kernel estimation of a four-dimensional R-vine copula corresponding to the tree sequence given in Figure 5.1. Assume we are given iid samples* $(\boldsymbol{u}_1, \dots, \boldsymbol{u}_4) := (u_1^{(i)}, \dots, u_4^{(i)})_{i=1,\dots,n}$ *from the R-vine copula.*

1. *Estimation in* T_1*:*

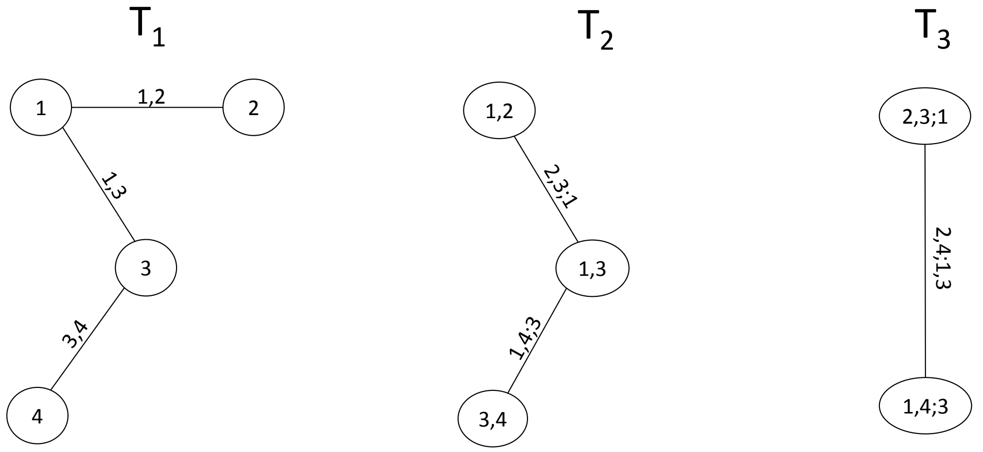


Figure 5.1: R-vine tree sequence for Example 5.1.

(i) Based on the samples $(\boldsymbol{u}_1, \boldsymbol{u}_2) \sim C_{1,2}, (\boldsymbol{u}_1, \boldsymbol{u}_3) \sim C_{1,3}, (\boldsymbol{u}_3, \boldsymbol{u}_4) \sim C_{3,4}$, obtain kernel estimates of the bivariate copula densities. For all $u_1, u_2, u_3, u_4 \in [0, 1]$, this gives us

$$\widehat{c}_{1,2}(u_1, u_2), \quad \widehat{c}_{1,3}(u_1, u_3), \quad \widehat{c}_{3,4}(u_3, u_4).$$

2. Transition to T_2:

(i) Based on the samples $(\boldsymbol{u}_1, \boldsymbol{u}_2) \sim C_{1,2}, (\boldsymbol{u}_1, \boldsymbol{u}_3) \sim C_{1,3}, (\boldsymbol{u}_3, \boldsymbol{u}_4) \sim C_{3,4}$, obtain kernel estimates of the required h-functions. For all $u_1, u_2, u_3, u_4 \in [0, 1]$, this gives us

$$\widehat{h}_{2|1}(u_2|u_1), \quad \widehat{h}_{3|1}(u_3|u_1), \quad \widehat{h}_{1|3}(u_1|u_3), \quad \widehat{h}_{4|3}(u_4|u_3).$$

(ii) Define the pseudo-samples

$$\begin{aligned} u_{2|1}^{(i)} &:= \widehat{h}_{2|1}(u_2^{(i)}|u_1^{(i)}), \quad & u_{3|1}^{(i)} &:= \widehat{h}_{3|1}(u_3^{(i)}|u_1^{(i)}), \\ u_{1|3}^{(i)} &:= \widehat{h}_{1|3}(u_1^{(i)}|u_3^{(i)}), \quad & u_{4|3}^{(i)} &:= \widehat{h}_{4|3}(u_4^{(i)}|u_3^{(i)}), \end{aligned}$$

for all $i = 1, \ldots n$.

3. Estimation in T_2:

(i) Based on the pseudo-samples $(\boldsymbol{u}_{2|1}, \boldsymbol{u}_{3|1}) \sim C_{2,3;1}$, $(\boldsymbol{u}_{1|3}, \boldsymbol{u}_{4|3}) \sim C_{1,4;3}$, obtain kernel estimates of the bivariate copula densities. For all $u_{2|1}, u_{3|1}, u_{1|3}, u_{4|3} \in [0, 1]$, this gives us

$$\widehat{c}_{2,3;1}(u_{2|1}, u_{3|1}), \quad \widehat{c}_{1,4;3}(u_{1|3}, u_{4|3}).$$

4. Transition to T_3:

(i) *Based on the pseudo-samples* $(\boldsymbol{u}_{2|1}, \boldsymbol{u}_{3|1}) \sim C_{2,3;1}$, $(\boldsymbol{u}_{1|3}, \boldsymbol{u}_{4|3}) \sim C_{1,4;3}$, *obtain kernel estimates of the required h-functions. For all* $u_{2|1}, u_{3|1}, u_{1|3}, u_{4|3} \in [0,1]$, *this gives us*

$$\widehat{h}_{2|3;1}(u_{2|1}|u_{3|1}), \quad \widehat{h}_{4|1;3}(u_{4|3}|u_{1|3}).$$

(ii) *Define the pseudo-samples*

$$u_{2|3;1}^{(i)} := \widehat{h}_{2|3;1}(u_{2|1}^{(i)}|u_{3|1}^{(i)}), \quad u_{4|1;3}^{(i)} := \widehat{h}_{4|1;3}(u_{4|3}^{(i)}|u_{1|3}^{(i)}),$$

for all $i = 1, \dots n$.

5. *Estimation in* T_3*:*

(i) *Based on the pseudo-samples* $(\boldsymbol{u}_{2|3;1}, \boldsymbol{u}_{4|1;3}) \sim C_{2,4;1,3}$, *obtain a kernel estimate of the bivariate copula density. For all* $u_{2|3;1}, u_{4|1;3}, \in [0,1]$, *this gives us*

$$\widehat{c}_{2,4;1,3}(u_{2|3;1}, u_{4|1;3}).$$

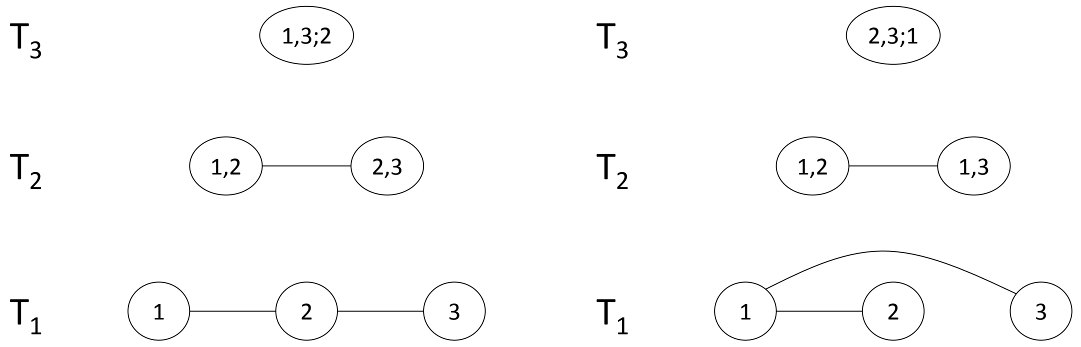
Finally, the kernel density estimate of the full R-vine is given by the product of the estimated bivariate copula densities. For all $(u_1, u_2, u_4, u_4) \in [0,1]^4$,

$$\begin{aligned}
\widehat{c}(u_1, u_2, u_3, u_4) &= \widehat{c}_{1,2}(u_1, u_2) \cdot \widehat{c}_{1,3}(u_1, u_3) \cdot \widehat{c}_{3,4}(u_3, u_4) \\
&\quad \cdot \widehat{c}_{2,3;1}(u_{2|1}, u_{3|1}) \cdot \widehat{c}_{1,4;3}(u_{1|3}, u_{4|3}) \\
&\quad \cdot \widehat{c}_{2,4;1,3}(u_{2|3;1}, u_{4|1;3}) \\
&= \widehat{c}_{1,2}(u_1, u_2) \cdot \widehat{c}_{1,3}(u_1, u_3) \cdot \widehat{c}_{3,4}(u_3, u_4) \\
&\quad \cdot \widehat{c}_{2,3;1}\Big(\widehat{h}_{2|1}(u_2|u_1), \widehat{h}_{3|1}(u_3|u_1)\Big) \cdot \widehat{c}_{1,4;3}\Big(\widehat{h}_{1|3}(u_1|u_3), \widehat{h}_{4|3}(u_4|u_3)\Big) \\
&\quad \cdot \widehat{c}_{2,4;1,3}\left(\widehat{h}_{2|3;1}\Big(\widehat{h}_{2|1}(u_2|u_1)\Big|\widehat{h}_{3|1}(u_3|u_1)\Big), \widehat{h}_{4|1;3}\Big(\widehat{h}_{4|3}(u_4|u_3)\Big|\widehat{h}_{1|3}(u_1|u_3)\Big)\right).
\end{aligned}$$

We can expect consistency of this estimator when the density and h-function estimators are consistent. Furthermore, the accuracy of the R-vine density estimator will be directly related to the accuracy of these two components. However, a duly investigation of the theoretical properties of such an estimator seems quite cumbersome, if not infeasible.

5.2 Simulations

In the following, we will illustrate the estimator's ability in two small simulation examples. For demonstrative purposes, both examples will just involve three-dimensional vines. Afterwards we give a short discussion of the computational demands of the estimator.



(a) $\mathcal{V}$

(b) $\tilde{\mathcal{V}}$

Figure 5.2: R-vine tree sequences of the true ($\mathcal{V}$) and alternative ($\tilde{\mathcal{V}}$) structures used in the simulation examples.

The setup

The general setup in both examples is as follows: We specify a vine copula model with tree sequence $\mathcal{V}$ (see Figure 5.2a) and a set of bivariate copulas $\mathcal{C}$ (see Table 5.1). We simulate from this model and obtain parametric and kernel estimates using structure $\mathcal{V}$. Clearly, the parametric estimator will perform better in this case, because all the bivariate copula families we try to estimate belong to known parametric families. In practice, this does not have to be the case and also the true structure is usually unknown. So we will also obtain estimates using an alternative structure $\tilde{\mathcal{V}}$ (Figure 5.2b). The pair-copulas in this alternative model do not necessarily conform with parametric models anymore. Of course, the parameters in the two examples will be set in a way that enables us to illustrate the advantages of the kernel estimators. Our view will therefore be a little biased.

	Family	Kendall's τ	Parameter
$c_{1,2}$	Joe	0.35	$\theta = 2$
$c_{2,3}$	Joe	0.35	$\theta = 2$
$c_{1,3;2}$	90° rotated Gumbel	-0.85	$\theta = 8$

(a) Example 1

	Family	Kendall's τ	Parameter(s)
$c_{1,2}$	Joe	0.51	$\theta = 3$
$c_{2,3}$	Clayton	0.6	$\theta = 3$
$c_{1,3;2}$	Tawn	0.74	$(\theta, \alpha_1, \alpha_2) = (12, 0.8, 1)$

(b) Example 2

Table 5.1: Specifications of the simulation models in the two examples.

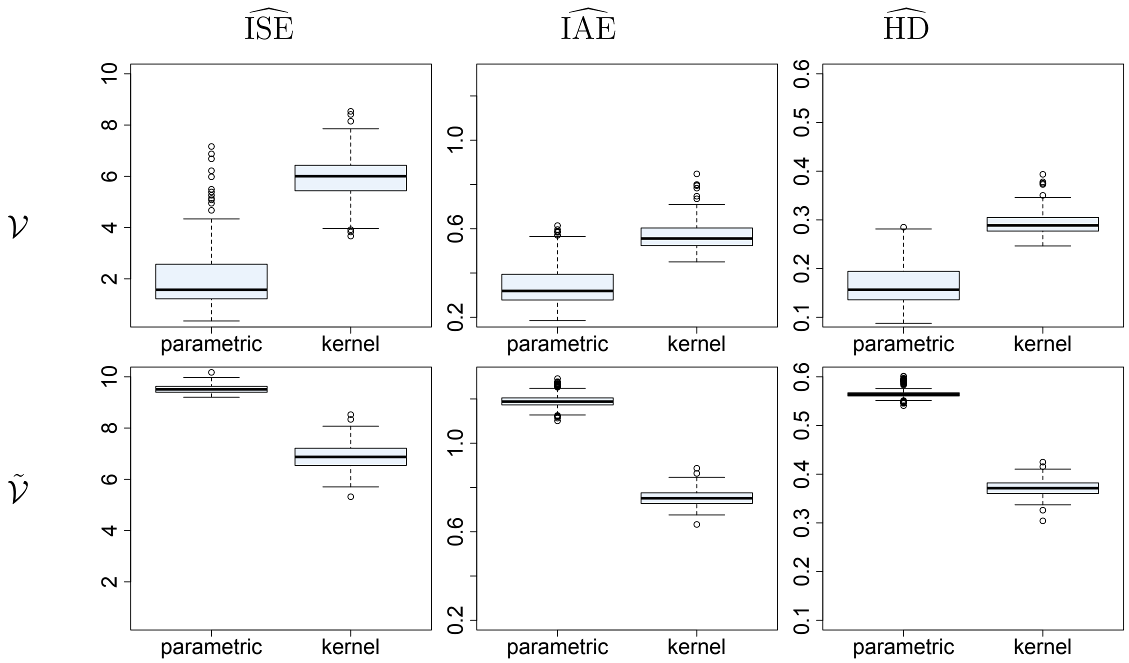


Figure 5.3: Example 1: Box plots of the estimated integrated squared error, integrated absolute error, and Hellinger distance for parametric and kernel estimators. Data was simulated with structure $\mathcal{V}$ and pair-copulas given in Table 5.1a. Results are based on estimates using structure $\mathcal{V}$ (top row) and $\tilde{\mathcal{V}}$ (bottom row).

The performance of both estimation approaches was measured as follows. For each scenario:

- Simulate samples of size $n = 500$.
- Measure distance between true and estimated density by integrated squared error (ISE), integrated absolute error (IAE) and Hellinger distance (HD) (defined analogous to Section 3.6). For numerical convenience, all measures are estimated on an equally spaced grid of 25^3 points.
- The experiment is repeated 500 times.

For parametric estimation, we use the `RVineCopSelect` function of the `VineCopula` package (Schepsmeier et al., 2013) allowing for the full set of implemented families (including all families used in this thesis and more; for details see package description). The function estimates each pair-copula (and thereby h-function) by maximum-likelihood for a large variety of families and chooses the best model with the best AIC. For kernel estimation we employ the approach described in the last section where we use the transformation local likelihood (TLL) estimator for the densities and the local linear (LL) estimator for h-functions.

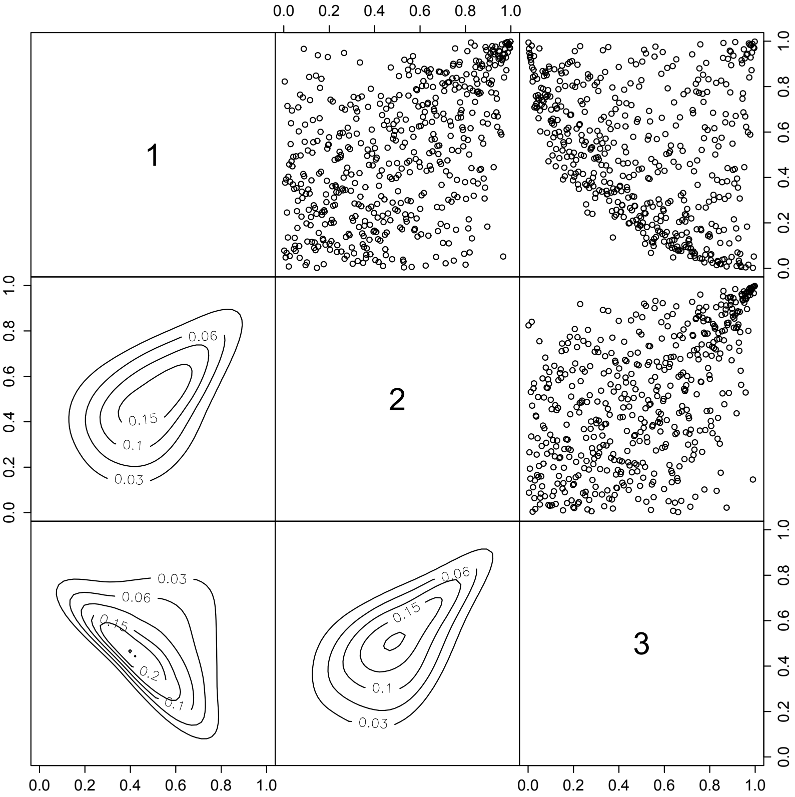


Figure 5.4: Example 1: Pairwise scatter plots of simulated data (above diagonal) and corresponding marginal normal contour plots of kernel estimates (below diagonal) for the simulation model ($n = 500$).

Example 1

Let us start with a look on the performance and later investigate what caused the outcome. Figure 5.3 shows boxplots of the three performance measures, one in each column. They were obtained from estimates using the true structure $\mathcal{V}$ (first row), as well as the alternative structure $\tilde{\mathcal{V}}$ (second row). Box plots for a particular measure are kept on the same scale for better comparability.

Not surprisingly, the parametric estimator outperforms the kernel estimator, when the true structure ($\mathcal{V}$) is used. It is interesting to see the variability in the performance, though. Occasionally, the parametric estimator performs just 'as bad' as the kernel estimator. This occurs when the parametric model selection procedure chooses the wrong family for one or more of the three pair-copulas. This becomes an even bigger issue in higher dimensions, as the number of pairs one has to estimate increases very fast, thereby increasing the probability of misspecification. For estimates based on the 'wrong' structure, $\tilde{\mathcal{V}}$, the picture is reversed. For all three measures, the kernel estimator does significantly better. Also, there is very little variability in the accuracy of the parametric estimator which indicates a systematic failure.

To explain what is going on in this example, it is important to recall that there are different pair-copulas to be estimated for structures $\mathcal{V}$ resp. $\tilde{\mathcal{V}}$. For example, in

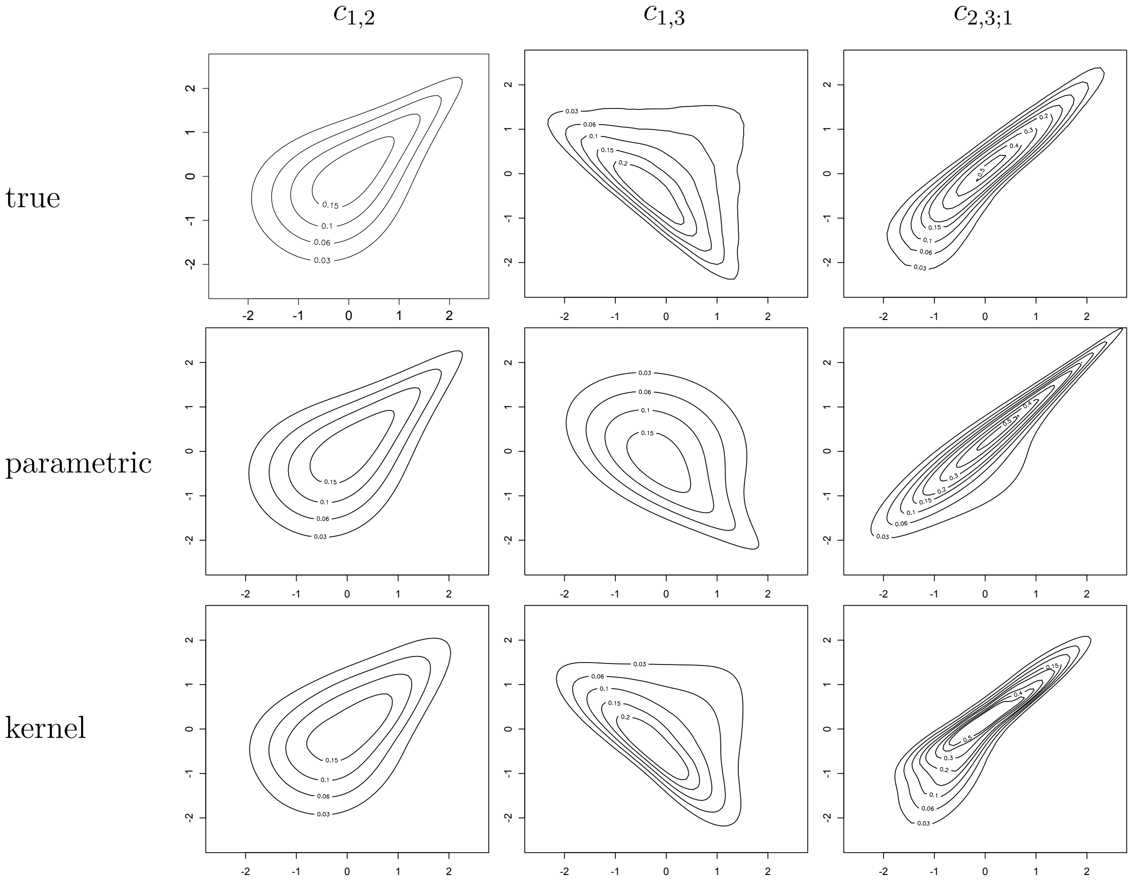


Figure 5.5: Example 1: Estimated marginal normal contours using structure $\tilde{\mathcal{V}}$. A kernel estimate based on 10^5 samples ('true', top row) as well as parametric (middle row) and kernel (bottom row) estimates based on 500 samples.

the first tree of $\mathcal{V}$ we estimate the copulas of the pairs $\{1,2\}$ and $\{2,3\}$, whereas in the first tree of $\tilde{\mathcal{V}}$ we estimate the copulas of the pairs $\{1,2\}$ and $\{1,3\}$. In the specification of the simulation model, we use structure $\mathcal{V}$ and thereby specify the copulas for pairs $\{1,2\}$ and $\{2,3\}$, while the copula corresponding to $\{1,3\}$ is not given explicitly. With the parameters we chose in this examples, the latter turns out to be very hard to capture by a parametric model. This can be seen in Figure 5.4, where pairwise scatter plots for all pairs and marginal normal contour plots of kernel density estimates are shown. The copula of the pair $\{1,3\}$ reveals asymmetry in its components and does not conform with any of the parametric families introduced in Chapter 2.

Figure 5.5 shows parametric and kernel estimates of all the pair-copulas corresponding to structure $\tilde{\mathcal{V}}$, based on an exemplary sample of size 500 (second and third row). We compare them with a kernel estimate based on a sample of size 10^5 which should pretty accurately resemble the true copula densities (first row). The density $c_{1,2}$ is estimated reasonably well by both estimation techniques, the parametric approach performing a little better. In contrast, the parametric estimate is quite off for $c_{1,3}$. The estimated density corresponds to a Tawn copula which indeed

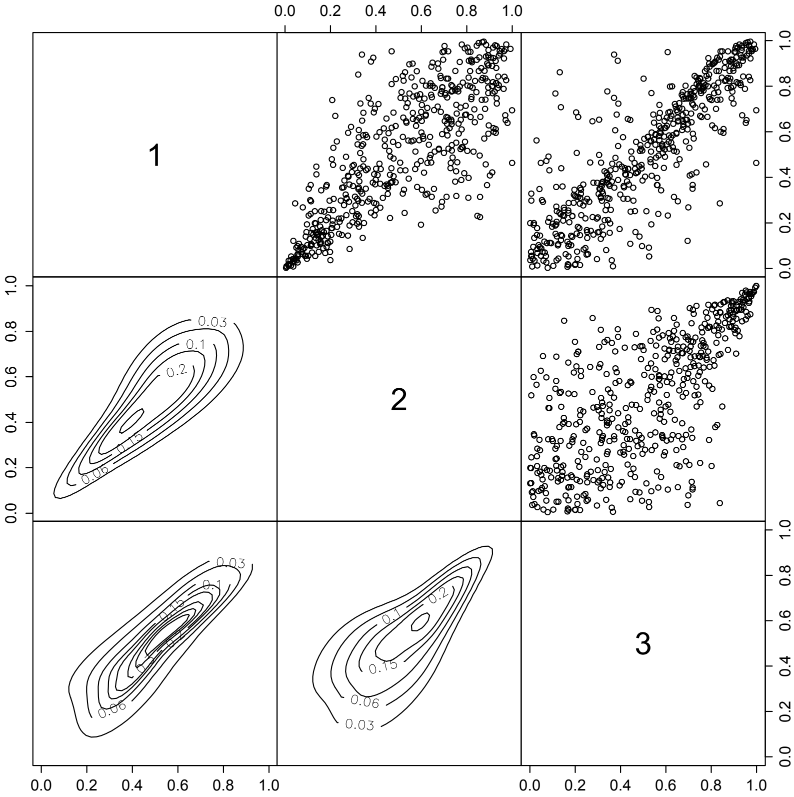


Figure 5.6: Example 2: Pairwise scatter plots of simulated data (above diagonal) and corresponding marginal normal contour plots of kernel density estimates (below diagonal) for the simulation model ($n = 500$).

reflects the asymmetry in the copula's arguments, but also features asymmetry w.r.t. upper and lower tails as well as tail dependence in the lower tail. Neither of these features seem to be present in the true density. This drastically reduces the accuracy of the whole estimate. The kernel estimator on the other hand seems quite accurate.

In the second tree, copula estimates are based on pseudo-samples that depend on estimates (of h-functions) in the first tree. Systematic errors in the first tree, will therefore result in systematic errors the second tree. This could be one reason, why the kernel estimator appears to give a much more accurate estimate for $c_{2,3;1}$ compared with the parametric estimator. The more obvious reason is that, again, the true copula density is not well approximated by any of the parametric families.

Example 2

In the previous example, a simple exploratory analysis of the pairwise scatter plots would have indicated that a parametric estimator will run into trouble. In the second example, we want to illustrate that this is not necessarily the case.

Figure 5.6 shows pairwise scatter plots of the three variables. The plot corresponding to the unspecified pair $\{1, 3\}$ does look a little suspicious. There might be a slight asymmetry in the components and the observations appear to be a little

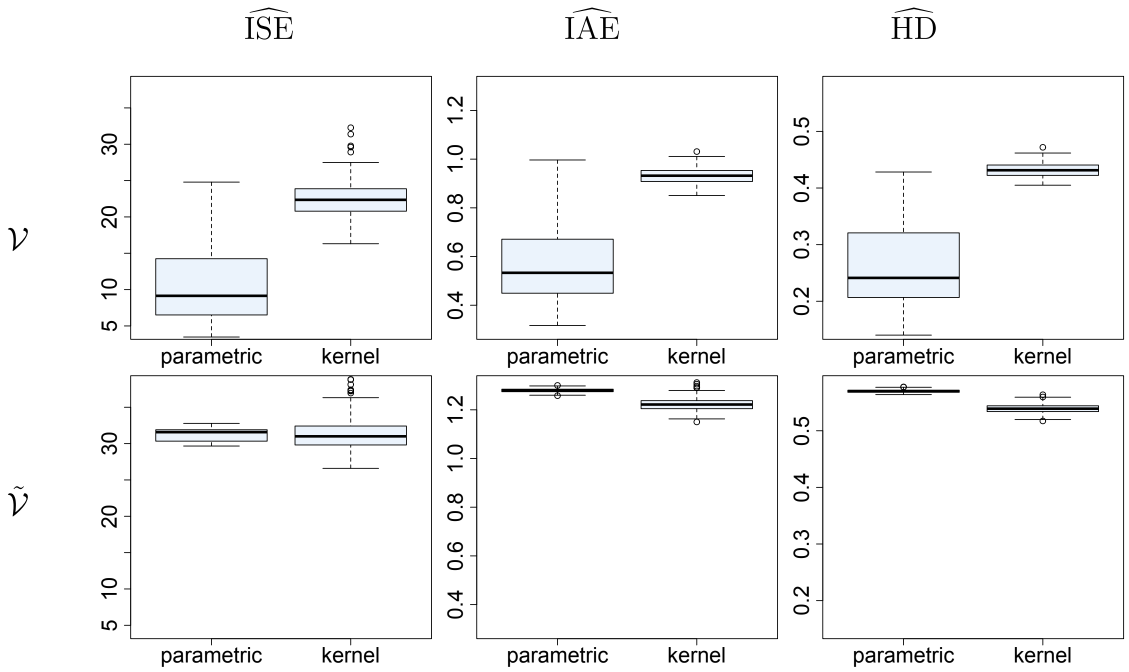


Figure 5.7: Example 2: Box plots of the estimated integrated squared error, integrated absolute error and Hellinger distance for parametric and kernel estimators. Data was simulated with structure $\mathcal{V}$ and pair-copulas given in Table 5.1b. Results are based on estimates using structure $\mathcal{V}$ (top row) and $\tilde{\mathcal{V}}$ (bottom row).

narrower clustered in the upper right corner compared with the lower left. But this is usually not something to worry about as it could very well be just a result of random variation. It can, in particular, not explain the severe drop in the parametric estimator's performance when comparing estimates based on structure $\mathcal{V}$ with estimates based on structure $\tilde{\mathcal{V}}$ (see Figure 5.7). Just as in the previous example also the variability in the performance of the parametric estimator is strongly reduced indicating a severe systematic failure. This also holds true for the kernel estimator, but a little less pronounced.

The cause for all of that is not revealed until we look at the kernel estimate of the pair-copula in the second tree. Figure 5.8 shows marginal normal contour plots of exemplary parametric and kernel estimates based on structure $\tilde{\mathcal{V}}$. While both parametric and kernel estimators appear to provide reasonable estimates of the copulas in the first tree (first two columns), the parametric estimator basically collapses in the second tree (third column). When we look at the 'true' contours it becomes clear that a) none of the parametric families is even rudimentarily adequate, and b) the complex shape of the copula is also very hard to estimate for a nonparametric estimator. This explains the poor performance of both estimation techniques. Nevertheless, the kernel estimate at least indicates that something unusual is going on in the second tree, which would have never come to light when only a parametric estimate would have been considered.

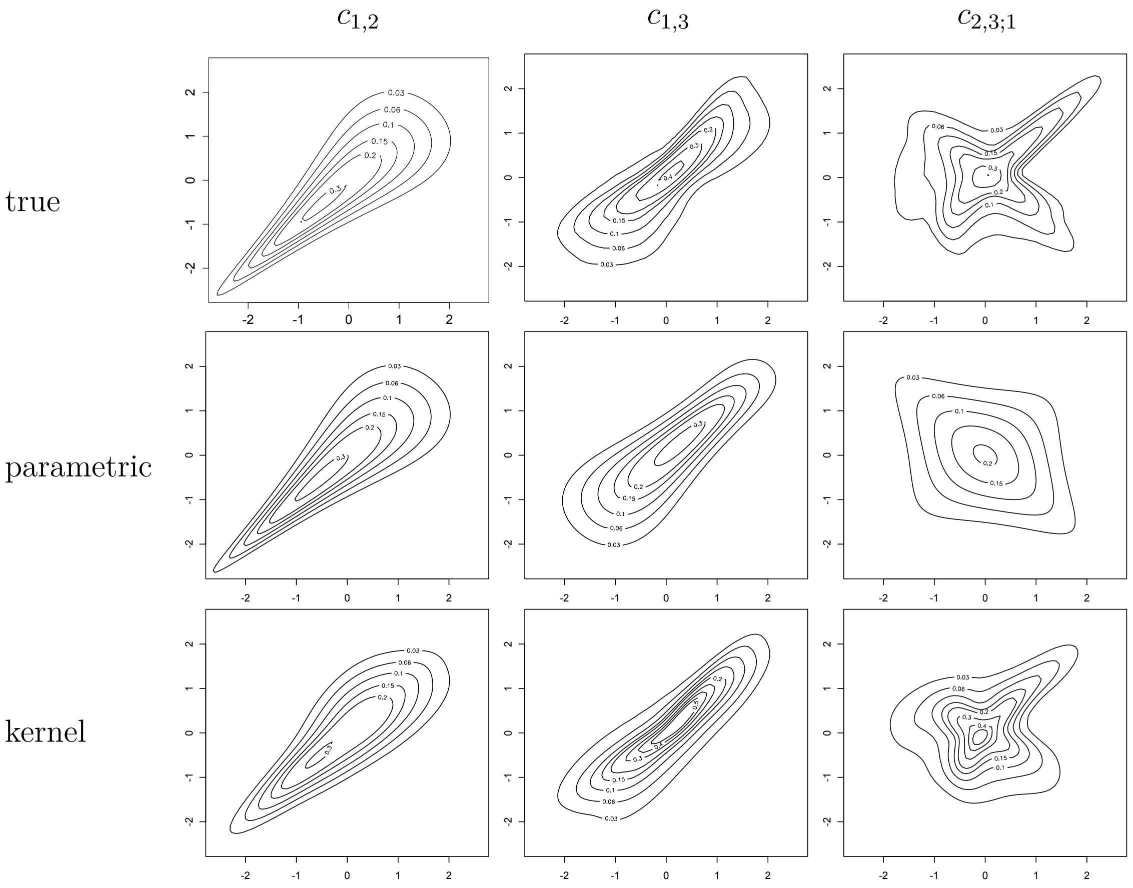
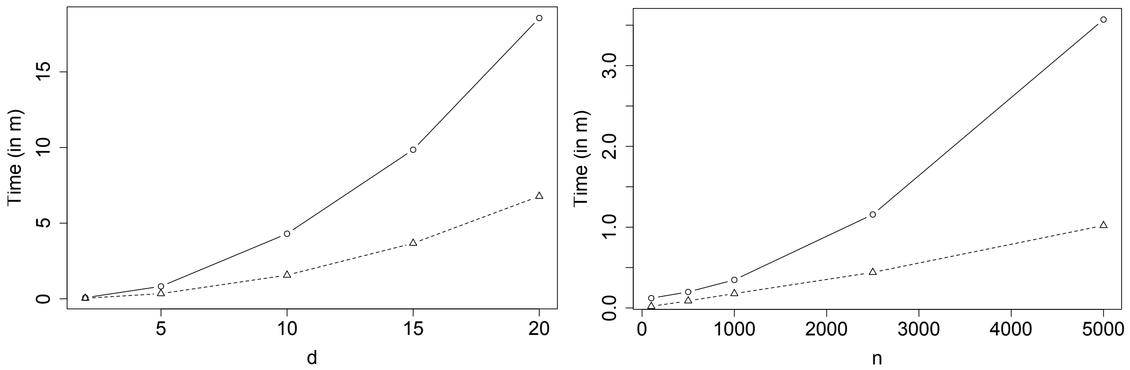


Figure 5.8: Example 2: Estimated marginal normal contours using structure $\tilde{\mathcal{V}}$. A kernel estimate based on 10^5 samples ('true', top row) as well as parametric (middle row) and kernel estimates based on 500 samples (bottom row).



(a) varying d, $n = 500$ **(b)** varying n, $d = 3$

Figure 5.9: Mean computation time over 20 runs of the kernel (solid line) and parametric (dashed line) estimators.

A note on computation time

One of the major disadvantages of the kernel estimator is that it is computationally quite expensive. In the following we give the results of a small experiment illustrating the influence of the sample size n as well as the dimension d on the computation time.

The computations are conducted on a customary Windows 7 Lenovo laptop with Intel Core Duo i3-3120 CPU @ 2.50 GHz and 8 GB RAM. We simulate data from a Gaussian copula with correlation parameter $\rho = 0.4$ and run both parametric and kernel estimation algorithms (including bandwidth selection for the latter). The structure specified before running the algorithm is a so-called *D-Vine*, that is an R-Vine where each tree consists of a single path. The particular structure should play a minor role, however. The mean computation time over 20 repetitions is reported.

Figure 5.9a shows the computation time for varying dimension where we fixed the sample size to $n = 500$. We can see that the time grows approximately quadratically in the dimension. That is because the number of pair copulas in the vine is $d(d-1)/2$ and, thus, grows quadratically. The picture is similar for varying n and fixed $d = 3$ (Figure 5.9b). Now, that might be surprising at first, since in the last to chapters we saw that the computation time of both the density and h-function estimators grew linearly in the sample size. When estimating a vine copula however, there is a second effect coming in. Increasing the sample size also leads to an increasing number of required evaluations of the h-function when defining the pseudo-observations in higher trees. The aggregation of both effects leads to the non-linear growth in computational expense.

Compared with the parametric estimator, the kernel estimator takes considerably more time. While the difference is negligible for small n and d, the gap widens quickly as sample size and dimension increase. Because of the nonlinear growth of computation time, it is also less scalable to huge data sets. Using a kernel estimator in high dimensions and/or on a large number of samples requires patience, but is still tolerable in most practical situations. For extremely large data sets or many variables, truncation of the vine as in Brechmann et al. (2012) is advisable.

Summary

To start of with, it should be said once more that we chose the parameterization of the examples in order to illustrate the advantages of a nonparametric approach. In many other cases, the easier parametric approach is sufficient and will lead to more accurate estimates. Nevertheless, we learned two important lessons:

Firstly, when some of the pair-copulas have an unusual shape (in the sense that no parametric family conforms with it), the kernel estimator can give significantly better estimates. Despite the increased complexity of the estimator, the increased accuracy can be of high value in applications, e.g. the estimation of tail probabilities etc.

Secondly, even in cases where the performance is not increased, the nonparametric estimator can help to reveal anomalies in higher trees that a parametric estimator would conceal. At the very least, it can serve as a warning signal that a more careful modeling approach is necessary. Often, the pitfalls can easily be avoided by changing the structure of the model. In general, the kernel estimator was shown to be a very powerful tool for exploratory analysis of higher-dimensional data.

In practice time is an important factor. In this regard, the kernel estimator has a clear disadvantage over the computationally more simple parametric approach. Still, the time required for computing the kernel estimate is in the order of minutes in most practical situations and, therefore, a viable tool for estimation and exploratory analysis.

5.3 Real data application: Breast cancer diagnosis

Despite many years of intensive research and preventative measures, breast cancer continues to be the second largest cause of cancer death amongst women in the developed world (Stewart and Wild, 2014). An attractive diagnostic alternative to the popular mammography or surgical biopsy is *fine-needle aspiration (FNA)* biopsy. It is a minimally invasive technique that allows to extract tissue from suspicious lumps. Extracted cells can be analyzed subsequently either histologically or by digital imaging. In what follows, we will consider data of measurements taken on fine-needle aspirates. After a short description of the data, we will model the dependence between selected variables by a vine copula and obtain kernel estimates of its density.

The data

The data under consideration are from the Wisconsin Breast Cancer Database. They were collected at the University of Wisconsin Hospitals, Madison, from Dr. William H. Wolberg and provided by the UCI Machine Learning Repository (Bache and Lichman, 2013). It contains measurements taken on nuclei of fine-needle aspirates from 569 women. 357 masses turned out to be *benign*, the remaining 212 as *malignant* tumors. The measurements were obtained by digital imaging and contain information on size, shape and texture of the cells. In total, there are a number of 30 features computed for each individual. For more details on measures and methodology we refer to Wolberg et al. (1994). To keep things ostensive, we will restrict our analysis to the mean (over the cells of one individual) measurements of the six shape variables: smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

Analysis of dependence

The main interest is the diagnosis of cancer, that is discriminating between benign and malignant mass. We will therefore split the data and treat the two cases separately. Note that there are more observations for benign masses than for malignant tumors, so the statistical significance of our estimates is higher in that case. Again, we will use the TLL estimator for densities and the LL estimator for h-functions.

To assess the dependence between the variables, all measurements are transformed by application of the empirical marginal *cdfs* to pseudo-copula data as in Definition 2.8. Figures 5.10 and 5.11 show pairwise scatter plots of the variables as well as marginal normal contour plots of kernel density estimates with empirical Kendall's τ superimposed. In the benign samples, the strength of dependence ranges from very weak dependence (e.g. symmetry/concave points) over medium dependence (e.g. concave points/smoothness) to strong dependence (e.g. concave points/concavity). Furthermore, we observe a variety of shapes, some of which indicate a slight asymmetry (e.g. smoothness/compactness).

Considering the malignant samples, we find evidence for an increase in the strength of dependence for almost all pairs. The pair compactness/concave points is the only exception, but the small drop in Kendall's τ by just 0.02 is negligible. The strength of dependence for other pairs, e.g. compactness/symmetry, increases drastically. Moreover, the asymmetries have mostly disappeared. This could, however, also be a consequence of the smaller sample size (and larger bandwidth) in this case. Nevertheless, we can conclude that there is evidence that pairwise dependence between the measurements contains some information on whether the mass is benign or malignant.

Now assume that the joint copula density of the six variables can be captured by a simplified vine copula model. As a first step, we have to specify the structure. Since we do not have any prior information, we will utilize the sequential selection heuristic (c.f. Definition 2.13). To make the models for benign and malignant masses comparable, we will use the same structure for both cases for now. As there are more observations for the benign masses, we will apply the algorithm to these observations and use the resulting structure (see Figure 5.14) for the malignant samples as well.

Figures 5.12 shows marginal normal contour plots of all estimated pair-copulas for benign samples. Although some of the copulas in higher trees might as well correspond to independence, others reveal weak, but notable dependence, especially those corresponding to $\{1,3;2\}$, $\{1,6;2\}$, and $\{4,6;1,2,3,5\}$. This implies that there is more going on than just the simple pairwise dependence seen before. For example, the negative dependence for the copula corresponding to $\{1,3;2\}$ indicates that when the effect of variable 2 (compactness) is removed, variables 1 (smoothness)

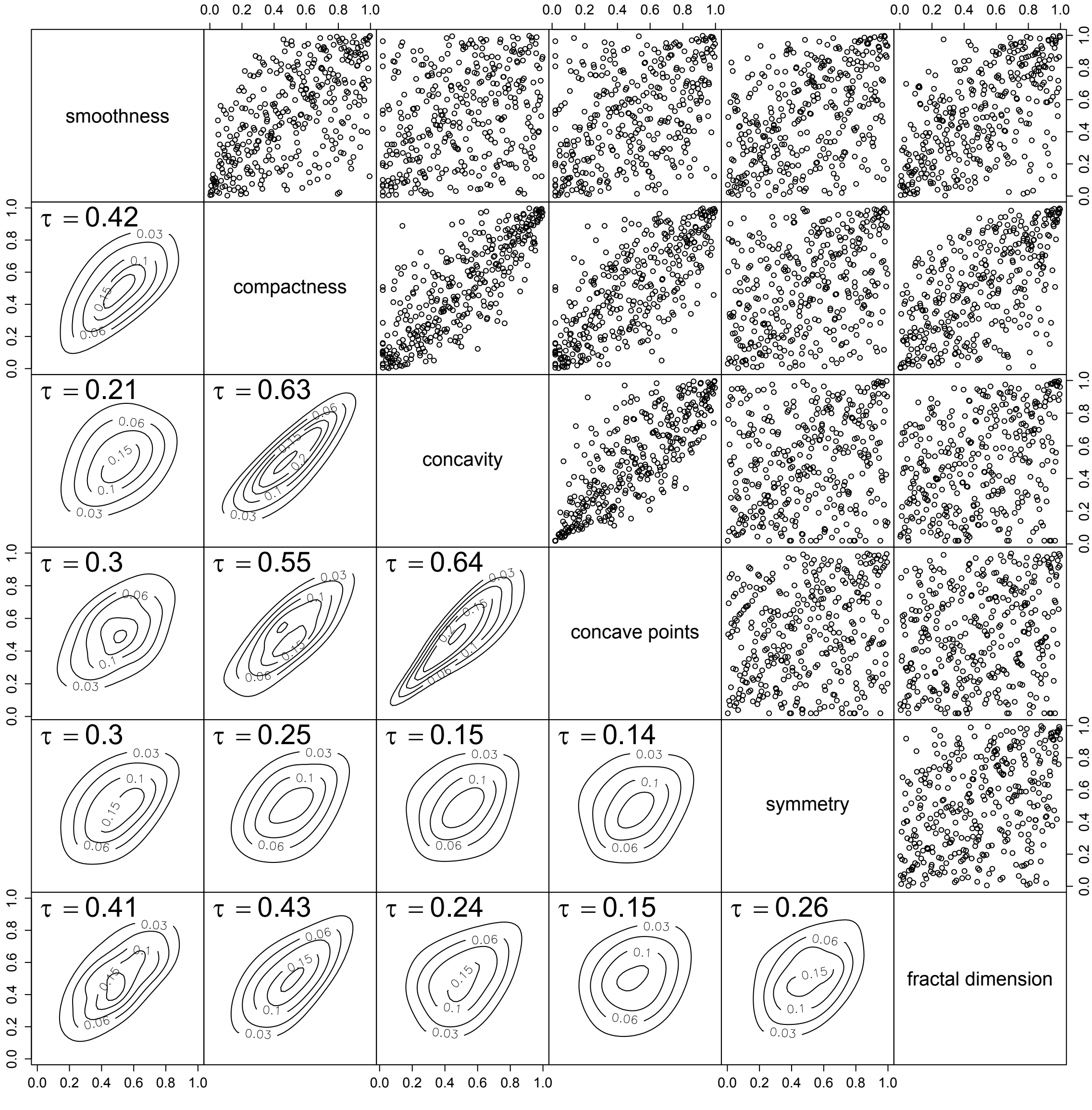


Figure 5.10: Benign samples: Pairwise scatter plots (above diagonal) and marginal normal contour plots of kernel density estimates of corresponding bivariate copulas with empirical Kendall's τ superimposed (below diagonal).

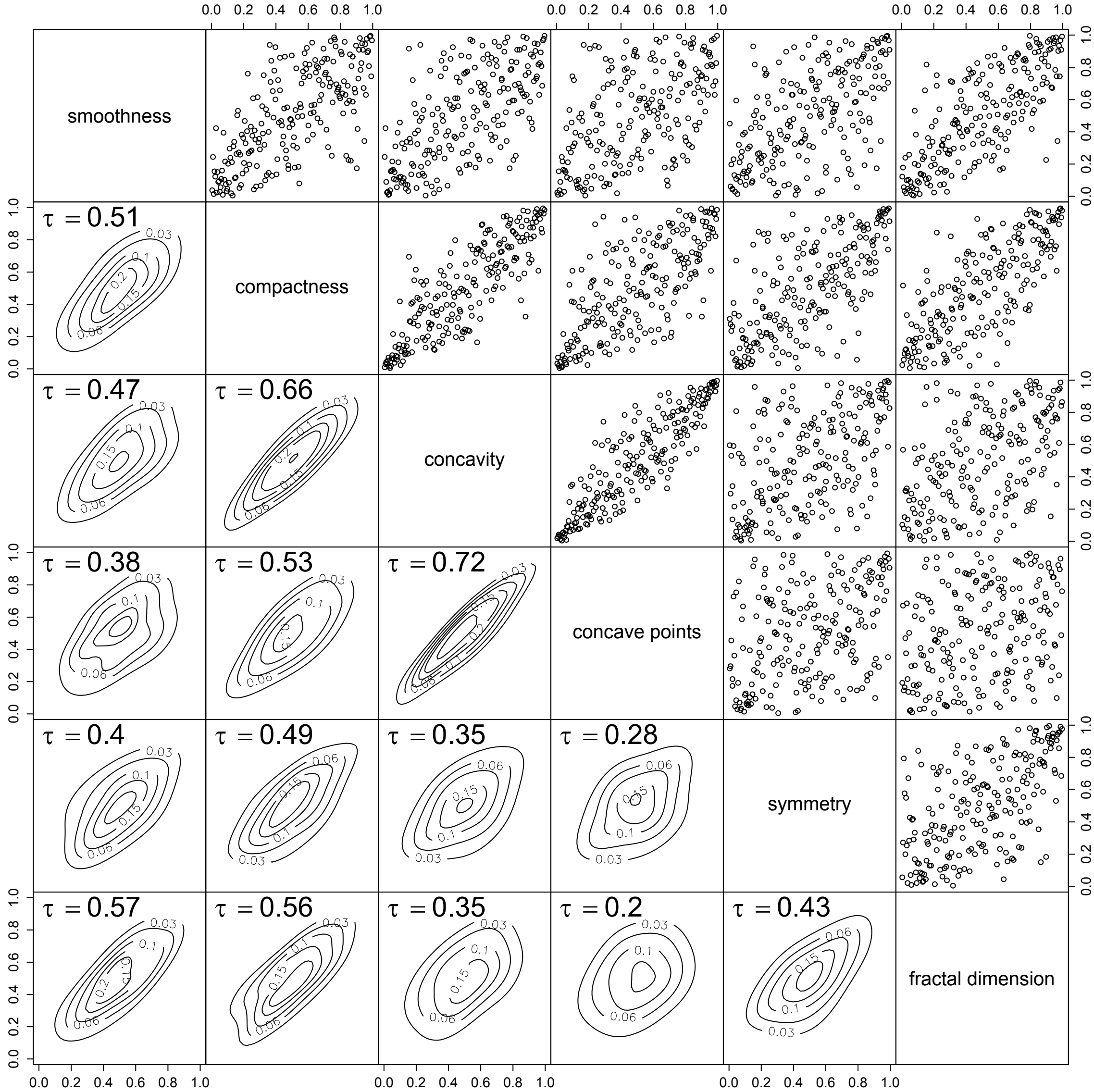


Figure 5.11: Malignant samples: Pairwise scatter plots (above diagonal) and marginal normal contour plots of kernel density estimates of corresponding bivariate copulas with empirical Kendall's τ superimposed (below diagonal).

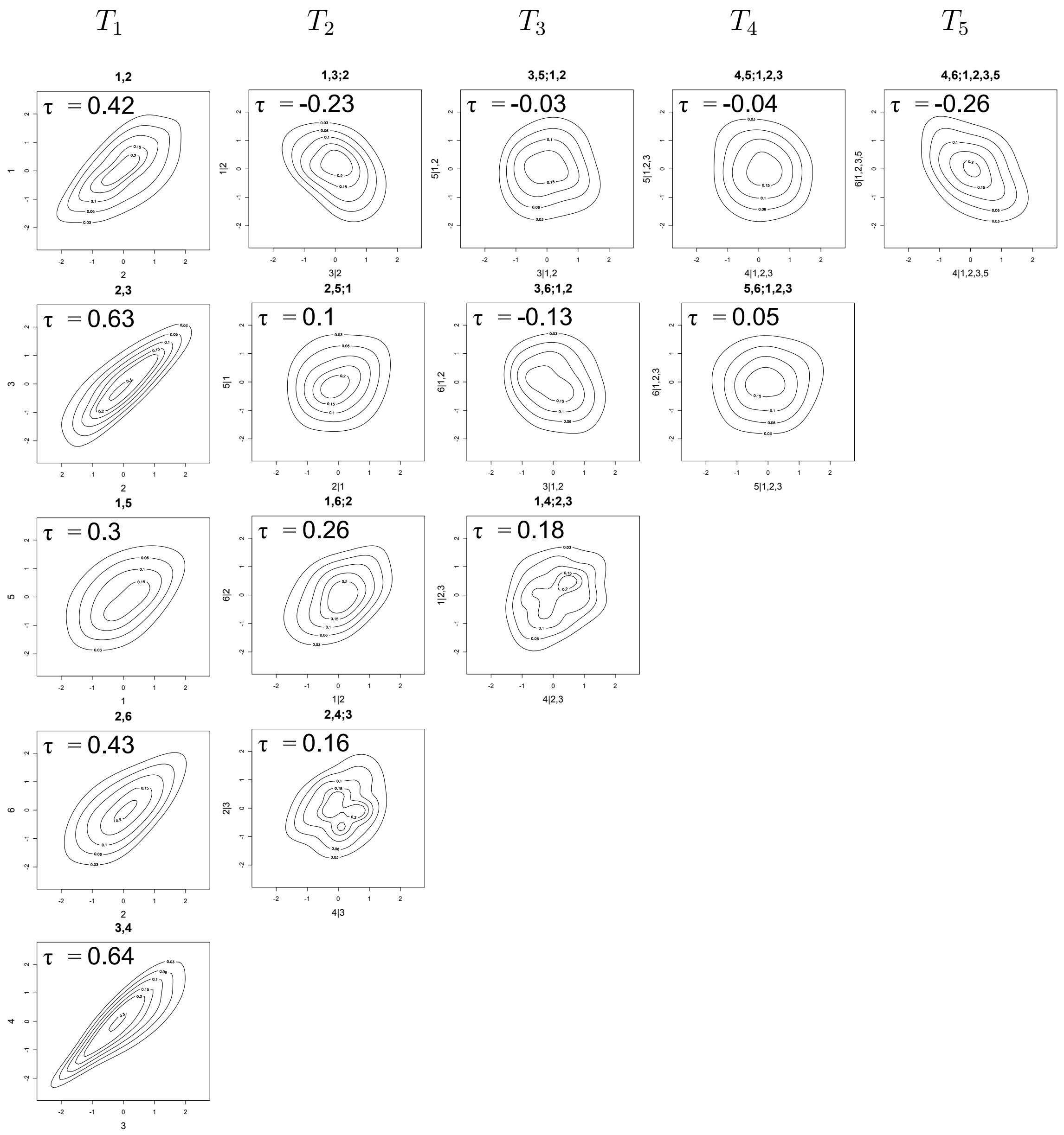


Figure 5.12: Benign samples: Marginal normal contour plots of all estimated pair-copulas in the vine copula model with empirical Kendall's τ superimposed. Variables are: 1 – smoothness, 2 – compactness, 3 – concavity, 4 – concave points, 5 – symmetry, 6 – fractal dimension.

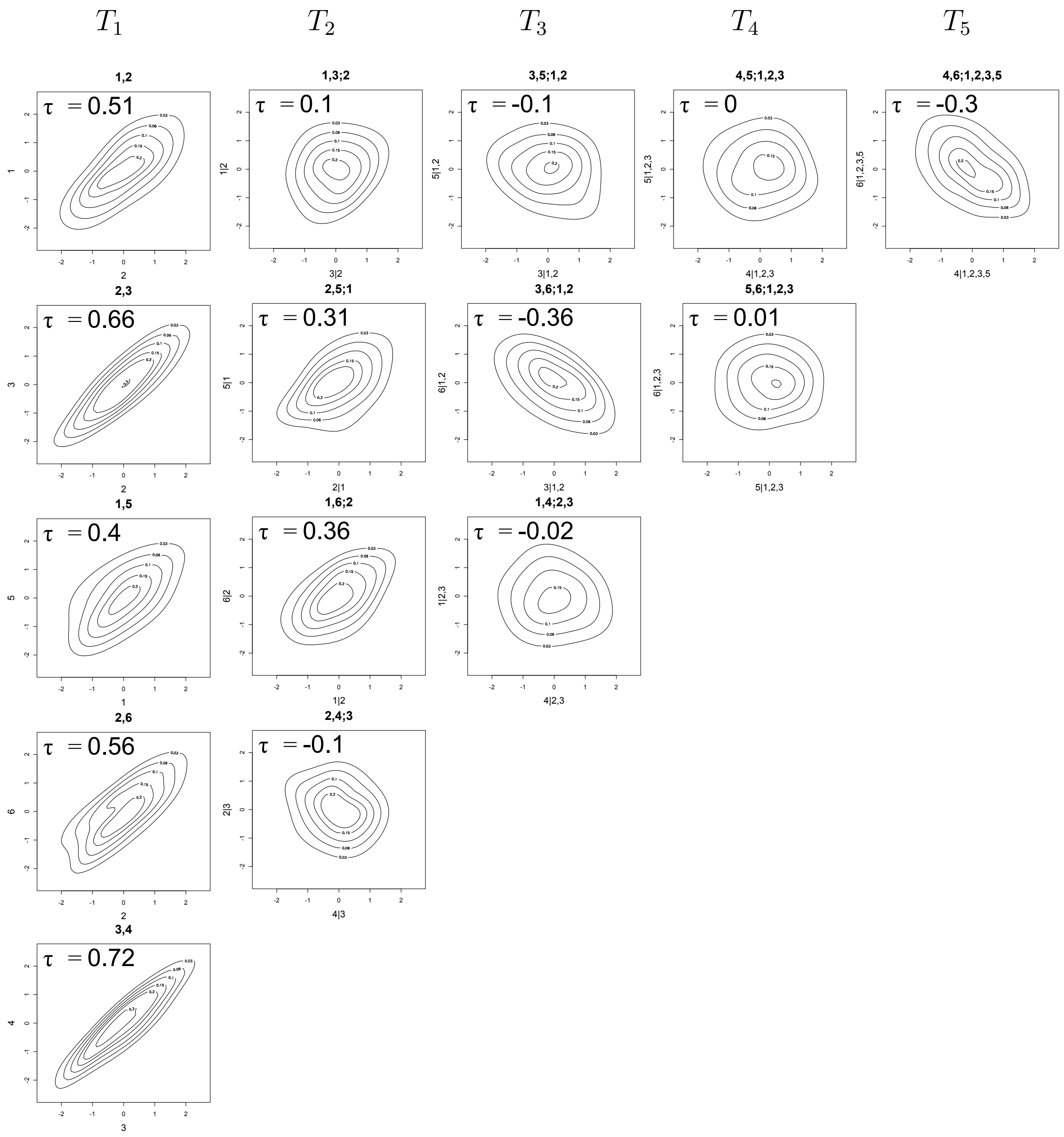


Figure 5.13: Malignant samples: Marginal normal contour plots of all estimated pair-copulas in the vine copula model with empirical Kendall's τ superimposed. Variables are: 1 – smoothness, 2 – compactness, 3 – concavity, 4 – concave points, 5 – symmetry, 6 – fractal dimension.

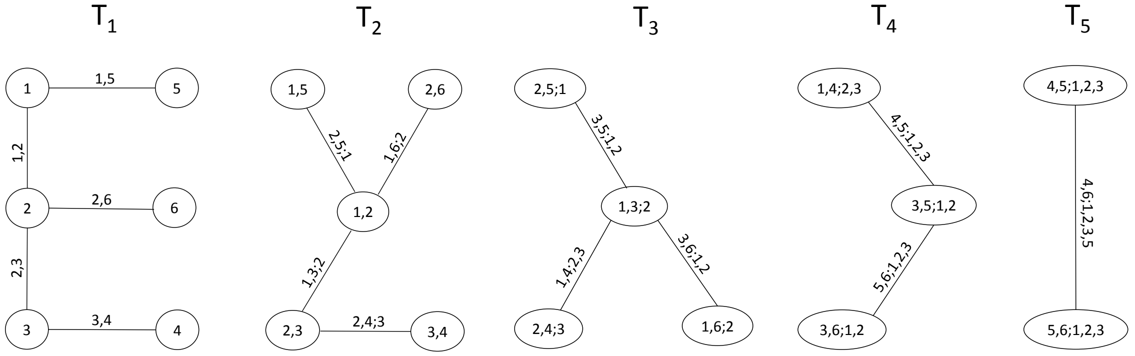


Figure 5.14: Structure obtained by the sequential selection approach assuming a vine copula model for benign observations. Variables are: 1 – smoothness, 2 – compactness, 3 – concavity, 4 – concave points, 5 – symmetry, 6 – fractal dimension.

and 3 (concavity) are negatively dependent. This stands in contrast to the positive pairwise dependence between the two variables seen in Figure 5.10. The ability to gain such insights constitutes one of the advantages of vine copula models, since such complex patterns of dependence are hard to capture and assess in standard multivariate models.

Next, we want to compare these patterns with the estimated copula densities for the malignant samples (see Figure 5.13). We observe that some of the pair-copulas are approximately the same as in the benign case, for example the pair-copulas in the fourth and fifth trees. There are some differences, though. The copula corresponding to $\{1, 3; 2\}$ does not feature negative dependence anymore. In fact, the estimated copula for the malignant samples gives slight positive dependence. On the other hand, the copula corresponding to $\{2, 4; 3\}$ changes from positive to negative dependence. Furthermore, dependence in the copulas corresponding to $\{2, 5; 1\}$ and $\{1, 6; 2\}$ notably increases, whereas dependence in the whole third tree decreases. Overall, we can infer that also the joint dependence of multiple variables could be informative about the malignancy of the cells. Actually, the differences in the second and third tree are more prominent than the differences in pairwise dependence. Hence, it should be worthwhile to take joint dependence into account.

Another interesting question is whether the structure selected by the algorithm differs between benign and malignant samples. In order to address that, we apply the structure selection procedure to the malignant samples as well. The resulting structure is shown in Figure 5.15. We find that every tree is different compared with the previous structure. Now, there is no reason to believe that the algorithm selects the true structure. However, our observation once more indicates that there is a notable difference in the dependency of measurements on benign and malignant cells.

In summary, we found evidence that the dependence between the considered features contains information about the malignancy of the cells. In fact, dissimilar dependence patterns were found for all features. This is true for both pairwise and joint dependence. This information could prove useful in building a statistical model for the diagnosis of breast cancer based on fine-needle aspirates.

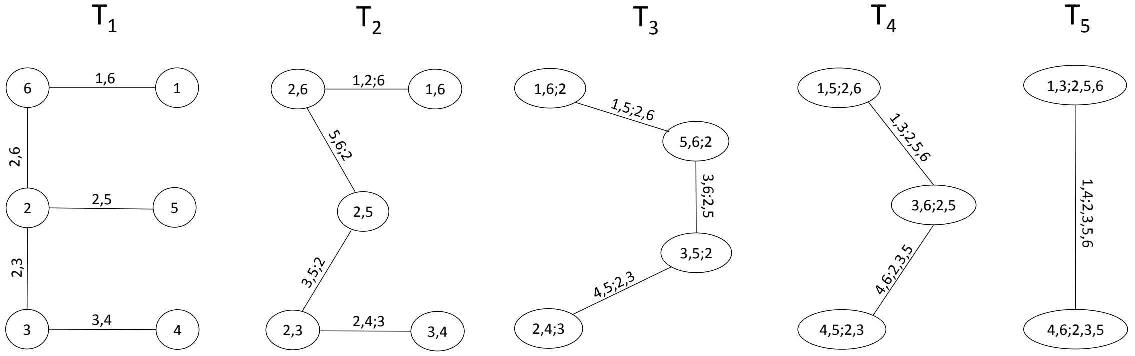


Figure 5.15: Structure obtained by the sequential selection approach assuming a vine copula model for malignant observations. Variables are: 1 – smoothness, 2 – compactness, 3 – concavity, 4 – concave points, 5 – symmetry, 6 – fractal dimension.

5.4 Further discussion

The presented approach gives a highly flexible way to estimate dependence in high-dimensional settings. It can furthermore serve as the foundation of more advanced methods which we will discuss in the following paragraphs. Some of them will be subject to the author's future research.

Combining parametric and nonparametric estimation

Throughout this thesis, we repeatedly pointed out the differences between parametric and kernel estimators. It became clear that both approaches have their right to exist and it depends on the particular data at hand which is to be preferred. A natural idea is to combine the strengths of both philosophies in one estimation approach. Within our estimation procedure it is certainly possible to estimate some of the pair-copulas parametrically and the others nonparametrically.

The difficulty is to know *a priori* which of the two philosophies is more appropriate for a particular pair. The issue calls for criteria that tell the estimator which of the estimators will be more accurate without knowing the true density. Cross-validation techniques are some kind of all-purpose solutions to such a problem, but are computationally expensive. For practical purposes, we would also like the criterion to be computationally efficient. In this regard, AIC and BIC (see Section 2.1.4) are most common for checking parametric models against each other. Nonparametric analogues can be obtained by interchanging the number of parameters in their formulas by a nonparametric analogue, the *effective number of parameters* (c.f. Hastie et al., 2001, Section 7.6). For example, a bias-corrected version of the nonparametric AIC was studied in a regression context by Hurvich et al. (1998) and shown to perform well. Despite all that, it is not at all clear how these criteria perform when checking a parametric against a nonparametric model. An attempt to tackle the parametric vs. nonparametric model selection problem directly was taken by Liu and Yang (2012) who introduced a *parametricness index* for regression models. Its performance relies heavily on asymptotics of the index, though, and its usefulness for finite samples or density estimation in general has yet to be established.

Kernel estimation of general multivariate densities

Although the estimator was built to estimate dependence between random variables, it can be valuable for estimation of high-dimensional densities in general. In practice, nonparametric density estimation is usually restricted to two- or three-dimensional settings.

The reason can be understood as follows: Assume we want to estimate a high-dimensional ($d > 3$) density at a particular point in the d-dimensional space. A nonparametric estimate usually gathers information from data in a local neighborhood of that point. The size of this neighborhood is usually controlled by a bandwidth parameter (or equivalent), which also balances the bias-variance trade-off (c.f. Section 2.2.3). A good balance usually means, that the neighborhood is as small (local) as possible and as large (global) as necessary to give a sufficiently informative estimate. In higher dimensions, chances are that there is not a single observation in a small neighborhood. When we extend the size neighborhood such that it contains a reasonable amount of observations, it can usually not be considered 'local' anymore and the huge bias renders the estimate useless. This issue is well known in statistics and usually referred to as the *curse of dimensionality* (see e.g. Scott, 2008).

Combining univariate estimation of the marginal densities and the presented kernel estimation approach of the copula density may overcome this issue. The possibility to separately estimate marginals and copula of a multivariate distribution is one of the benefits of Sklar's theorem. Univariate kernel density estimation is known to work extremely well and has been thoroughly studied in the last decades. It remains to estimate a high-dimensional copula density. But here, the curse of dimensionality can be mitigated by a vine copula model, since it builds of just two-dimensional blocks. So overall, the estimation reduces to several one- and two-dimensional tasks. A possible issue is the aggregation or multiplication of errors across the distinct estimation tasks. It should be interesting to investigate, whether such an estimator can improve over the existing multivariate nonparametric density estimators.

Estimation of non-simplified vine copulas

Another field that is yet to be explored is the nonparametric estimation of non-simplified vines. As mentioned in Section 2.1.5, so far we always assumed that the simplifying assumption is valid, i.e. that a pair-copula does not depend on realizations in lower trees (although its value might). The most urging task in this context is to develop kernel estimators of conditional copulas and h-functions. Gijbels et al. (2011) took a first step in this direction. They presented two estimators that are essentially based on an empirical copula that smoothes w.r.t. to the conditional variable. They do, however, not deal with the estimation of densities, but the copula function directly. Nevertheless, a similar approach could be used to construct an estimator of conditional copula densities and h-functions.

The mathematics underlying the analysis of such estimators are quite involved and the resulting estimators would be considerably more complex and computationally intensive than the ones presented in this thesis. A related question is therefore how to decide whether such a complex estimation procedure is necessary for a particular data set, or if a simplified model is sufficient. Hence, it should also be interesting to develop tests for constancy of the conditional copula.

Chapter 6

Conclusion

This thesis is concerned with kernel estimation of vine copula densities. We presented a novel approach that splits the estimation in two parts: estimation of bivariate copula densities, and estimation of h-functions.

For the first part, we discussed asymptotic properties and bandwidth selection for a variety of kernel density estimators. All proposed methods were compared in a simulation study where we found the transformation local likelihood (TLL) estimator of Geenens et al. (2014) to perform best overall. The second part was tackled by relating the problem of h-function estimation to a regression equation. We proposed to use the modified local-linear kernel regression of Hall et al. (1999) as a solution and gave advise for bandwidth selection. Finally, we put the two pieces together resulting in a fully nonparametric sequential estimation method for regular vine copula densities.

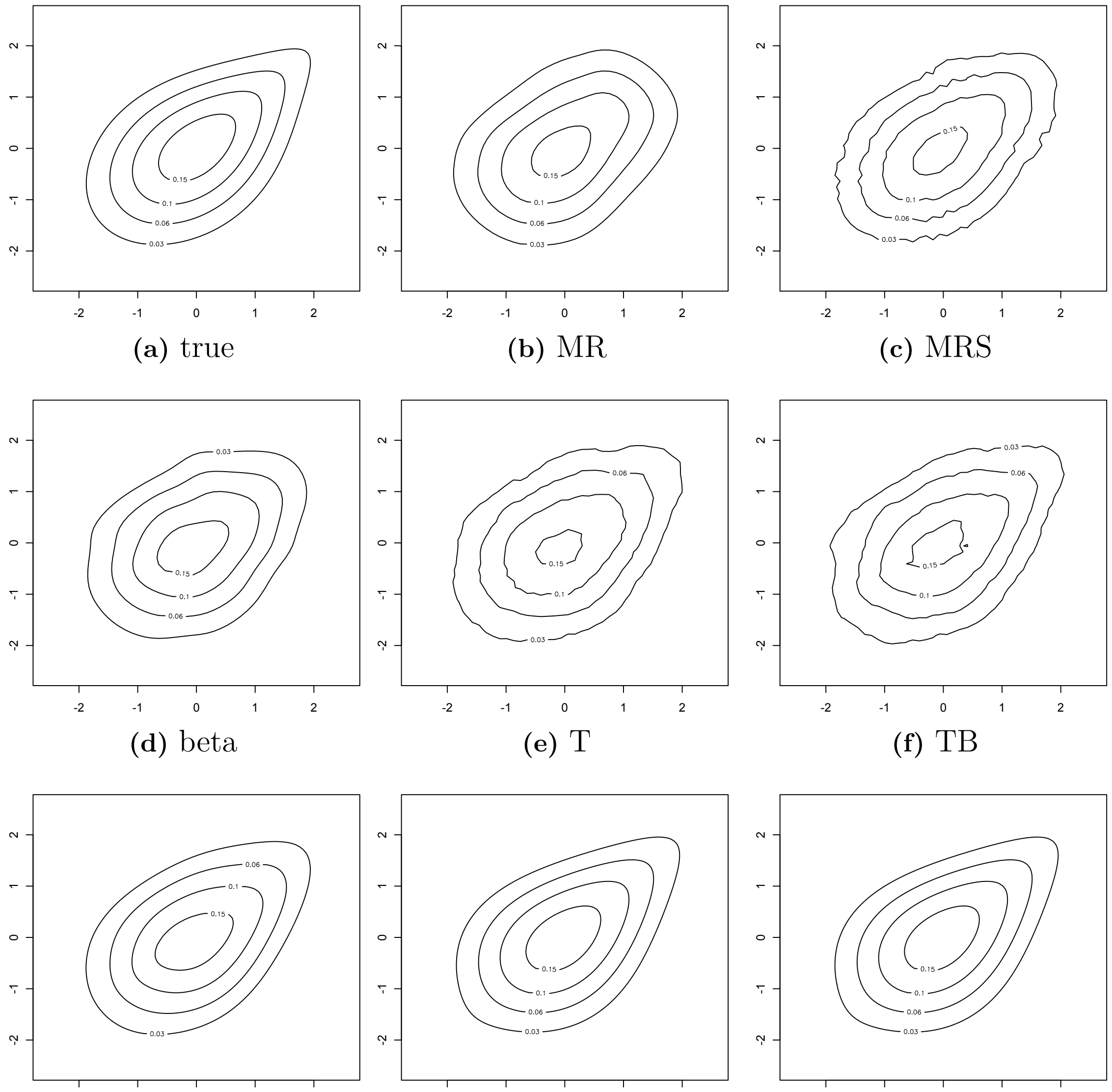
In simulations, our method was shown to be superior to the parametric approach in situations where some of the pair-copulas do not conform with any of the popular parametric families. Furthermore, it serves as a powerful tool for exploratory analysis of dependence in high-dimensional data. However, all that comes at the expense of increased computational demands. Lastly, the estimator's abilities were demonstrated with a real-data application in breast cancer research where we found evidence that both pairwise and joint dependence between selected variables contains information on the malignancy of cells.

Appendix A

Simulation study results

On the following pages we give more detailed results of the simulation study discussed in Section 3.6. Odd-numbered figures show marginal normal contour plots of the true density as well as exemplary estimates of this density for all considered methods. Even-numbered figures show boxplots of the estimators' performance measured by integrated squared error (ISE), integrated absolute error (IAE), and Hellinger distance (HD). Measures are reported for sample sizes $n = 250$ and $n = 1\,000$. For details on methodology see Section 3.6.1.

A.1 Gumbel copula



(a) true **(b)** MR **(c)** MRS

(d) beta **(e)** T **(f)** TB

(g) TLL **(h)** par- **(i)** par

Figure A.1: Gumbel copula, $\tau = 0.3$. Marginal normal contour plots of true density and estimates on an exemplary sample of size $n = 1\,000$.

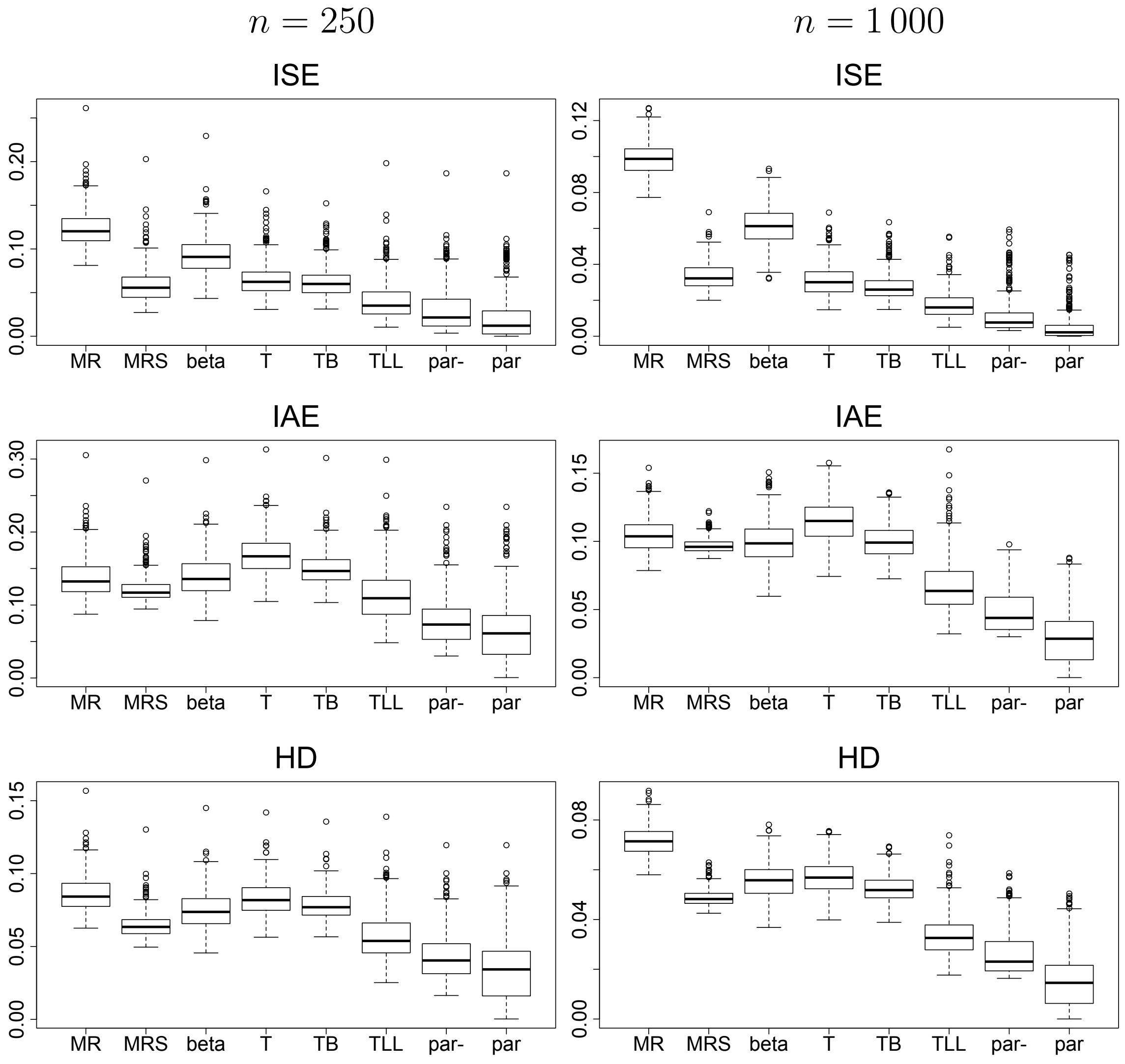
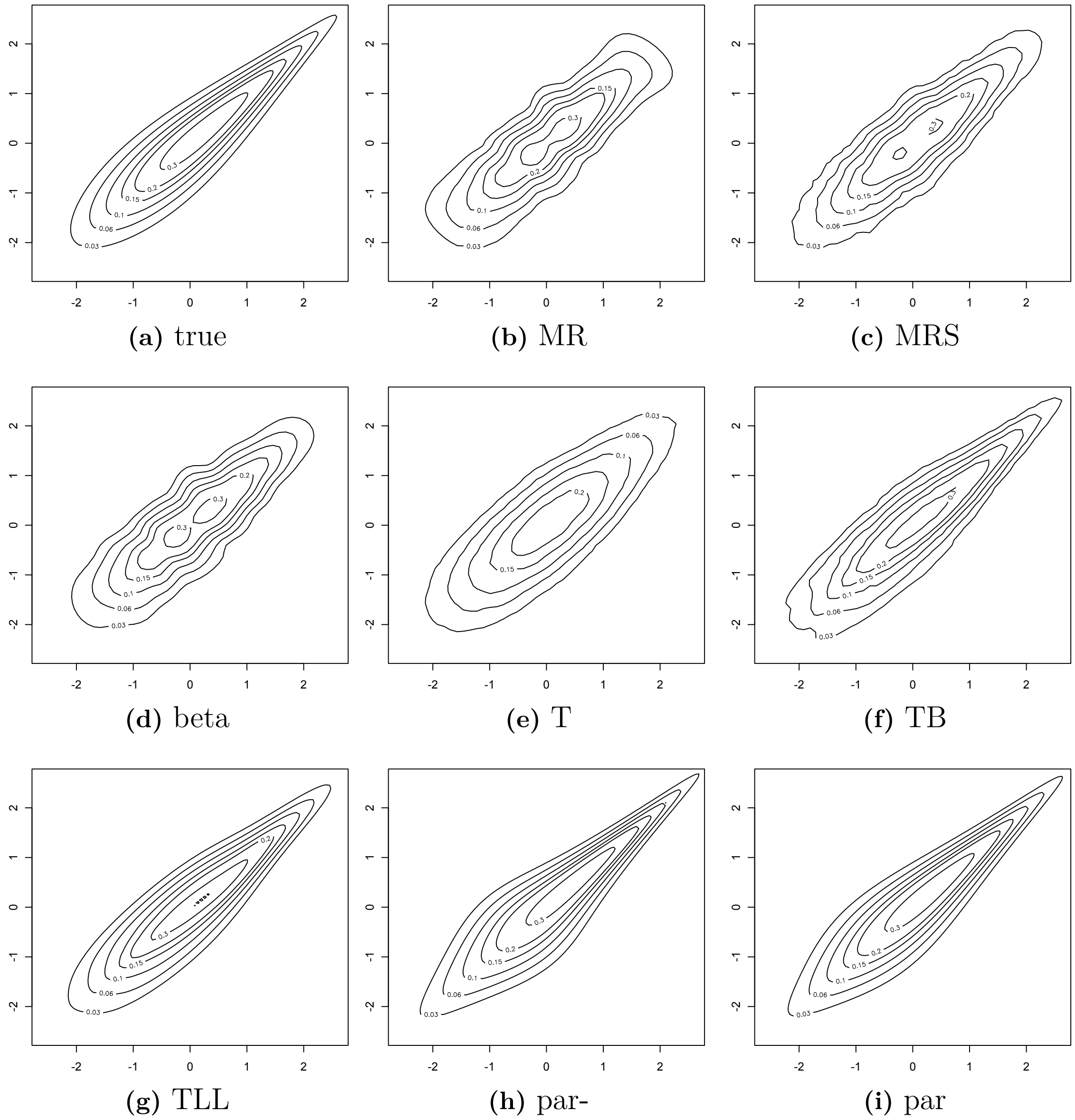


Figure A.2: Gumbel copula, $\tau = 0.3$. Boxplots of integrated squared error (ISE), integrated absolute error (IAE), and Helling distance (HD) for sample sizes $n = 250$ and $n = 1\,000$. For details on methodology see Section 3.6.1.



(a) true **(b)** MR **(c)** MRS

(d) beta **(e)** T **(f)** TB

(g) TLL **(h)** par- **(i)** par

Figure A.3: Gumbel copula, $\tau = 0.7$. Marginal normal contour plots of true density and estimates on an exemplary sample of size $n = 1\,000$.

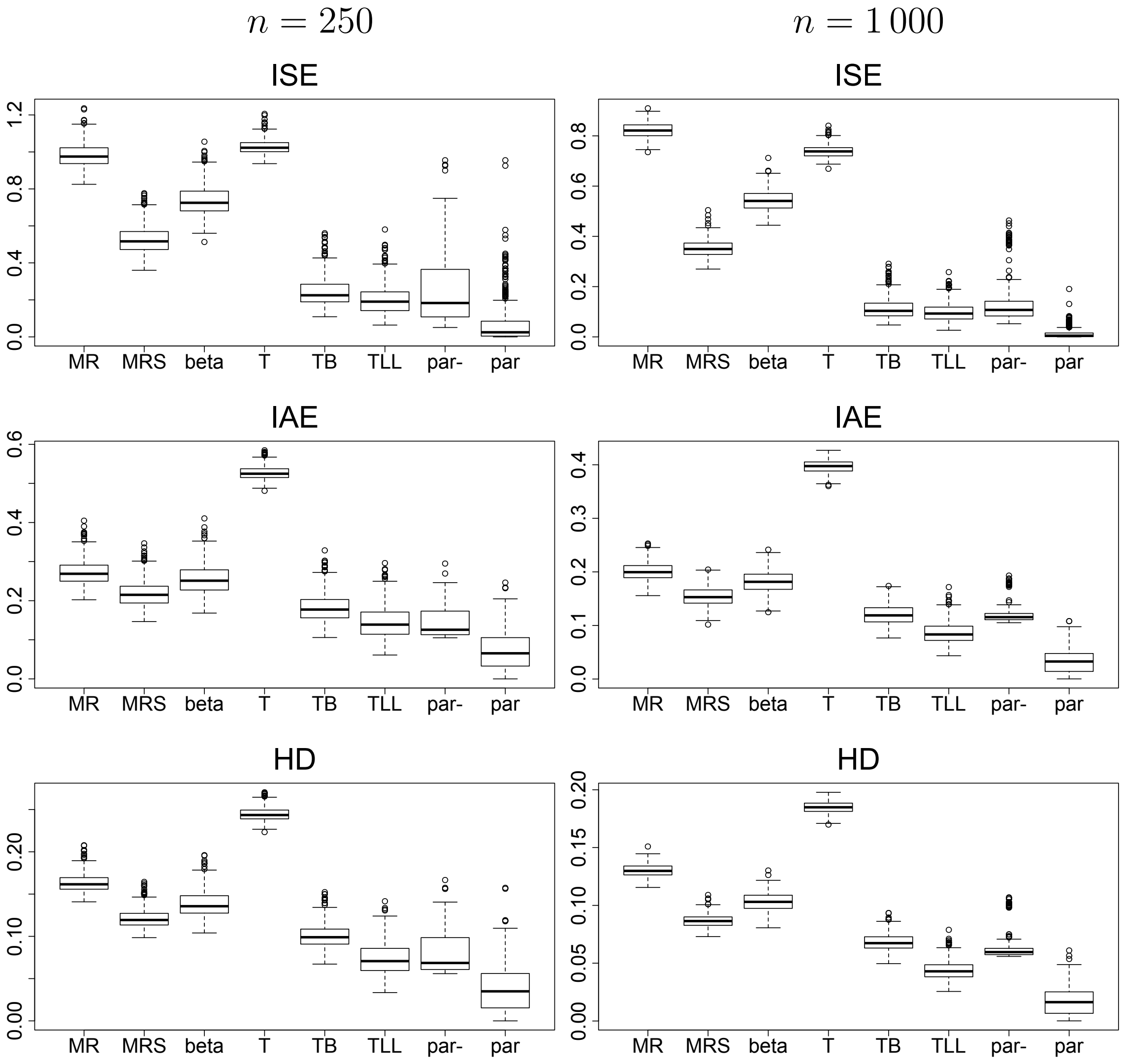
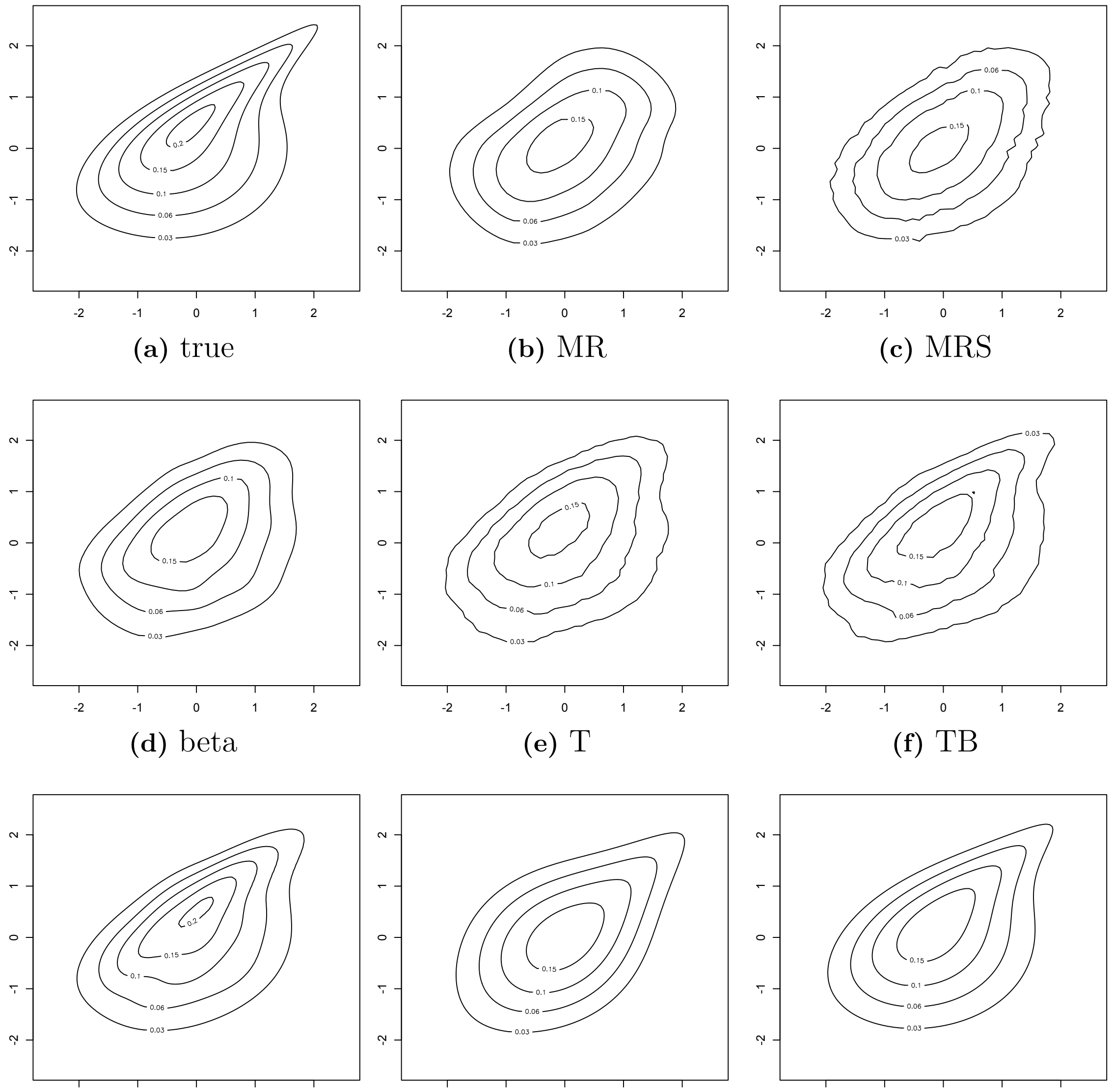


Figure A.4: Gumbel copula, $\tau = 0.7$. Boxplots of integrated squared error (ISE), integrated absolute error (IAE), and Helling distance (HD) for sample sizes $n = 250$ and $n = 1\,000$. For details on methodology see Section 3.6.1.

A.2 Tawn copula



(a) true

(b) MR

(c) MRS

(d) beta

(e) T

(f) TB

(g) TLL

(h) par-

(i) par

Figure A.5: Tawn copula, $\tau = 0.3$. Marginal normal contour plots of true density and estimates on an exemplary sample of size $n = 1\,000$.

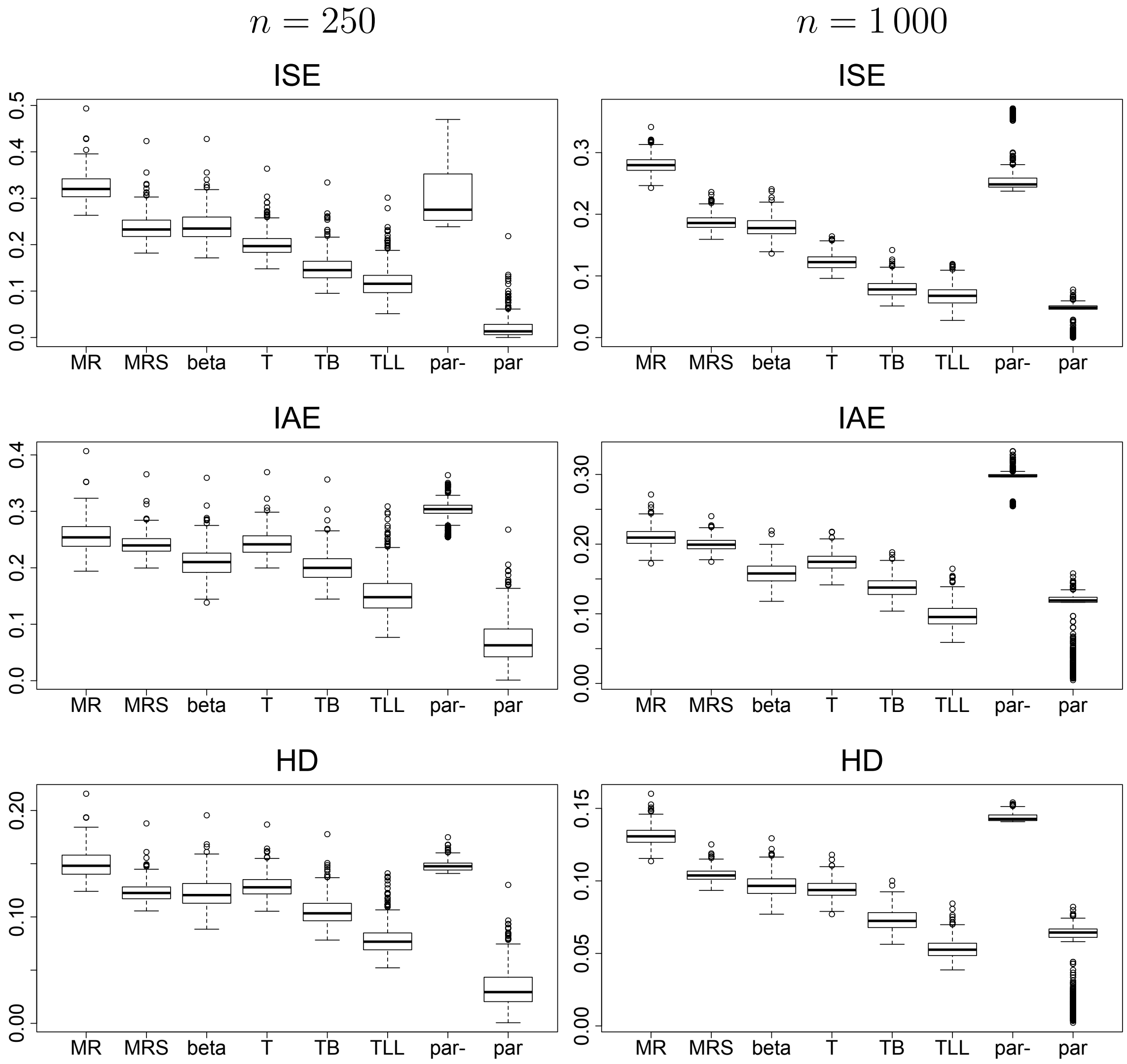
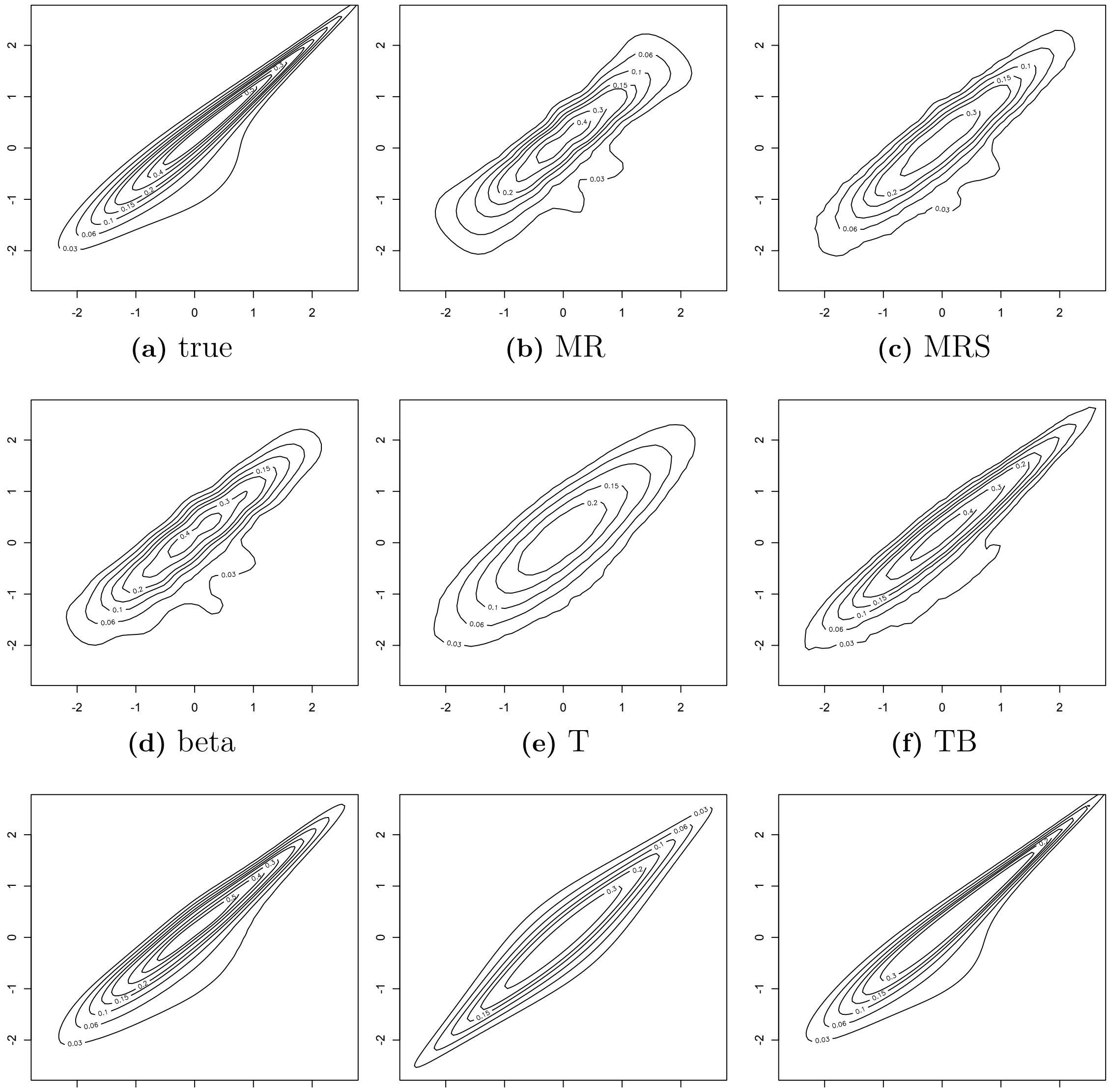


Figure A.6: Tawn copula, $\tau = 0.3$. Boxplots of integrated squared error (ISE), integrated absolute error (IAE), and Helling distance (HD) for sample sizes $n = 250$ and $n = 1\,000$. For details on methodology see Section 3.6.1.



(a) true

(b) MR

(c) MRS

(d) beta

(e) T

(f) TB

(g) TLL

(h) par-

(i) par

Figure A.7: Tawn copula, $\tau = 0.7$. Marginal normal contour plots of true density and estimates on an exemplary sample of size $n = 1\,000$.

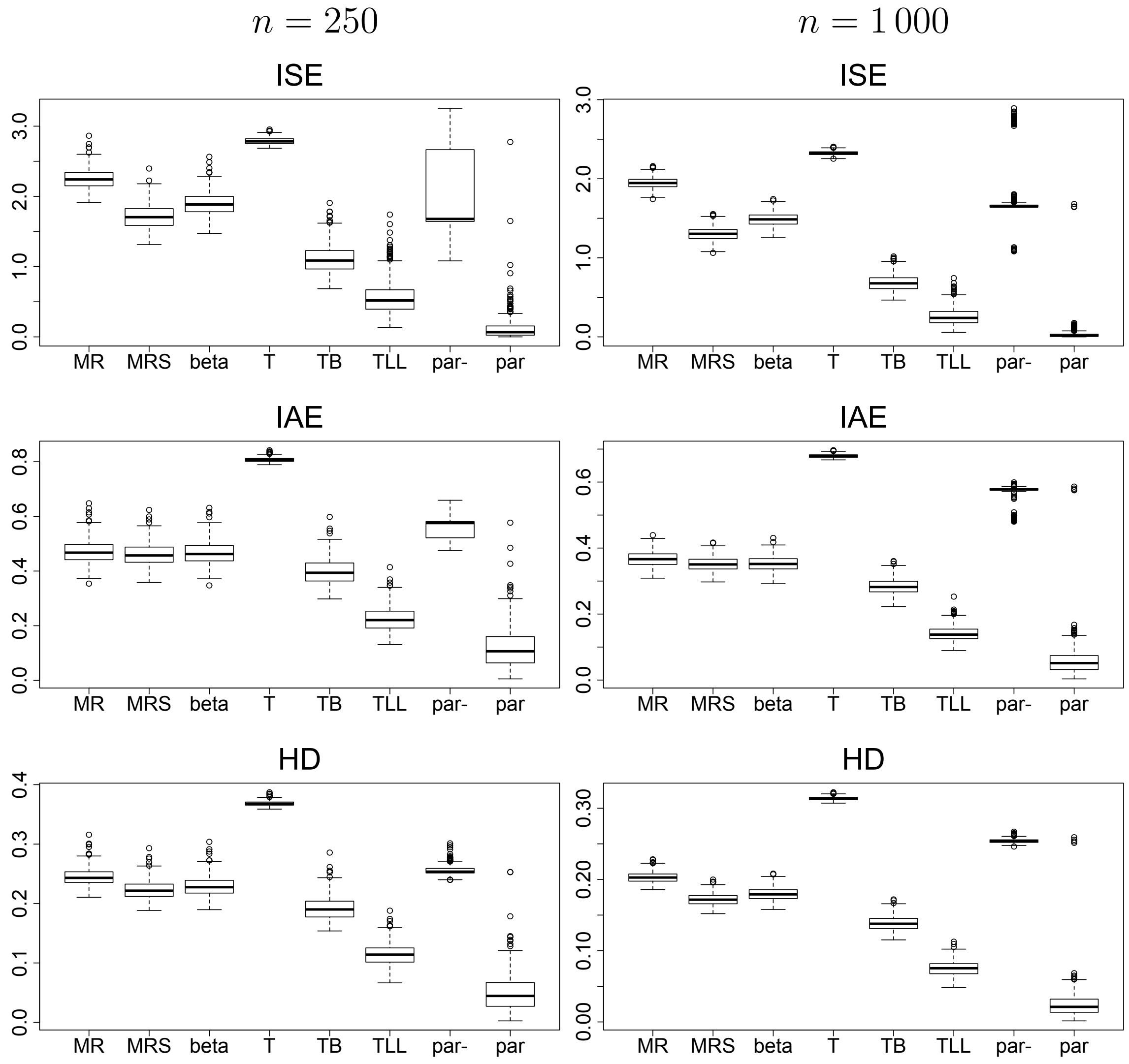
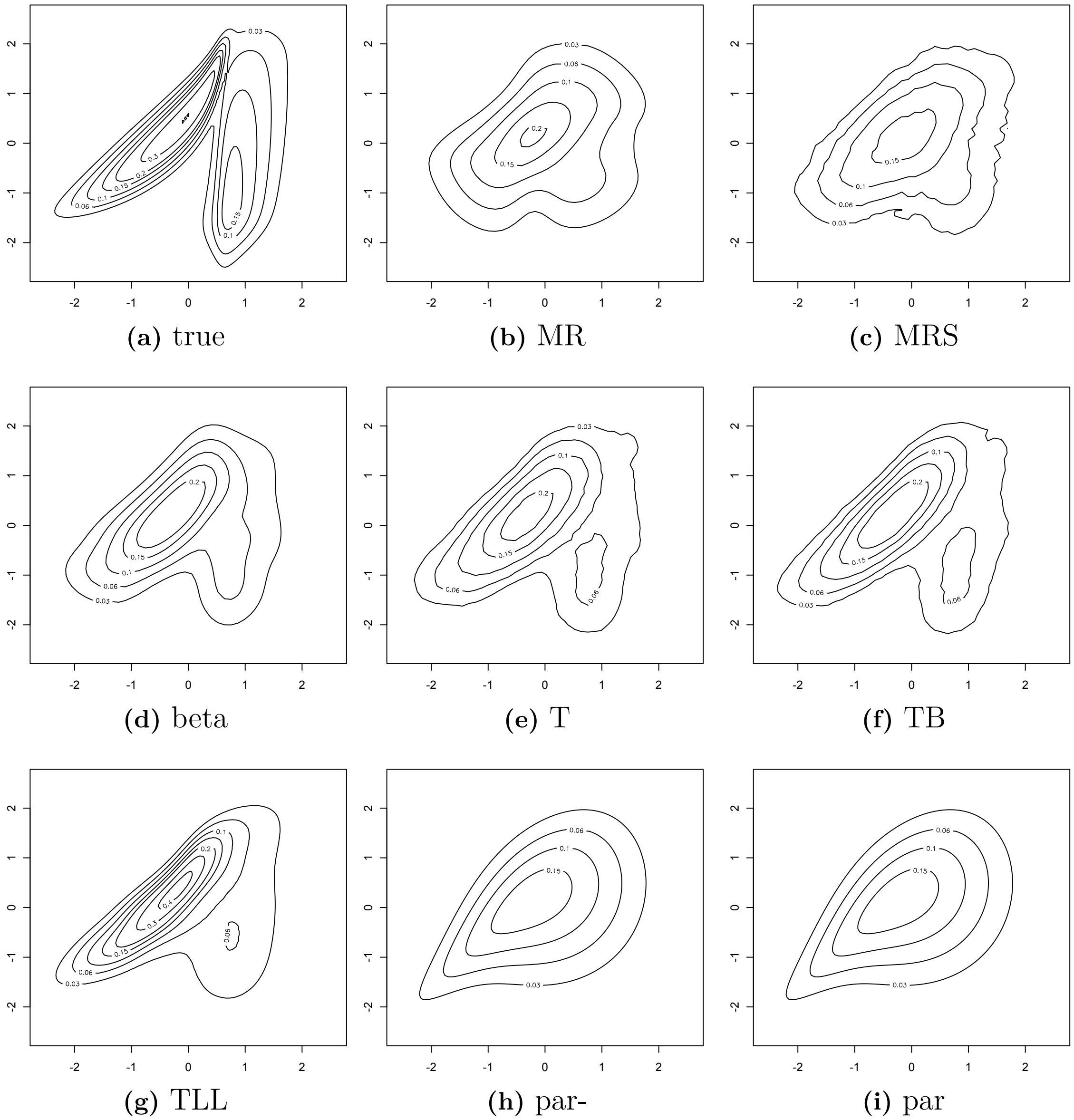


Figure A.8: Tawn copula, $\tau = 0.7$. Boxplots of integrated squared error (ISE), integrated absolute error (IAE), and Helling distance (HD) for sample sizes $n = 250$ and $n = 1\,000$. For details on methodology see Section 3.6.1.

A.3 Gaussian mixture copula



(a) true **(b)** MR **(c)** MRS

(d) beta **(e)** T **(f)** TB

(g) TLL **(h)** par- **(i)** par

Figure A.9: Gaussian mixture copula, $\tau = 0.3$. Marginal normal contour plots of true density and estimates on an exemplary sample of size $n = 1\,000$.

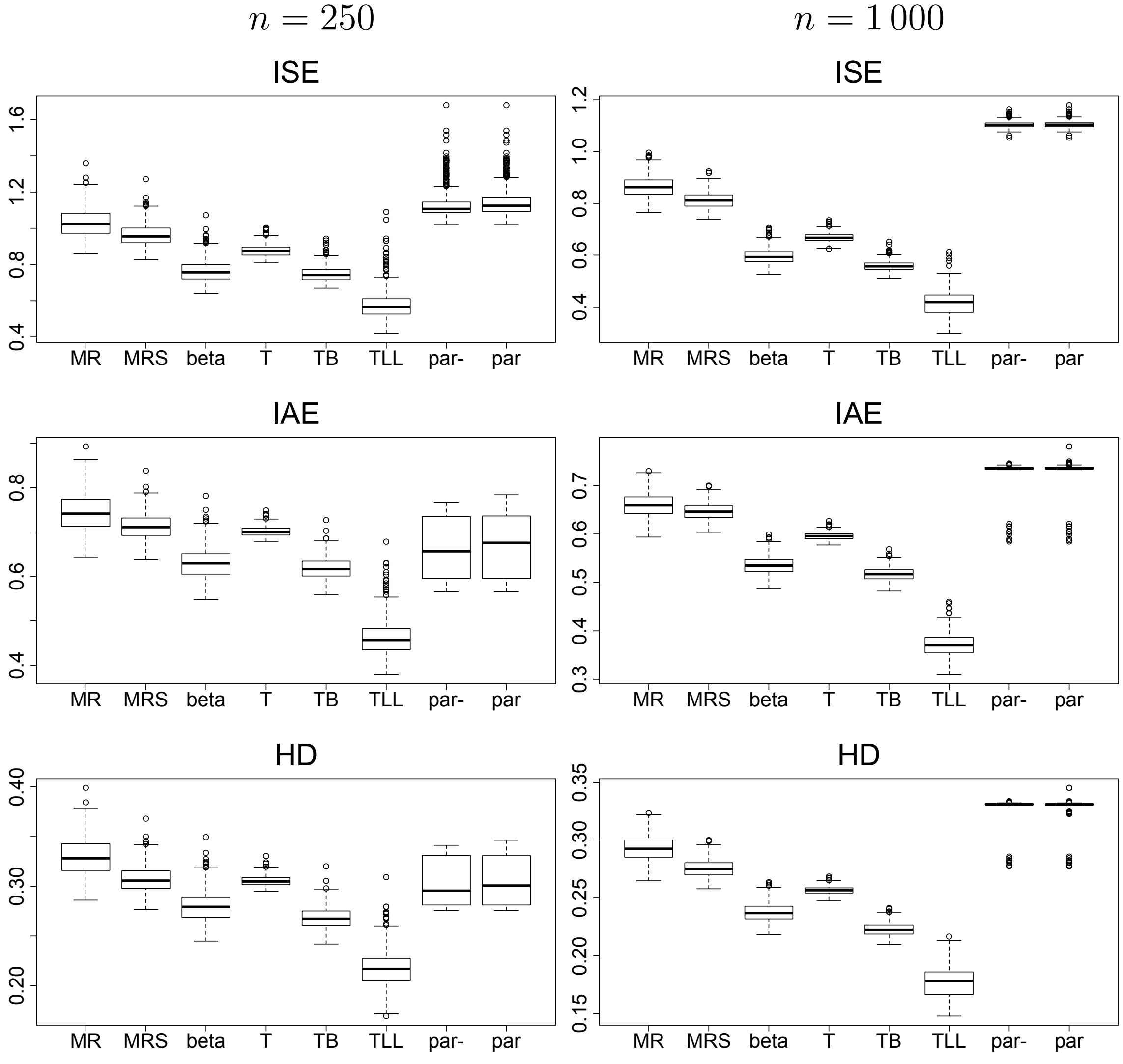
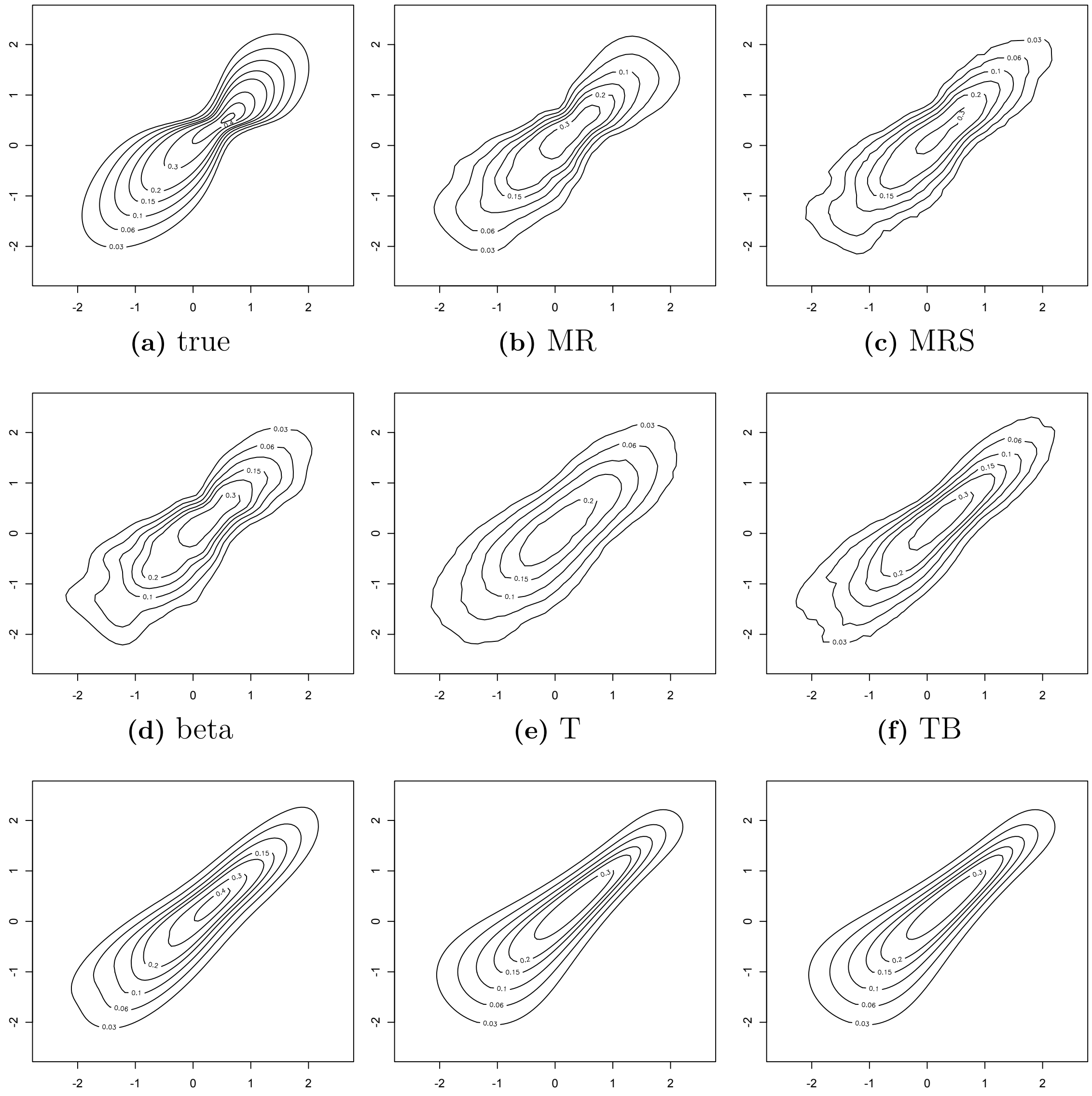


Figure A.10: Gaussian mixture copula, $\tau = 0.3$. Boxplots of integrated squared error (ISE), integrated absolute error (IAE), and Helling distance (HD) for sample sizes $n = 250$ and $n = 1\,000$. For details on methodology see Section 3.6.1.



(a) true **(b)** MR **(c)** MRS

(d) beta **(e)** T **(f)** TB

(g) TLL **(h)** par- **(i)** par

Figure A.11: Gaussian mixture copula, $\tau = 0.7$. Marginal normal contour plots of true density and estimates on an exemplary sample of size $n = 1\,000$.

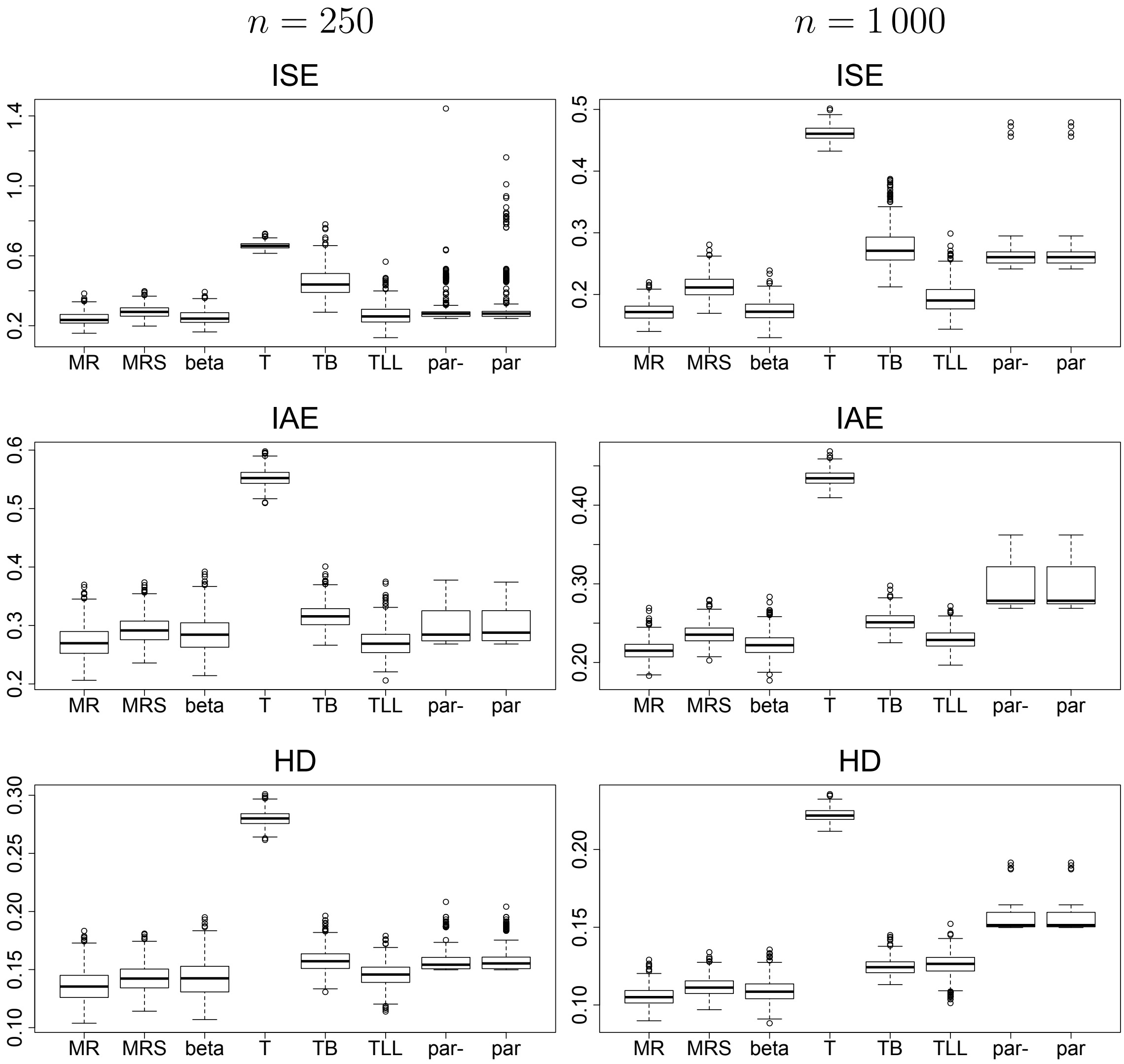


Figure A.12: Gaussian mixture copula, $\tau = 0.3$. Boxplots of integrated squared error (ISE), integrated absolute error (IAE), and Helling distance (HD) for sample sizes $n = 250$ and $n = 1\,000$. For details on methodology see Section 3.6.1.

Bibliography

Aas, K., Czado, C., Frigessi, A., and Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182–198.

Acar, E. F., Genest, C., and Nešlehová, J. (2012). Beyond simplified pair-copula constructions. *Journal of Multivariate Analysis*, 110(0):74 – 90. Special Issue on Copula Modeling and Dependence.

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723.

Bache, K. and Lichman, M. (2013). UCI machine learning repository, *url: http://archive.ics.uci.edu/ml.*

Bedford, T. and Cooke, R. (2001). Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial Intelligence*, 32(1-4):245–268.

Bedford, T. and Cooke, R. M. (2002). Vines–a new graphical model for dependent random variables. *The Annals of Statistics*, 30(4):1031–1068.

Blomqvist, N. (1950). On a measure of dependence between two random variables. *The Annals of Mathematical Statistics*, 21(4):593–600.

Brechmann, E. C., Czado, C., and Aas, K. (2012). Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics*, 40(1):68–85.

Charpentier, A., Fermanian, J.-D., and Scaillet, O. (2006). The estimation of copulas: Theory and practice. In Rank, J., editor, *Copulas: From theory to application in finance*. Risk Books.

Chen, S. X. (1999). Beta kernel estimators for density functions. *Computational Statistics & Data Analysis*, 31(2):131–145.

Chen, S. X. (2000). Beta kernel smoothers for regression curves. *Statistica Sinica*, 10(1):73–91.

Craven, P. and Wahba, G. (1978). Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403.

Czado, C., Jeske, S., and Hofmann, M. (2013). Selection strategies for regular vine copulae. *Journal of the French Statistical Society*, 154(1).

Devroye, L. and Györfi, L. (1985). *Nonparametric density estimation: The L_1 View.* John Wiley and Sons.

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(0):52 – 69.

Embrechts, P. (2009). Copulas: A personal view. *Journal of Risk & Insurance*, 76(3):639–650.

Embrechts, P., Lindskog, F., and Mcneil, A. (2003). Modelling Dependence with Copulas and Applications to Risk Management. In Rachev, S., editor, *Handbook of Heavy Tailed Distributions in Finance*, chapter 8, pages 329–384. Elsevier.

Epanechnikov, V. (1969). Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158.

Fan, J. and Gijbels, I. (1992). Variable bandwidth and local linear regression smoothers. *The Annals of Statistics*, 20(4):2008–2036.

Geenens, G., Charpentier, A., and Paindaveine, D. (2014). Probit transformation for nonparametric kernel estimation of the copula density. *arXiv:1404.4414 [stat.ME].*

Gijbels, I. and Mielniczuk, J. (1990). Estimating the density of a copula function. *Communications in Statistics - Theory and Methods*, 19(2):445–464.

Gijbels, I., Veraverbeke, N., and Omelka, M. (2011). Conditional copulas, association measures and their applications. *Comput. Stat. Data Anal.*, 55(5):1919–1932.

Gross, J. L. and Yellen, J. (2005). *Graph Theory and Its Applications.* Chapman & Hall/CRC.

Gudendorf, G. and Segers, J. (2009). Extreme-value copulas. *arXiv:0911.1015 [math.ST].*

Hall, P., Wolff, R. C. L., and Yao, Q. (1999). Methods for estimating a conditional distribution function. *Journal of the American Statistical Association*, 94(445):154–163.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Henderson, D. J. and Parmeter, C. F. (2011). Normal Reference Bandwidths for the General Order, Multivariate Kernel Density Derivative Estimator. Working Papers 2011-15, University of Miami, Department of Economics.

Hjort, N. L. and Jones, M. C. (1996). Locally parametric nonparametric density estimation. *The Annals of Statistics*, 24(4):1619–1647.

Hobæk Haff, I., Aas, K., and Frigessi, A. (2010). On the simplified pair-copula construction — simply useful or too simplistic? *Journal of Multivariate Analysis*, 101(5):1296 – 1310.

Hobæk Haff, I. and Segers, J. (2012). Nonparametric estimation of pair-copula constructions with the empirical pair-copula. *arXiv:1201.5133 [stat.ME]*.

Hurvich, C. M., Simonoff, J. S., and Tsai, C.-L. (1998). Smoothing parameter selection in nonparametric regression using an improved akaike information criterion. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(2):271–293.

Joe, H. (1997). *Multivariate models and dependence concepts*. Chapman & Hall, London.

Kendall, M. G. (1938). A New Measure of Rank Correlation. *Biometrika*, 30(1/2):81–93.

Kurowicka, D. and Joe, H., editors (2010). *DEPENDENCE MODELING:Vine Copula Handbook*. Number 7699 in World Scientific Books. World Scientific Publishing Co. Pte. Ltd.

Liu, W. and Yang, Y. (2012). Parametric or nonparametric? a parametricness index for model selection. *arXiv:1202.0391 [math.ST]*.

Loader, C. (1999). *Local regression and likelihood*. New York: Springer-Verlag.

Loader, C. (2013). *locfit: Local Regression, Likelihood and Density Estimation*. R package version 1.5-9.1.

Lopez-Paz, D., Hernández-Lobato, J. M., and Schölkopf, B. (2013). Semi-supervised domain adaptation with non-parametric copulas. *arXiv:1301.0142 [stat.ML]*.

Mack, Y. P. and Rosenblatt, M. (1979). Multivariate k-nearest neighbor density estimates. *Journal of Multivariate Analysis*, 9(1):1–15.

Nelsen, R. B. (2006). *An Introduction to Copulas (Springer Series in Statistics)*. Springer-Verlag New York, Inc.

Omelka, M., Gijbels, I., and Veraverbeke, N. (2009). Improved kernel estimation of copulas: Weak convergence and goodness-of-fit testing. *The Annals of Statistics*, 37(5B):3023–3058.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rizzo, M. L. (2008). *Statistical computing with R*. Computer science and data analysis series. Chapman & Hall/CRC.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837.

Rudemo, M. (1982). Empirical choice of histograms and kernel density estimators. *Scandinavian journal of statistics*.

Salvadori, G., Michele, C., Kottegoda, N., and Rosso, R. (2007). Multivariate extreme value theory. In *Extremes in Nature*, volume 56 of *Water Science and Technology Library*, pages 113–129. Springer Netherlands.

Schellhase, C. (2012). Density and copula estimation using penalized spline smoothing. *Dissertation, Universität Bielefeld*.

Schepsmeier, U., Stoeber, J., Brechmann, E. C., and Graeler, B. (2013). *VineCopula: Statistical inference of vine copulas*. R package version 1.2.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Scott, D. W. (2008). The curse of dimensionality and dimension reduction. In *Multivariate Density Estimation: Theory, Practice, and Visualization*, pages 195–217. John Wiley & Sons, Inc.

Scott, D. W. and Terrell, G. R. (1987). Biased and unbiased cross-validation in density estimation. *Journal of the American Statistical Association*, 82(400):1131–1146.

Segers, J. (2012). Asymptotics of empirical copula processes under non-restrictive smoothness assumptions. *Bernoulli*, 18(3):764–782.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall.

Simonoff, J. (1996). *Smoothing Methods in Statistics*. Springer Series in Statistics. Springer, New York, NY.

Sklar, A. (1959). *Fonctions de répartition à n dimensions et leurs marges*. Université Paris 8.

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):pp. 72–101.

Stewart, B. W. and Wild, C. P., editors (2014). *World Cancer Report 2014*. IARC Nonserial.

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

Stöber, J., Joe, H., and Czado, C. (2012). Simplified pair copula constructions — limits and extensions. *arXiv:1205.4844 [stat.ME].*

Wand, M. P. (1992). Error analysis for general multtvariate kernel estimators. *Journal of Nonparametric Statistics*, 2(1):1–15.

Wand, M. P. and Jones, M. C. (1993). Comparison of smoothing parameterizations in bivariate kernel density estimation. *Journal of the American Statistical Association*, 88(422):520–528.

Wand, M. P. and Jones, M. C. (1994). *Kernel Smoothing*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Chapman and Hall/CRC, 1 edition.

Weiß, G. and Scheffer, M. (2012). Smooth nonparametric bernstein vine copulas. *arXiv:1210.2043 [q-fin.RM].*

Wolberg, W. H., Street, W., and Mangasarian, O. (1994). Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates. *Cancer Letters*, 77(2–3):163 – 171.