

Archimedean Copulas in High Dimensions: Estimators and Numerical Challenges Motivated by Financial Applications

Titre: Implémentations stables et estimateurs de copules Archimédiennes en grandes dimensions pour les applications financières

Marius Hofert¹, Martin Mächler² and Alexander J. McNeil³

Abstract: The study of Archimedean dependence models in high dimensions is motivated by current practice in quantitative risk management. The performance of known and new parametric estimators for the parameters of Archimedean copulas is investigated and related numerical difficulties are addressed. In particular, method-of-moments-like estimators based on pairwise Kendall's tau, a multivariate extension of Blomqvist's beta, minimum distance estimators, the maximum-likelihood estimator, a simulated maximum-likelihood estimator, and a maximum-likelihood estimator based on the copula diagonal are studied. Their performance is compared in a large-scale simulation study both under known and unknown margins (pseudo-observations), in small and high dimensions, under small and large dependencies, and various different Archimedean families. High dimensions up to one hundred are considered and computational problems arising from such large dimensions are addressed in detail. All methods are implemented in the open source R package `copula` and can thus be easily accessed and studied. The numerical solutions developed in this work extend to various asymmetric generalizations of Archimedean copulas and important quantities such as distributions of radial parts or the Kendall distribution function.

Résumé : Les pratiques du moment en gestion quantitative du risque nous amènent à étudier des modèles de dépendance en grandes dimensions construits à partir de copules Archimédiennes. La performance d'un grand nombre d'estimateurs, dont plusieurs sont originaux, est étudiée et des solutions sont proposées pour les problèmes numériques associés. Des estimateurs fondés sur le méthode des moments et le beta de Blomqvist ainsi que le tau de Kendall sont comparés à des estimateurs du minimum de distance, du maximum de vraisemblance, du maximum de vraisemblance simulé et du maximum de vraisemblance fondé sur la diagonale de la copule. Des simulations sont réalisées dans le cas où les marges sont connues et dans le cas où elles sont estimées non paramétriquement, en faibles et en grandes dimensions, pour plusieurs familles de copules Archimédiennes. Toutes les méthodes étudiées sont implantées dans le package R `copula` distribué sous une licence libre. Les solutions numériques présentées dans ce travail restent valides dans le cas de généralisations asymétriques des copules Archimédiennes et d'importantes quantités associées comme la fonction de distribution de Kendall.

Keywords: Archimedean copulas, parameter estimation, Kendall's tau, Blomqvist's beta, minimum distance estimators, (diagonal/simulated) maximum-likelihood estimation, quantitative risk management

Mots-clés : Copules Archimédiennes, estimation paramétrique, tau de Kendall, beta de Blomqvist, estimateurs du minimum de distance, estimateurs du maximum de vraisemblance, gestion quantitative du risque

AMS 2000 subject classifications: 62H12, 62F10, 62H99, 62H20, 65C60

¹ RiskLab, Department of Mathematics, ETH Zurich, 8092 Zurich, Switzerland. E-mail: marius.hofert@math.ethz.ch. The author (Willis Research Fellow) thanks Willis Re for financial support while this work was being completed.

² Seminar für Statistik, ETH Zurich, 8092 Zurich, Switzerland. E-mail: maechler@stat.math.ethz.ch.

³ Department of Actuarial Mathematics and Statistics, Heriot-Watt University, Edinburgh, EH14 4AS, Scotland, and Maxwell Institute for Mathematical Sciences. E-mail: A.J.McNeil@hw.ac.uk.

1. Introduction

A *copula* is a multivariate distribution function with standard uniform univariate margins. An important class of copulas, known as *Archimedean copulas*, is given by

$$C(\mathbf{u}) = \psi(\psi^{-1}(u_1) + \dots + \psi^{-1}(u_d)), \quad \mathbf{u} \in [0, 1]^d,$$

with *generator* ψ . As a consequence of this functional form, Archimedean copulas are exchangeable, that is, unaffected by permutations of u_1, \dots, u_d . In practical applications, ψ belongs to a parametric family $(\psi_{\boldsymbol{\theta}})_{\boldsymbol{\theta} \in \Theta}$ whose parameter vector $\boldsymbol{\theta}$ needs to be estimated; for most popular parametric Archimedean copula families, $\boldsymbol{\theta}$ is one-dimensional. The aims of this paper are two-fold:

1. To carry out a large-scale comparative study of many estimators for Archimedean copulas (including new ones), both under known and unknown margins (pseudo-observations);
2. To focus on the performance of estimators and computations in high dimensions, where considerable numerical challenges have to be overcome.

1.1. Motivation and examples

In high dimensions, none of the existing copula models typically fits data well. Nevertheless, in areas such as the modeling of financial and insurance risks, copula models are applied in high dimensions, often, in our experience, in dimensions that run in to the thousands. As an example, consider a portfolio of d financial instruments or assets with values $S_{t,1}, \dots, S_{t,d}$ at time point t , interpreted as today. These asset values are referred to as *risk factors* and the value of the portfolio is

$$V_t = \sum_{j=1}^d \beta_j S_{t,j},$$

where β_j denotes the number of units of asset j in the portfolio, $j \in \{1, \dots, d\}$. The asset log-returns are given by

$$X_{t+1,j} = \log(S_{t+1,j}) - \log(S_{t,j}) = \log(S_{t+1,j}/S_{t,j}), \quad j \in \{1, \dots, d\},$$

and the one-period ahead (sign adjusted) loss L of the portfolio is therefore

$$L = -(V_{t+1} - V_t) = -\sum_{j=1}^d \beta_j (S_{t+1,j} - S_{t,j}) = -\sum_{j=1}^d \beta_j S_{t,j} (\exp(X_{t+1,j}) - 1). \quad (1)$$

From a risk management perspective in the context of Basel II/III one is interested in the distribution of the loss L . As becomes clear from (1), the loss distribution only depends on the joint, and typically high-dimensional, distribution of the log-returns, all other quantities being known at time point t , that is, today.

Financial portfolios can consist of complex instruments, for example, options, futures, credit default swaps and insurance policies. In such cases, the formalism in (1) has to be extended to

incorporate changes in other risk factors, such as price volatilities, default indicators and mortality rates, for which less information is typically available than for simple asset prices. For computing risk measures such as Value-at-Risk or expected shortfall, modeling dependence between the risk-factor changes is crucial, as various examples in [McNeil et al., 2005] show.

Dimension reduction techniques cannot always be used in this context since it may be important to model future payments from each contract or policy. In the absence of more detailed information, exchangeability of the dependence structure is often assumed which makes Archimedean copulas candidate models.

The reason for this is not that these models provide a good fit to the data at hand. Rather, they are

1. comparably tractable from a numerical and computational point of view in high dimensions;
2. more plausible than models that assume independence and capable of giving indicative results on the impact of dependence.

Model misspecification is often accepted in these applications in return for this convenience.

Another area of applications where exchangeable dependence models are used is sparse data. This includes applications to operational risk modeling; see [Chavez-Demoulin et al., 2013] and references therein. If the tail is of major interest, Archimedean copulas such as the Clayton or Gumbel copula are often considered. They provide alternatives to the t copula (note: the Gaussian copula does not allow for tail dependence, see [Sibuya, 1959]) to model tail dependence and are also used for stress testing. A correct specification of the body of the distribution is often not too important. One such example is [Schönbucher and Schubert, 2001] and [Hofert and Scherer, 2011] in the context of high-dimensional credit-risk portfolios, where copulas are calibrated to essentially five data points only. Nevertheless, the dependence between the underlying 125 components in the portfolio plays a crucial role in pricing these products, which has also been learned from the recent crisis; see, for example, [Donnelly and Embrechts, 2010]. As shown in [Hofert and Scherer, 2011], even a highly exchangeable model with only one or two parameters can provide a much more accurate pricing model for such portfolio products. More parameters may not be justifiable in such applications given the small amount of data available and also not computationally feasible since each evaluation of the object function is based on a Monte Carlo simulation. From a practical point of view, one-parameter models are often easy to interpret due to a one-to-one correspondence with measures of association such as Kendall's tau. This has contributed to their popularity; see [Embrechts and Hofert, 2011] for a discussion.

Other examples not detailed here can be found in the area of survival analysis, see [Hougaard, 2000] and [Duchateau and Janssen, 2008].

Because they are used in practice in high dimensions, it is important to investigate Archimedean copulas in that context. Our paper [Hofert et al., 2012] was a first theoretical step in this direction. The present paper complements this work in a statistical and computational direction, detailed below. The results extend beyond Archimedean copulas and should be viewed in a wider context.

As the following examples show, the need for accurately evaluating (or estimating) the derivatives of generators of Archimedean copulas arises in a variety of contexts.

- The density of a (possibly asymmetric, that is, non-exchangeable) Khoudraji-transformed

Archimedean copula is given by

$$c(\mathbf{u}) = \sum_{J \subseteq \{1, \dots, d\}} \psi^{(|J|)} \left(\sum_{j=1}^d \psi^{-1}(u_j^{\alpha_j}) \right) \prod_{j \in J} \alpha_j (\psi^{-1})'(u_j^{\alpha_j}) \prod_{j \notin J} (1 - \alpha_j) u_j^{-\alpha_j},$$

see [Hofert et al., 2012].

- [Hofert, 2010, pp. 117] presents ways to construct Archimedean copulas with more than one parameter. One example are outer power Archimedean copulas (with generator $\tilde{\psi}(t) = \psi(t^{1/\beta})$, $\beta \geq 1$). As Archimedean copulas, their density involves the generator derivatives, given by

$$(-1)^d \tilde{\psi}^{(d)}(t) = P^{\text{op}}(t^{1/\beta})/t^d, \quad d \in \mathbb{N}, \quad \text{where} \quad P^{\text{op}}(x) = \sum_{k=1}^d a_{dk}^G(1/\beta) (-1)^k \psi^{(k)}(x) x^k.$$

For an extension of this result, see [Hofert and Pham, 2013]. In this reference, densities of nested Archimedean copulas are derived. They again heavily involve generator derivatives.

- As a special case of the construction of [Hofert and Vriens, 2013], Archimedean Sibuya copulas arise. As a tractable subclass, asymmetric extensions of Archimedean copulas of the form

$$C(\mathbf{u}) = \frac{\psi_V(\sum_{j=1}^d \psi_{\mu_j+V}^{-1}(u_j))}{\prod_{j=1}^d \psi_V(\psi_{\mu_j+V}^{-1}(u_j))} \Pi(\mathbf{u}),$$

are presented, for a Laplace-Stieltjes transform ψ_V of a positive random variable V and corresponding Laplace-Stieltjes transforms ψ_{μ_j+V} of the random variables $\mu_j + V$ for deterministic μ_1, \dots, μ_d . Their densities are given by

$$c(\mathbf{u}) = \prod_{j=1}^d \frac{1}{\psi_V(\psi_{\mu_j+V}^{-1}(u_j))} \sum_{J \subseteq \{1, \dots, d\}} \psi_V^{(|J|)} \left(\sum_{j=1}^d \psi_{\mu_j+V}^{-1}(u_j) \right) \cdot \prod_{j \in J} \frac{u_j}{\psi_{\mu_j+V}'(\psi_{\mu_j+V}^{-1}(u_j))} \prod_{j \notin J} \left(1 - \frac{u_j \psi_V'(\psi_{\mu_j+V}^{-1}(u_j))}{\psi_{\mu_j+V}'(\psi_{\mu_j+V}^{-1}(u_j)) \psi_V(\psi_{\mu_j+V}^{-1}(u_j))} \right),$$

thus again involve generator derivatives.

- High-order generator derivatives also appear in several statistically important quantities such as
- the distribution function of the radial part

$$F_R(t) = 1 - \sum_{k=0}^{d-1} \frac{(-1)^k \psi^{(k)}(t)}{k!} t^k, \quad t \in [0, \infty),$$

(here: a slightly simplified version), see [McNeil and Nešlehová, 2009];

- the Kendall distribution function

$$K(t) = \sum_{k=0}^{d-1} (-\psi^{-1}(t))^k \psi^{(k)}(\psi^{-1}(t))/k!, \quad t \in [0, 1],$$

(again a slightly simplified version here), see [Barbe et al., 1996] or [Genest et al., 2011].

For statistical applications, it is therefore crucial to be able to accurately compute (or estimate) densities and generator derivatives of Archimedean copulas. A considerable amount of time has gone into numerically stable and efficient implementation in the R package `copula`. The simulation results reported in this paper also serve as checks on the implementations in the package. Moreover, the numerical tricks and results presented also apply in the other contexts mentioned above.

We believe that issues of this type will become more important in the future as copula models in higher dimensions gain in applied and theoretical interest. Our computations will also point out interesting (and partly surprising) results, which might lead to further research in this direction.

1.2. Overview and organization of the paper

There are several known approaches for estimating bivariate parametric Archimedean copula families. Assuming the copula density to exist, maximum-likelihood estimation is one option; see [Genest et al., 1995] or [Tsukahara, 2005]. Another estimator resembles the method-of-moments estimator and consists of choosing the copula parameter such that a certain dependence measure, for example, Kendall's tau, equals its empirical counterpart; see [Genest and Rivest, 1993]. Although there is no theoretical justification for applying this method in more than two dimensions, using the mean of pairwise empirical Kendall's taus and estimating the copula parameter such that the population version of Kendall's tau equals this mean also appears in the literature; see [Berg, 2009] or [Savu and Trede, 2010]. A similar but different estimator is applied in [Kojadinovic and Yan, 2010]. Another method in higher dimensions based on the moments of the Kendall distribution function is given in [Brahimi and Necir, 2011]. Other estimation methods include approximating the probability integral transform with splines and using a minimum distance approach between this distribution function and an empirical counterpart; see [Dimitrova et al., 2008]. Splines also appear in [Lambert, 2007] for approximating a certain ratio involving the generator of the Archimedean copula to be estimated. [Tsukahara, 2005] considers minimum distance estimators based on Cramér-von Mises or Kolmogorov-Smirnov distances and compares their performance to rank approximate Z-estimators in a simulation study involving the bivariate Archimedean Clayton, Frank, and Gumbel copula. Another estimation procedure in the bivariate case is given by [Qu et al., 2010] based on minimizing a Cramér-von Mises distance between the empirical distribution function of a certain univariate random sample and the standard uniform distribution. The approach described in [Stephenson, 2009] in the context of extreme-value distributions can be applied for estimating the parameter of a Gumbel copula in a Bayesian setup. A non-parametric estimation procedure is introduced in [Genest et al., 2011]. For more general information concerning copula parameter or copula density estimation in parametric and (especially) non-parametric set-ups, see [Charpentier et al., 2007].

In this work, we compare several known and new parametric estimators for Archimedean copulas both under known and unknown margins (the margins being estimated non-parametrically). In the large-scale simulation study carried out, we compare the following estimators based on well-known one-parameter generators (for two-parameter families, see [Hofert et al., 2012]):

1. We consider the method-of-moments estimator based on averaged pairwise sample versions of Kendall's tau. We also consider the average of pairwise Kendall's tau estimators.

2. We apply a multivariate version of the measure of concordance known as Blomqvist's beta for estimating Archimedean copulas. Blomqvist's beta has the advantage of being given explicitly in terms of the copula. Similar to the method-of-moments estimation procedure introduced by [Genest and Rivest, 1993], the copula parameters are estimated such that the population and sample version of Blomqvist's beta coincide.
3. We present several minimum distance estimators for estimating Archimedean copulas. Recently, a transformation of random variables following an Archimedean copula to uniform random variables (similar to Rosenblatt's transformation but simpler to compute) was introduced by [Hering and Hofert, 2013]. The minimum distance estimators presented here estimate the parameters as the minimum of certain Cramér-von Mises or Kolmogorov-Smirnov distances based on the transformation of [Hering and Hofert, 2013].
4. We consider maximum-likelihood estimation. Although the density of an Archimedean copula has an explicit form in theory, deriving and evaluating the required derivatives is known to be challenging from both a theoretical and a numerical perspective, especially in large dimensions; see also our motivation in Section 1.1. As mentioned below, computations based on computer algebra systems often fail already in low dimensions or require high precision (and are therefore too slow to be applied, for example, in large-scale simulation studies). We present explicit formulas for the densities of well-known Archimedean families and efficiently evaluate them. These results are based on the recent findings of [Hofert et al., 2012].
5. We introduce a simulated maximum-likelihood estimator to estimate Archimedean copulas. This estimator can be applied if the generator derivatives cannot be evaluated accurately but the copula is easy to sample.
6. We present maximum-likelihood estimation based on the diagonal of the Archimedean copula. The main advantage is that the resulting estimation method is comparably easy and fast to apply in virtually any dimension.

The paper is organized as follows. In Section 2, we briefly recall the notion of Archimedean copula. Section 3 introduces and presents the different estimators investigated in this work. Section 4 contains the large-scale simulation carried out. Section 5 addresses numerical issues when working in large dimensions and provides solutions to some of the problems mentioned. Section 6 concludes.

2. Archimedean copulas

Definition 2.1

An (*Archimedean*) *generator* is a continuous, decreasing function $\psi : [0, \infty] \rightarrow [0, 1]$ which satisfies $\psi(0) = 1$, $\psi(\infty) = \lim_{t \rightarrow \infty} \psi(t) = 0$, and which is strictly decreasing on $[0, \inf\{t : \psi(t) = 0\}]$. A d -dimensional copula C is called *Archimedean* if it permits the representation

$$C(\mathbf{u}) = \psi(t(\mathbf{u})), \quad \text{where} \quad t(\mathbf{u}) = \sum_{j=1}^d \psi^{-1}(u_j), \quad \mathbf{u} \in [0, 1]^d, \quad (2)$$

for some generator ψ with inverse $\psi^{-1} : [0, 1] \rightarrow [0, \infty]$, where $\psi^{-1}(0) = \inf\{t : \psi(t) = 0\}$.

[McNeil and Nešlehová, 2009] show that a generator defines an Archimedean copula of any dimension $k \in \{2, \dots, d\}$ if and only if ψ is d -monotone, that is, ψ is continuous on $[0, \infty]$, admits derivatives up to the order $d - 2$ satisfying $(-1)^k \frac{d^k}{dt^k} \psi(t) \geq 0$ for all $k \in \{0, \dots, d - 2\}$, $t \in (0, \infty)$, and $(-1)^{d-2} \frac{d^{d-2}}{dt^{d-2}} \psi(t)$ is decreasing and convex on $(0, \infty)$. We mainly assume ψ to be *completely monotone*, meaning that ψ is continuous on $[0, \infty]$ and $(-1)^k \frac{d^k}{dt^k} \psi(t) \geq 0$ for all $k \in \mathbb{N}_0$, $t \in (0, \infty)$, so that ψ is the Laplace-Stieltjes transform $\mathcal{L}\mathcal{S}[F]$ of a distribution function F on the positive real line; see Bernstein's Theorem in [Feller, 1971, p. 439]. The class of all such generators is denoted by Ψ_∞ and it is clear that a $\psi \in \Psi_\infty$ generates an Archimedean copula in any dimensions d .

There are several known parametric Archimedean generators (see, for example, [Nelsen, 2006, pp. 116]) also referred to as *Archimedean families*. Among the most widely used in applications are those of Ali-Mikhail-Haq (A), Clayton (C), Frank (F), Gumbel (G), and Joe (J). We will consider these generators as working examples; see Table 1 which also includes population versions of Kendall's tau for these families. Here, $D_1(\theta) = \int_0^\theta t / (\exp(t) - 1) dt / \theta$ denotes the *Debye function of order one*. Detailed information about the distribution functions F corresponding to the given generators can be found in [Hofert, 2012] and references therein.

TABLE 1. Well-known one-parameter Archimedean generators ψ with corresponding Kendall's tau. The range of attainable Kendall's tau is $(0, 1/3)$ for A, $(0, 1)$ for C and F, and $[0, 1)$ for G and J.

Family	Parameter	$\psi(t)$	τ
A	$\theta \in [0, 1)$	$(1 - \theta) / (\exp(t) - \theta)$	$1 - 2(\theta + (1 - \theta)^2 \log(1 - \theta)) / (3\theta^2)$
C	$\theta \in (0, \infty)$	$(1 + t)^{-1/\theta}$	$\theta / (\theta + 2)$
F	$\theta \in (0, \infty)$	$-\log(1 - (1 - e^{-\theta}) \exp(-t)) / \theta$	$1 + 4(D_1(\theta) - 1) / \theta$
G	$\theta \in [1, \infty)$	$\exp(-t^{1/\theta})$	$(\theta - 1) / \theta$
J	$\theta \in [1, \infty)$	$1 - (1 - \exp(-t))^{1/\theta}$	$1 - 4 \sum_{k=1}^{\infty} 1 / (k(\theta k + 2)(\theta(k - 1) + 2))$

3. Estimation methods for Archimedean copulas

Assume that we have given realizations $\mathbf{x}_i, i \in \{1, \dots, n\}$, of independent and identically distributed (iid) random vectors $\mathbf{X}_i, i \in \{1, \dots, n\}$, from a joint distribution function H with known margins $F_j, j \in \{1, \dots, d\}$, Archimedean copula C generated by ψ , and corresponding density c . The generator ψ is assumed to belong to a parametric family $(\psi_\theta)_{\theta \in \Theta}$ with parameter vector $\theta \in \Theta \subseteq \mathbb{R}^p$, $p \in \mathbb{N}$, and the true but unknown vector is θ_0 (similarly, $C = C_{\theta_0}$ and $c = c_{\theta_0}$). If the margins $F_j, j \in \{1, \dots, d\}$, are known, $u_{ij} = F_j(x_{ij}), i \in \{1, \dots, n\}, j \in \{1, \dots, d\}$, is a random sample from C . In practice, the margins are typically unknown and must be estimated parameterically or non-parameterically. In the following, whenever working under unknown margins, we will assume the latter approach and thus consider the *pseudo-observations*

$$\hat{u}_{ij} = \frac{n}{n+1} \hat{F}_{n,j}(x_{ij}) = \frac{r_{ij}}{n+1}, \quad (3)$$

where $\hat{F}_{n,j}$ denotes the *empirical distribution function* corresponding to the j th margin and r_{ij} denotes the *rank* of x_{ij} among all $x_{ij}, i \in \{1, \dots, n\}$.

For estimating θ_0 , we now present several methods, some of which (such as the simulated or the diagonal maximum-likelihood estimator) are new. We give the formulas in terms of a

random sample $\mathbf{U}_i, i \in \{1, \dots, n\}$, from C . In Section 4, this random sample is replaced either by realizations $\mathbf{u}_i, i \in \{1, \dots, n\}$ (when working under known margins) or by the pseudo-observations $\hat{\mathbf{u}}_i, i \in \{1, \dots, n\}$, when working under unknown margins.

3.1. Pairwise Kendall's tau

Kendall's tau is defined to be

$$\tau = \mathbb{E}[\text{sign}((X_1 - X'_1)(X_2 - X'_2))],$$

where $(X_1, X_2)^\top$ is a vector of two continuously distributed random variables, $(X'_1, X'_2)^\top$ is an independent copy of $(X_1, X_2)^\top$, and $\text{sign}(x) = \mathbb{1}_{(0, \infty)}(x) - \mathbb{1}_{(-\infty, 0)}(x)$ denotes the *signum function*. Kendall's tau is a measure of concordance (see [Scarsini, 1984]) and therefore measures the strength of association (as a number in $[-1, 1]$) between large values of one variable and large values of the other. Note that Archimedean copulas with generator $\psi \in \Psi_\infty$ are positive lower orthant dependent, thus Kendall's tau always lies in $[0, 1]$ for such copulas; see, for example, [Hofert, 2010, pp. 59]. Kendall's tau has an obvious estimator, referred to as the *sample version of Kendall's tau*. Based on the random sample $\mathbf{U}_i = (U_{i1}, U_{i2})^\top, i \in \{1, \dots, n\}$, it is given by

$$\hat{\tau}_n = \binom{n}{2}^{-1} \sum_{1 \leq i_1 < i_2 \leq n} \text{sign}((U_{i_1 1} - U_{i_2 1})(U_{i_1 2} - U_{i_2 2})).$$

It can also be estimated directly from the bivariate sample $\mathbf{X}_i, i \in \{1, \dots, n\}$.

If C is a bivariate Archimedean copula generated by a twice continuously differentiable generator ψ with $\psi(t) > 0$ for all $t \in [0, \infty)$, Kendall's tau can be represented in semi-closed form as

$$\tau = 1 + 4 \int_0^1 \frac{\psi^{-1}(t)}{(\psi^{-1}(t))'} dt = 1 - 4 \int_0^\infty t(\psi'(t))^2 dt$$

(see [Joe, 1997, p. 91]) which can often be computed explicitly; see Table 1.

[Genest and Rivest, 1993] introduce a method-of-moments estimator for bivariate one-parameter Archimedean copulas based on Kendall's tau. The copula parameter $\theta_0 \in \Theta \subseteq \mathbb{R}$ is estimated by $\hat{\theta}_n$ such that

$$\tau(\hat{\theta}_n) = \hat{\tau}_n,$$

where $\tau(\theta)$ denotes Kendall's tau of the corresponding Archimedean family viewed as a function of the parameter $\theta \in \Theta \subseteq \mathbb{R}$. In other words,

$$\hat{\theta}_n = \tau^{-1}(\hat{\tau}_n), \quad (4)$$

assuming the inverse τ^{-1} of τ exists. This estimation method obviously only applies to one-parameter families. Otherwise, the set of all parameters with equal Kendall's tau is a level curve and so Kendall's tau cannot be uniquely inverted. If (4) has no solution, this estimation method does not lead to an estimator. Note that unless there is an explicit form for τ^{-1} , $\hat{\theta}_n$ is computed by numerical root finding.

[Berg, 2009] and [Savu and Tiede, 2010] apply this method to data of dimension $d > 2$ by using pairwise sample versions of Kendall's tau. If $\hat{\tau}_{n,j_1j_2}$ denotes the sample version of Kendall's tau between the j_1 th and j_2 th data column, then θ is estimated by

$$\hat{\theta}_n = \tau^{-1} \left(\binom{d}{2}^{-1} \sum_{1 \leq j_1 < j_2 \leq d} \hat{\tau}_{n,j_1j_2} \right). \quad (5)$$

We denote this estimator or estimation method by $\tau_{\hat{\theta}}$. Intuitively, the parameter is chosen such that Kendall's tau equals the average over all pairwise sample versions of Kendall's tau. Note that properties of this estimator are not known and also not easy to derive since the average is taken over dependent data columns. In particular, although $\binom{d}{2}^{-1} \sum_{1 \leq j_1 < j_2 \leq d} \hat{\tau}_{n,j_1j_2}$ is unbiased for $\tau(\theta_0)$, the estimator in (5) need not be unbiased for θ_0 .

Another "pairwise" estimator can be obtained by first computing the $\binom{d}{2}$ pairwise estimators as given in (4) and then average over the estimators, that is,

$$\hat{\theta}_n = \binom{d}{2}^{-1} \sum_{1 \leq j_1 < j_2 \leq d} \tau^{-1}(\hat{\tau}_{n,j_1j_2}).$$

This unbiased estimator can be found in [Kojadinovic and Yan, 2010]; see, for example, the function `fitCopula(, method="itau")` in the R package `copula`. We denote it or the corresponding estimation method by $\tau_{\hat{\theta}}$.

3.2. Blomqvist's beta

Blomqvist's beta (see, for example, [Nelsen, 2006, p. 182]) is also a measure of concordance. In the bivariate case with $X_j \sim F_j$, $j \in \{1, 2\}$, it is defined by

$$\beta = \mathbb{P}((X_1 - F_1^-(1/2))(X_2 - F_2^-(1/2)) > 0) - \mathbb{P}((X_1 - F_1^-(1/2))(X_2 - F_2^-(1/2)) < 0)$$

and therefore measures the probability of falling into the first or third quadrant minus the probability of falling into the second or fourth quadrant, the quadrants being defined by the medians $F_j^-(1/2)$, $j \in \{1, 2\}$. This measure can be expressed in terms of the copula of $(X_1, X_2)^\top$. It also allows for a natural generalization to $d > 2$, given by

$$\beta = \frac{2^{d-1}}{2^{d-1} - 1} (C(1/2, \dots, 1/2) + \hat{C}(1/2, \dots, 1/2) - 2^{1-d});$$

see, for example, [Schmid and Schmidt, 2007]. Here, \hat{C} denotes the survival copula corresponding to C . For Archimedean copulas as given in (2), Blomqvist's beta is easily seen to be

$$\beta = \frac{2^{d-1}}{2^{d-1} - 1} \left(\psi(d\psi^{-1}(1/2)) + \left(\sum_{j=0}^d \binom{d}{j} (-1)^j \psi(j\psi^{-1}(1/2)) \right) - 2^{1-d} \right). \quad (6)$$

Given the random sample \mathbf{U}_i , $i \in \{1, \dots, n\}$, the *sample version of Blomqvist's beta* is given by

$$\hat{\beta}_n = \frac{2^{d-1}}{2^{d-1} - 1} \left(\frac{1}{n} \sum_{i=1}^n \left(\prod_{j=1}^d \mathbb{1}_{\{U_{ij} \leq 1/2\}} + \prod_{j=1}^d \mathbb{1}_{\{U_{ij} > 1/2\}} \right) - 2^{1-d} \right) \quad (7)$$

For asymptotic properties of $\hat{\beta}_n$, see [Schmid and Schmidt, 2007].

A method-of-moments estimator based on Blomqvist's beta can be obtained via

$$\hat{\theta}_n = \beta^{-1}(\hat{\beta}_n),$$

where $\beta(\theta)$ denotes β as a function of the parameter $\theta \in \Theta \subseteq \mathbb{R}$. We denote this estimator or estimation method by β . As for Kendall's tau, this estimation method only applies to the one-parameter case. Typically, $\hat{\theta}_n$ is computed via numerical root finding.

3.3. Minimum distance estimation

[Hering and Hofert, 2013] present a transformation for Archimedean copulas that is analogous to Rosenblatt's transform but simpler to compute. Consider a d -monotone generator ψ and let \mathbf{U} follow the Archimedean copula C with generator ψ . Furthermore, let the *Kendall distribution function* K (that is, the distribution function of the *probability integral transformation* $C(\mathbf{U})$) be continuous. Then, the transformed random vector $\mathbf{U}' = T_\psi(\mathbf{U})$ with

$$U'_j = \left(\frac{\sum_{k=1}^j \psi^{-1}(U_k)}{\sum_{k=1}^{j+1} \psi^{-1}(U_k)} \right)^j, \quad j \in \{1, \dots, d-1\}, \quad U'_d = K(C(\mathbf{U})) \quad (8)$$

follows a uniform distribution on $[0, 1]^d$, denoted by $\mathbf{U}' \sim U[0, 1]^d$. Note that if $\psi \in \Psi_\infty$, then $K(t) = \sum_{k=0}^{d-1} \frac{\psi^{(k)}(\psi^{-1}(t))}{k!} (-\psi^{-1}(t))^k$; see [Barbe et al., 1996] or [McNeil and Nešlehová, 2009]. The transformation (8) allows one to easily derive a minimum distance estimator. First, one transforms the random vectors $\mathbf{U}_i, i \in \{1, \dots, n\}$, with T_ψ and then minimizes a "distance" between the transformed variates and the multivariate uniform distribution. This could be achieved, for example, with the statistics $S_n^{(B)}$ or $S_n^{(C)}$ used by [Genest et al., 2009]. For simplicity and run-time performance, however, we map the transformed variates to univariate quantities via

$$Y_i^n = \sum_{j=1}^d (\Phi^{-1}(U'_{ij}))^2 \quad \text{or} \quad Y_i^1 = \sum_{j=1}^d -\log U'_{ij}, \quad i \in \{1, \dots, n\},$$

where Φ^{-1} denotes the quantile function of the standard normal distribution. Such mappings to a univariate setting are known from goodness-of-fit testing; see [D'Agostino and Stephens, 1986, p. 97]. If the transformation $T_\psi(\mathbf{U})$ is applied with the correct parameter, then $Y_i^n \sim F_{\chi_d^2}$ and $Y_i^1 \sim F_{\Gamma_d}$, $i \in \{1, \dots, n\}$, that is, Y_i^n and Y_i^1 should follow a chi-square distribution with d degrees of freedom and a $\Gamma(d, 1)$ distribution, respectively. Hence, minimum distance estimators can be

obtained via the Cramér-von Mises and Kolmogorov-Smirnov type of distances

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_n^{\text{n,CvM}} &= \operatorname{arginf}_{\boldsymbol{\theta} \in \Theta} n \int_{-\infty}^{\infty} |\hat{F}_{n,Y^n}(x) - F_{\chi_d^2}(x)|^2 dF_{\chi_d^2}(x) \\
&= \operatorname{arginf}_{\boldsymbol{\theta} \in \Theta} \frac{1}{12n} + \sum_{i=1}^n \left(\frac{2i-1}{2n} - F_{\chi_d^2}(Y_{(i)}^n) \right)^2, \\
\hat{\boldsymbol{\theta}}_n^{\text{n,KS}} &= \operatorname{arginf}_{\boldsymbol{\theta} \in \Theta} \sup_x |\hat{F}_{n,Y^n}(x) - F_{\chi_d^2}(x)| \\
&= \operatorname{arginf}_{\boldsymbol{\theta} \in \Theta} \max_{i \in \{1, \dots, n\}} \left\{ F_{\chi_d^2}(Y_{(i)}^n) - \frac{i-1}{n}, \frac{i}{n} - F_{\chi_d^2}(Y_{(i)}^n) \right\}, \\
\hat{\boldsymbol{\theta}}_n^{\text{l,CvM}} &= \operatorname{arginf}_{\boldsymbol{\theta} \in \Theta} n \int_{-\infty}^{\infty} |\hat{F}_{n,Y^1}(x) - F_{\Gamma_d}(x)|^2 dF_{\Gamma_d}(x) \\
&= \operatorname{arginf}_{\boldsymbol{\theta} \in \Theta} \frac{1}{12n} + \sum_{i=1}^n \left(\frac{2i-1}{2n} - F_{\Gamma_d}(Y_{(i)}^1) \right)^2, \\
\hat{\boldsymbol{\theta}}_n^{\text{l,KS}} &= \operatorname{arginf}_{\boldsymbol{\theta} \in \Theta} \sup_x |\hat{F}_{n,Y^1}(x) - F_{\Gamma_d}(x)| \\
&= \operatorname{arginf}_{\boldsymbol{\theta} \in \Theta} \max_{i \in \{1, \dots, n\}} \left\{ F_{\Gamma_d}(Y_{(i)}^1) - \frac{i-1}{n}, \frac{i}{n} - F_{\Gamma_d}(Y_{(i)}^1) \right\},
\end{aligned}$$

where \hat{F}_{n,Y^n} and \hat{F}_{n,Y^1} denote the empirical distribution functions based on $(Y_i^n)_{i \in \{1, \dots, n\}}$ and $(Y_i^1)_{i \in \{1, \dots, n\}}$, respectively, and $Y_{(i)}^n$ and $Y_{(i)}^1$, $i \in \{1, \dots, n\}$, denote the order statistics of $(Y_i^n)_{i \in \{1, \dots, n\}}$ and $(Y_i^1)_{i \in \{1, \dots, n\}}$, respectively. We denote these four estimators or estimation methods by $\text{MDE}_{\chi}^{\text{CvM}}$, $\text{MDE}_{\chi}^{\text{KS}}$, $\text{MDE}_{\Gamma}^{\text{CvM}}$, and $\text{MDE}_{\Gamma}^{\text{KS}}$, respectively.

In large dimensions, one can omit the possibly costly computation of U'_d and work with U'_j , $j \in \{1, \dots, d-1\}$, only, see [Hering, 2011, pp. 52]. Note that minimum distance estimators naturally also work for $p \geq 2$, that is, parameter vectors $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$.

3.4. Maximum-likelihood estimation

According to [McNeil and Nešlehová, 2009], an Archimedean copula C admits a density c if and only if $\psi^{(d-1)}$ exists and is absolutely continuous on $(0, \infty)$. In this case, c is given by

$$c(\mathbf{u}) = \psi^{(d)}(t(\mathbf{u})) \prod_{j=1}^d (\psi^{-1})'(u_j) = \frac{\psi^{(d)}(t(\mathbf{u}))}{\prod_{j=1}^d \psi'(\psi^{-1}(u_j))}, \quad \mathbf{u} \in (0, 1)^d, \quad (9)$$

where, as in (2), $t(\mathbf{u}) = \sum_{j=1}^d \psi^{-1}(u_j)$. Note that for computing the log-density, it is convenient to write c as

$$c(\mathbf{u}) = (-1)^d \psi^{(d)}(t(\mathbf{u})) \prod_{j=1}^d -(\psi^{-1})'(u_j) = \frac{(-1)^d \psi^{(d)}(t(\mathbf{u}))}{\prod_{j=1}^d -\psi'(\psi^{-1}(u_j))}.$$

Given the sample $\mathbf{U}_i, i \in \{1, \dots, n\}$, finding the maximum-likelihood estimator (MLE) usually involves solving the optimization problem

$$\hat{\boldsymbol{\theta}}_n = \operatorname{argsup}_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^n \log c_{\boldsymbol{\theta}}(\mathbf{U}_i),$$

where here and in the following the subscript $\boldsymbol{\theta}$ is used to stress the dependence on $\boldsymbol{\theta}$. This requires an efficient strategy for evaluating the (log-)density. The most important part is to know how to derive and compute the generator derivatives. Tools like automatic differentiation, see [Griewank and Walther, 2003], might provide a solution. Recently, [Hofert et al., 2012] presented explicit formulas for all families listed in Table 1. The corresponding copula densities are reported here for the reader's convenience (note that $\alpha = 1/\theta$):

1. For the family of Ali-Mikhail-Haq,

$$c_{\theta}(\mathbf{u}) = \frac{(1-\theta)^{d+1}}{\theta^2} \frac{h_{\theta}^{\Lambda}(\mathbf{u})}{\prod_{j=1}^d u_j^2} \operatorname{Li}_{-d}(h_{\theta}^{\Lambda}(\mathbf{u})),$$

where $\operatorname{Li}_{-s}(z) = \sum_{k=1}^{\infty} \frac{z^k}{k^s}$ is the *polylogarithm of order s at z* and $h_{\theta}^{\Lambda}(\mathbf{u}) = \theta \prod_{j=1}^d \frac{u_j}{1-\theta(1-u_j)}$.

2. For the family of Clayton,

$$c_{\theta}(\mathbf{u}) = \prod_{k=0}^{d-1} (\theta k + 1) \left(\prod_{j=1}^d u_j \right)^{-(1+\theta)} (1 + t_{\theta}(\mathbf{u}))^{-(d+\alpha)}.$$

3. For the family of Frank,

$$c_{\theta}(\mathbf{u}) = \left(\frac{\theta}{1 - e^{-\theta}} \right)^{d-1} \operatorname{Li}_{-(d-1)}(h_{\theta}^{\text{F}}(\mathbf{u})) \frac{\exp(-\theta \sum_{j=1}^d u_j)}{h_{\theta}^{\text{F}}(\mathbf{u})},$$

where $h_{\theta}^{\text{F}}(\mathbf{u}) = (1 - e^{-\theta})^{1-d} \prod_{j=1}^d (1 - \exp(-\theta u_j))$.

4. For the family of Gumbel,

$$c_{\theta}(\mathbf{u}) = \theta^d \exp(-t_{\theta}(\mathbf{u})^{\alpha}) \frac{\prod_{j=1}^d (-\log u_j)^{\theta-1}}{t_{\theta}(\mathbf{u})^d \prod_{j=1}^d u_j} P_{d,\alpha}^{\text{G}}(t_{\theta}(\mathbf{u})^{\alpha}),$$

where

$$P_{d,\alpha}^{\text{G}}(x) = \sum_{k=1}^d a_{dk}^{\text{G}}(\alpha) x^k,$$

$$a_{dk}^{\text{G}}(\alpha) = (-1)^{d-k} \sum_{j=k}^d \alpha^j s(d, j) S(j, k) = \frac{d!}{k!} \sum_{j=1}^k \binom{k}{j} \binom{\alpha j}{d} (-1)^{d-j}, \quad k \in \{1, \dots, d\},$$

and s and S denote the *Stirling numbers of the first kind* and the *second kind*, respectively, given by the recurrence relations

$$s(n+1, k) = s(n, k-1) - ns(n, k),$$

$$S(n+1, k) = S(n, k-1) + kS(n, k),$$

for all $k \in \mathbb{N}, n \in \mathbb{N}_0$, with $s(0, 0) = S(0, 0) = 1$ and $s(n, 0) = s(0, n) = S(n, 0) = S(0, n) = 0$ for all $n \in \mathbb{N}$.

5. For the family of Joe,

$$c_{\theta}(\mathbf{u}) = \theta^{d-1} \frac{\prod_{j=1}^d (1 - u_j)^{\theta-1}}{(1 - h_{\theta}^J(\mathbf{u}))^{1-1/\theta}} P_{d,\alpha}^J \left(\frac{h_{\theta}^J(\mathbf{u})}{1 - h_{\theta}^J(\mathbf{u})} \right),$$

where

$$P_{d,\alpha}^J(x) = \sum_{k=0}^{d-1} a_{dk}^J(\alpha) x^k,$$

$$a_{dk}^J(\alpha) = S(d, k+1)(k - \alpha)_k, \quad k \in \{0, \dots, d-1\},$$

$h_{\theta}^J(\mathbf{u}) = \prod_{j=1}^d (1 - (1 - u_j)^{\theta})$, and $(k - \alpha)_k = \frac{\Gamma(k+1-\alpha)}{\Gamma(1-\alpha)}$ denotes the falling factorial.

Example 3.1

The left-hand side of Figure 1 shows the log-likelihood of a Clayton copula based on a 100-dimensional sample of size $n = 100$ with parameter $\theta_0 = 2$ such that the corresponding bivariate population version of Kendall’s tau equals $\tau(\theta_0) = 0.5$. The MLE is denoted by $\hat{\theta}_n$. The right-hand side of Figure 1 shows the log-likelihood plot of a 100-dimensional Gumbel family with parameter $\theta_0 = 2$ such that Kendall’s tau equals $\tau(\theta_0) = 0.5$. Both Figures are plotted on the interval $[\tau^{-1}(\tau(\theta_0) - h), \tau^{-1}(\tau(\theta_0) + h)]$, where $h = 0.05$ denotes a “distance” in terms of concordance. Note that evaluating the log-density of a Gumbel copula is numerically highly complicated; see Section 5.3 for more details.

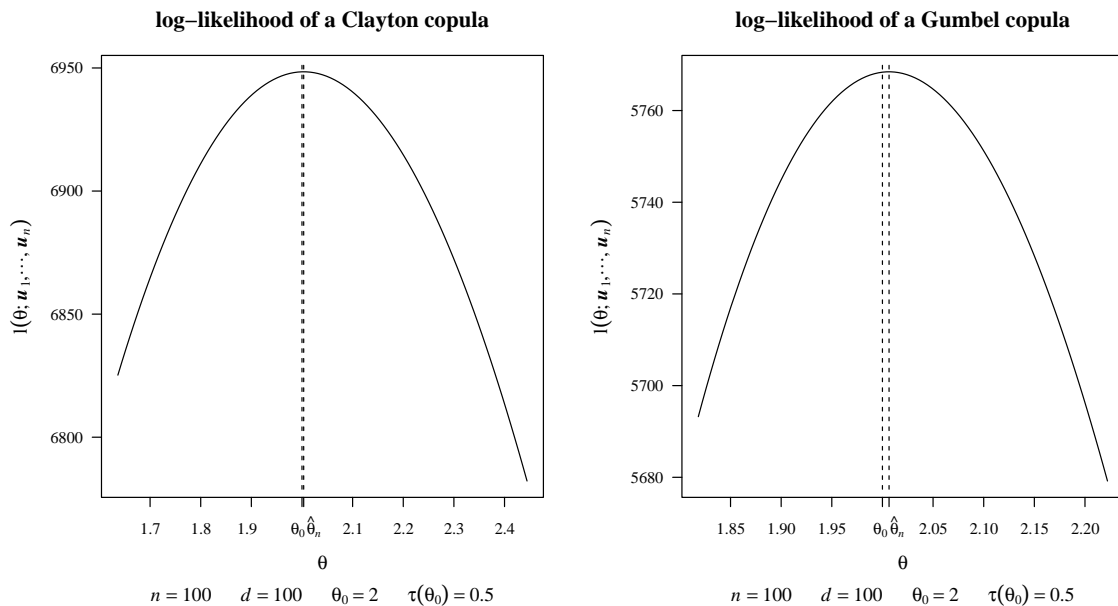


FIGURE 1. Plot of the log-likelihood of a Clayton (left) and Gumbel (right) copula based on a sample of size $n = 100$ in dimension $d = 100$ with parameter $\theta_0 = 2$ such that Kendall’s tau equals 0.5.

3.5. Simulated maximum-likelihood estimation

If the derivatives of a given Archimedean generator ψ are not known explicitly one may use the fact that ψ is an expectation in order to approximate the density of the generated copula. This way one can replace derivatives of higher order by just one integral (which can either be evaluated numerically or via Monte Carlo simulation). If $\psi \in \Psi_\infty$, then $\psi(t) = \mathcal{L}\mathcal{S}[F](t) = \int_0^\infty \exp(-xt) dF(x)$, so that by differentiating under the integral sign one obtains

$$(-1)^d \psi^{(d)}(t) = \int_0^\infty x^d \exp(-xt) dF(x) = \mathbb{E}[V^d \exp(-Vt)], \quad t \in (0, \infty),$$

where V has distribution function F . An approximation to $(-1)^d \psi^{(d)}(t)$ is thus given by

$$(-1)^d \psi^{(d)}(t) \approx \frac{1}{m} \sum_{k=1}^m V_k^d \exp(-V_k t), \quad t \in (0, \infty), \quad (10)$$

where $V_k \sim F$, $k \in \{1, \dots, m\}$, are realizations of iid random variables following $F = \mathcal{L}\mathcal{S}^{-1}[\psi]$. Instructions for how to sample F for the one-parameter families in Table 1 can be found, for example, in [Hofert, 2012]; see also [Hofert and Mächler, 2011]. This method can be used to evaluate the copula density. We refer to the corresponding MLE as *simulated maximum-likelihood estimator* (SMLE). Finally, note that both the MLE and the SMLE naturally apply to the multi-parameter case.

3.6. Diagonal maximum-likelihood estimation

The diagonal $\delta(u) = C(u, \dots, u)$ of a copula C suggests a simple and straightforward estimation procedure which, as far as we are aware, has not been exploited for estimating (Archimedean) copulas so far. Note that the diagonal δ_θ of a parametric copula family $(C_\theta)_{\theta \in \Theta}$ is a distribution function and, for $\mathbf{U} \sim C$,

$$Y = \max_{1 \leq j \leq d} U_j \sim \delta_\theta.$$

Based on the sample \mathbf{U}_i , $i \in \{1, \dots, n\}$, with corresponding maxima Y_i , $i \in \{1, \dots, n\}$, one can apply maximum-likelihood estimation to find an estimator $\hat{\theta}_n$ of the parameter vector θ_0 via

$$\hat{\theta}_n = \operatorname{argsup}_{\theta \in \Theta} \sum_{i=1}^n \log \delta'_\theta(Y_i), \quad (11)$$

where δ'_θ denotes the density of the distribution function δ_θ . We refer to the estimator $\hat{\theta}_n$ as *diagonal maximum-likelihood estimator* (DMLE). For Archimedean copulas, $\delta_\theta(u) = \psi_\theta(d\psi_\theta^{-1}(u))$ and hence,

$$\delta'_\theta(u) = d\psi'_\theta(d\psi_\theta^{-1}(u))(\psi_\theta^{-1})'(u), \quad u \in [0, 1]. \quad (12)$$

Therefore, one advantage of the DMLE is that the degree of numerical difficulty of the optimization in (11) (theoretically) remains rather unaffected by the dimension. Another advantage may be its

computational speed, which makes it useful for computing initial values for more sophisticated estimators. As a drawback, the DMLE only uses information from the copula diagonal.

For the one-parameter Gumbel family, (11) can be solved explicitly, the estimator $\hat{\theta}_n^G$ of θ being

$$\hat{\theta}_n^G = \frac{\log d}{\log n - \log(\sum_{i=1}^n -\log Y_i)}.$$

An adjusted estimator of the form

$$\hat{\theta}_n^{G,*} = \max\{\hat{\theta}_n^G, 1\}$$

is then guaranteed to provide an admissible parameter estimator for Gumbel's family.

4. A large-scale simulation study

In this section, we present a large-scale simulation study in which we compare the performance of the different estimators presented in Section 3 both under known and (from an applied point of view the more interesting case of) unknown margins (pseudo-observations). To the best of our knowledge, this is the first study of this kind also addressing large dimensions (up to 100). To be able to also include the estimators based on measures of concordance, we restrict ourselves to the one-parameter families as given in Table 1.

4.1. A word concerning the implementation

The results presented in this section are based on the following computational set-up. The procedures are computationally challenging in many ways and much effort has gone into accurate and efficient implementation in R; see Section 5. The latest version of the package can be accessed via <http://nacopula.r-forge.r-project.org/>. The computations are carried out on the computer cluster Brutus of ETH Zurich which runs CentOS 5.4. The batch jobs are run on nodes with four quad-core AMD Opteron 8380 CPUs and 32 GB of RAM. Apart from the physical structure of the grid, the compiler, and the programming language, note that run time also depends on other factors such as the quality of the implementation or the current load of the machine. The presented run times should therefore be viewed with this in mind.

4.2. The experimental design

In the simulation study carried out, we consider both known and unknown margins. For each of these cases, we generate $N = 1000$ samples of size n from iid random vectors following a d -dimensional Archimedean copula with prespecified parameter θ such that the corresponding Kendall's tau is $\tau \in \{0.25, 0.75\}$. For the case of known margins, we base estimation directly on these samples. For the case of unknown margins, we base estimation on the corresponding pseudo-observations as given in (3). Since we are mainly interested in the behavior for different dimensions d , we consider $d \in \{5, 20, 100\}$ and keep $n = 100$ fixed. The dimensions considered thus go up to the sample size n in which case the data matrices have 10000 entries; note that such

a setup is computationally highly challenging for all the estimators we consider and a novelty in the context of dependence modeling with copulas. We investigate the one-parameter families of Ali-Mikhail-Haq (only for $\tau = 0.25$ since the range of admissible Kendall's tau for this family is bounded from above by $1/3$), Clayton, Frank, Gumbel, and Joe. The average pairwise Kendall's tau estimator ($\tau_{\bar{\tau}}$), the average of Kendall's tau estimators ($\tau_{\bar{\theta}}$), the estimation method based on Blomqvist's beta (β), the four presented minimum distance estimators ($\text{MDE}_{\chi}^{\text{CvM}}$, $\text{MDE}_{\chi}^{\text{KS}}$, $\text{MDE}_{\Gamma}^{\text{CvM}}$, and $\text{MDE}_{\Gamma}^{\text{KS}}$), maximum-likelihood (MLE), simulated maximum-likelihood (SMLE), and diagonal maximum-likelihood (DMLE) estimators are applied to estimate the parameter for each of the N data sets. Finally, bias and root mean squared error (RMSE), as well as mean user run time over all N replications are computed.

For the required optimizations for the estimators based on Blomqvist's beta, all minimum distance estimators, MLE, SMLE, and DMLE, we use initial intervals determined from a large range of (admissible) Kendall's tau; see the implementation of the function `initOpt` in the R package `copula` for more details. For the minimum distance estimators to be competitive according to run time, we only include the Kendall distribution function in the transformation given in (8) in the five-dimensional case, but not for higher dimensions. For applying the SMLE, we draw 10000 random variates from $V \sim F = \mathcal{L}\mathcal{S}[\psi]$ for each evaluation of the density of the Archimedean copula.

4.3. Results under known margins

Tables 2, 3, and 4 in the appendix contain the bias (multiplied by 1000), the RMSE (multiplied by 1000), and the mean user times in milliseconds (MUT), respectively, for all investigated estimators under known margins. For each entry, the number in parentheses denotes the entry divided by the corresponding entry of the MLE column, so that the performance with respect to the MLE can easily be determined; the MLE itself thus has always 1.0 in parentheses. For the RMSEs, note that the reciprocals of the square of these numbers are also known as the (estimated) *relative efficiency* of the MLE with respect to the corresponding estimator.

Figures 2 and 3 graphically display the square root of the absolute error via box plots obtained from the $N = 1000$ replications. Due to readability, we exclude methods which perform so poorly that their boxplots dominate the scale of values. In particular, this is the case for the estimators based on Blomqvist's beta for $d = 100$ for the families of Clayton, Frank, and Joe and the SMLE for Clayton and $\tau = 0.25$.

The results from this study under known margins can be summarized as follows:

- The performance of the average of bivariate Kendall's tau estimator $\tau_{\bar{\theta}}$ as given in the end of Section 3.1 is very similar to the one of the averaged pairwise Kendall tau estimator $\tau_{\bar{\tau}}$. One problem that especially $\tau_{\bar{\theta}}$ faces is that sample versions of Kendall's tau are sometimes not in the range of tau as a function of theta. These values were then mapped to the range of admissible Kendall's tau, see the function `tau.checker` in `copula`. Furthermore, run time for method $\tau_{\bar{\theta}}$ is typically larger (especially in large dimensions) than that for $\tau_{\bar{\tau}}$, which is clear since more inversions of Kendall's tau have to be performed. Overall, $\tau_{\bar{\tau}}$ is thus preferred. Furthermore, due to only considering pairs at a time, this estimator is quite robust against numerical difficulties. A disadvantage, however, is its large run time due to the quadratic complexity in the dimension d .

- Although Blomqvist’s beta can be flawlessly applied to estimate the copula parameter θ for small and moderate dimensions d , this estimator shows serious numerical problems for large values of d uniformly over all investigated Archimedean families. One of the problems turns out to be that both products appearing in the sample version (7) of Blomqvist’s beta are sometimes zero, so that $\hat{\beta}_n < 0$ although $\beta \geq 0$. Another problem is that the evaluation of the survival copula at $(1/2, \dots, 1/2)^\top$ turns out to be numerically challenging for several families; see Section 5.6 for more details.
- The performance of the minimum distance estimators depends on the mapping to the one-dimensional setting applied. In particular, the estimators $\text{MDE}_\Gamma^{\text{CvM}}$ and $\text{MDE}_\Gamma^{\text{KS}}$ based on the logarithmic transformation to a Gamma distribution do not perform well in comparison to $\text{MDE}_\chi^{\text{CvM}}$ and $\text{MDE}_\chi^{\text{KS}}$. Furthermore, the minimum distance estimators based on the Kolmogorov-Smirnov distances are outperformed by those based on the Cramér-von Mises distances (also according to run time in most of the cases investigated, see Table 4). Note that run time for $d = 5$ is larger than for $d \in \{20, 100\}$ because we applied the full transformation T_ψ including the Kendall distribution function K in the five-dimensional case. Moreover, the distances (objective functions) had to be reparameterized in order for the minimum distance estimators to be computed; see Section 5.1 for more details.
- With the explicit formulas for the densities we presented, the MLE clearly shows the best performance under known margins. Note that the run times are much smaller than one would expect in comparison to other estimators, although, our implementation was written with focus on readability rather than run-time performance and thus several quantities are computed each time the density is evaluated. In contrast to statements found in the literature (see, for example, [Berg and Aas, 2009] or [Weiß, 2010]) this leaves no doubt that maximum-likelihood estimation is feasible in large dimensions and performs well; for the latter, see also [Hofert et al., 2012] who empirically show that the mean squared error (MSE) satisfies

$$\text{MSE} \propto \frac{1}{nd}$$

in the case of known margins.

- The SMLE also shows an incredible performance, the only exception being Clayton’s family; see Section 5.2 for more details. The RMSEs are close to the ones obtained by maximum-likelihood estimation. Moreover, this method is straightforward to implement given random number generators for the distribution corresponding to the generator under consideration. The drawback of this method is certainly a larger run time. Note that this could be partly reduced, for example, by using an adaptive technique in which smaller amounts of random variates are drawn during the first couple of steps the optimizer performs. Also note that with the constant use of 5000 random variates (instead of 10 000) per density evaluation, the overall performance of the SMLE is still slightly better than those of the minimum distance estimator $\text{MDE}_\chi^{\text{CvM}}$.
- The advantage of the DMLE lies in its speed. Due to this fact, this estimator could be used for finding initial values for more sophisticated estimation methods.

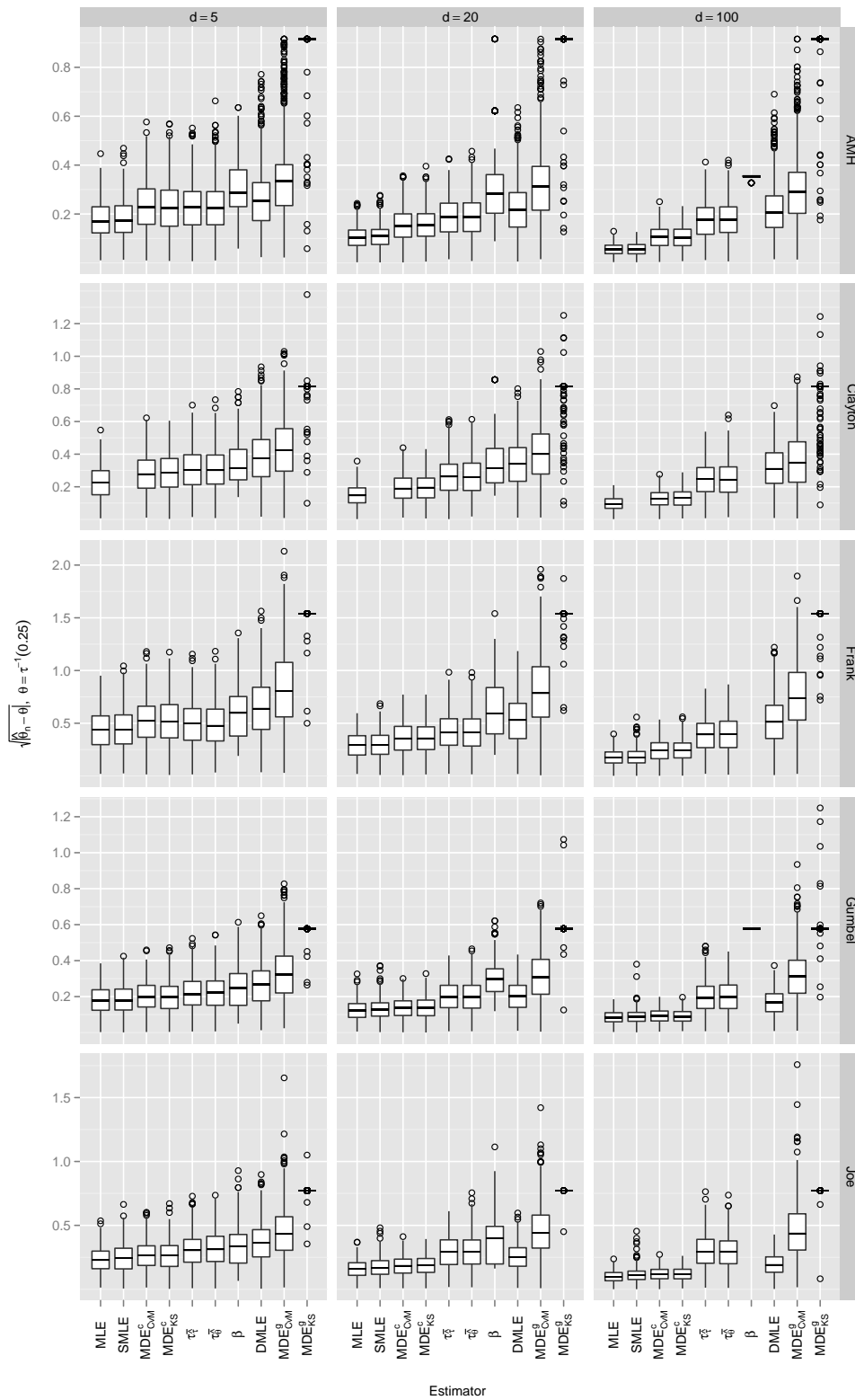


FIGURE 2. Box plots of the square root of the absolute error (under known margins) obtained from $N = 1000$ replications of sample size $n = 100$ for Kendall's tau equal to 0.25.

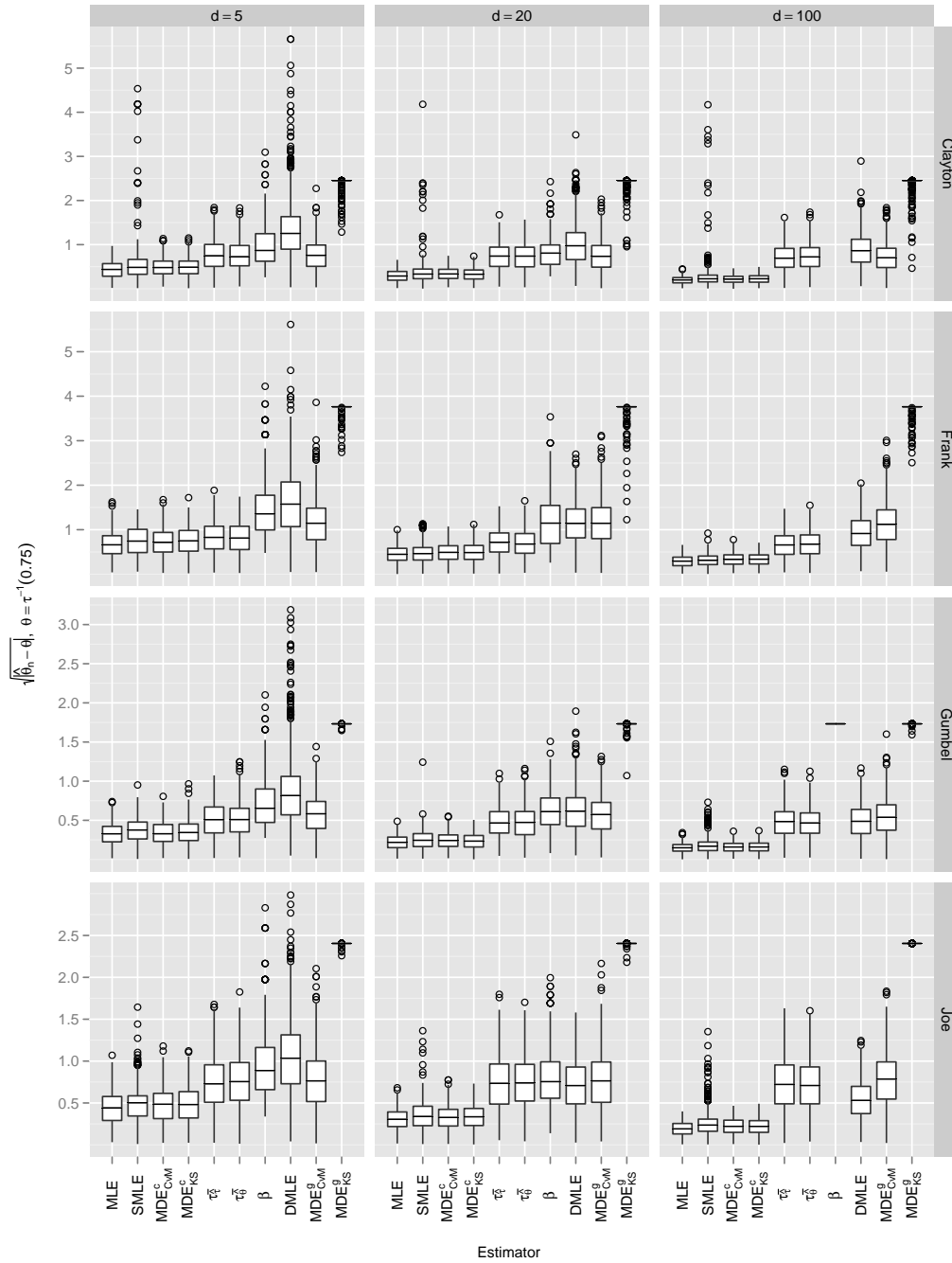


FIGURE 3. Box plots of the square root of the absolute error (under known margins) obtained from $N = 1000$ replications of sample size $n = 100$ for Kendall's tau equal to 0.75.

4.4. Results under unknown margins

Tables 5, 6, and 7 in the appendix contain the bias (multiplied by 1000), the RMSE (multiplied by 1000), and the mean user times in milliseconds (MUT), respectively, for all investigated estimators based on pseudo-observations.

Figures 4 and 5 graphically display the corresponding square root of the absolute error via box plots obtained from the $N = 1000$ replications. As for Figures 2 and 3, we exclude methods which perform so poorly that their boxplots dominate the scale of values. Under pseudo-observations, these were the same methods as under known margins with the only exceptions being $\text{MDE}_{\Gamma}^{\text{KS}}$ for $d \in \{20, 100\}$ which are excluded and SMLE for Clayton which performed better under pseudo-observations and is thus included in the figures; see Section 5.2 for an explanation.

The performance of the estimators based on pseudo-observations can be summarized as follows. Overall, the MLE still performs best, but the differences in absolute error are much less obvious. Furthermore, although a slight improvement of the performance of the MLE in larger dimensions d is visible, the rate of improvement does not seem to be as large as under known margins. Concerning the estimators based on Kendall's tau and the minimum distance estimators, the former performs well for the case $\tau = 0.25$, the latter performs well for the case $\tau = 0.75$.

5. Numerical issues and partial solutions

In this section, we address some specific numerical problems we encountered when working in high dimensions. These problems are not trivial to solve and for some, no simple solution exists to date. We included this section to emphasize that working in large dimensions is much more affected by numerical issues. This is not merely a problem of slow run times; it is also a huge problem for precision. As a general remark, let us stress that what is known about estimators in low dimensions does not always carry over to the high-dimensional case: Estimators that are fast in low dimensions may turn out to be too slow in large dimensions (although robust, the pairwise Kendall's tau estimators face this problem); estimators whose simple form suggest good performance in large dimensions may be highly prone to numerical errors (which is the case, for example, for Blomqvist's beta due to accessing the survival copula involved, a critical task in large dimensions).

5.1. Minimum distance estimators

The minimum distance estimators were especially prone to the problem of a flat objective function for the optimization for all but the Ali-Mikhail-Haq family. Note that there, the parameter θ runs in a bounded interval which is typically advantageous for optimization.

To show the problem, we consider the Gumbel copula and pick out the Cramér-von Mises distances based on the mapping to a χ^2 distribution (via the quantile function of the normal distribution) as described in Section 3.3. The left-hand side of Figure 6 shows the objective function (the distance to be minimized) based on samples of size $n = 100$ from Gumbel copulas in the dimensions $d \in \{5, 20, 100\}$ with parameter $\theta = 4/3$ such that Kendall's tau equals 0.25. We choose the same (large) plotting interval as is chosen for the optimization in the simulation study in Section 4. As can be seen from this figure, the distance to be minimized becomes flat already

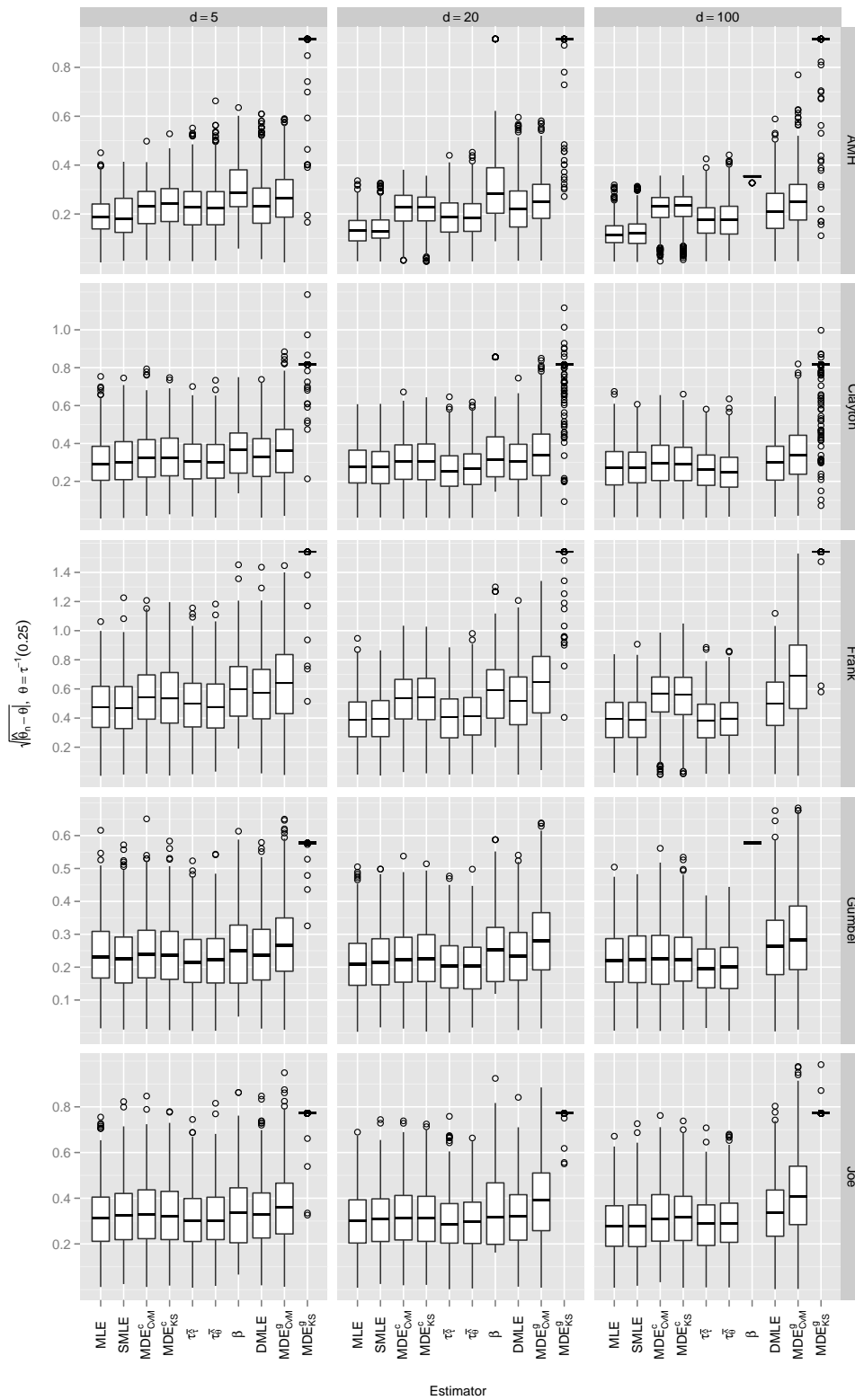


FIGURE 4. Box plots of the square root of the absolute error (under unknown margins) obtained from $N = 1000$ replications of sample size $n = 100$ for Kendall's tau equal to 0.25.

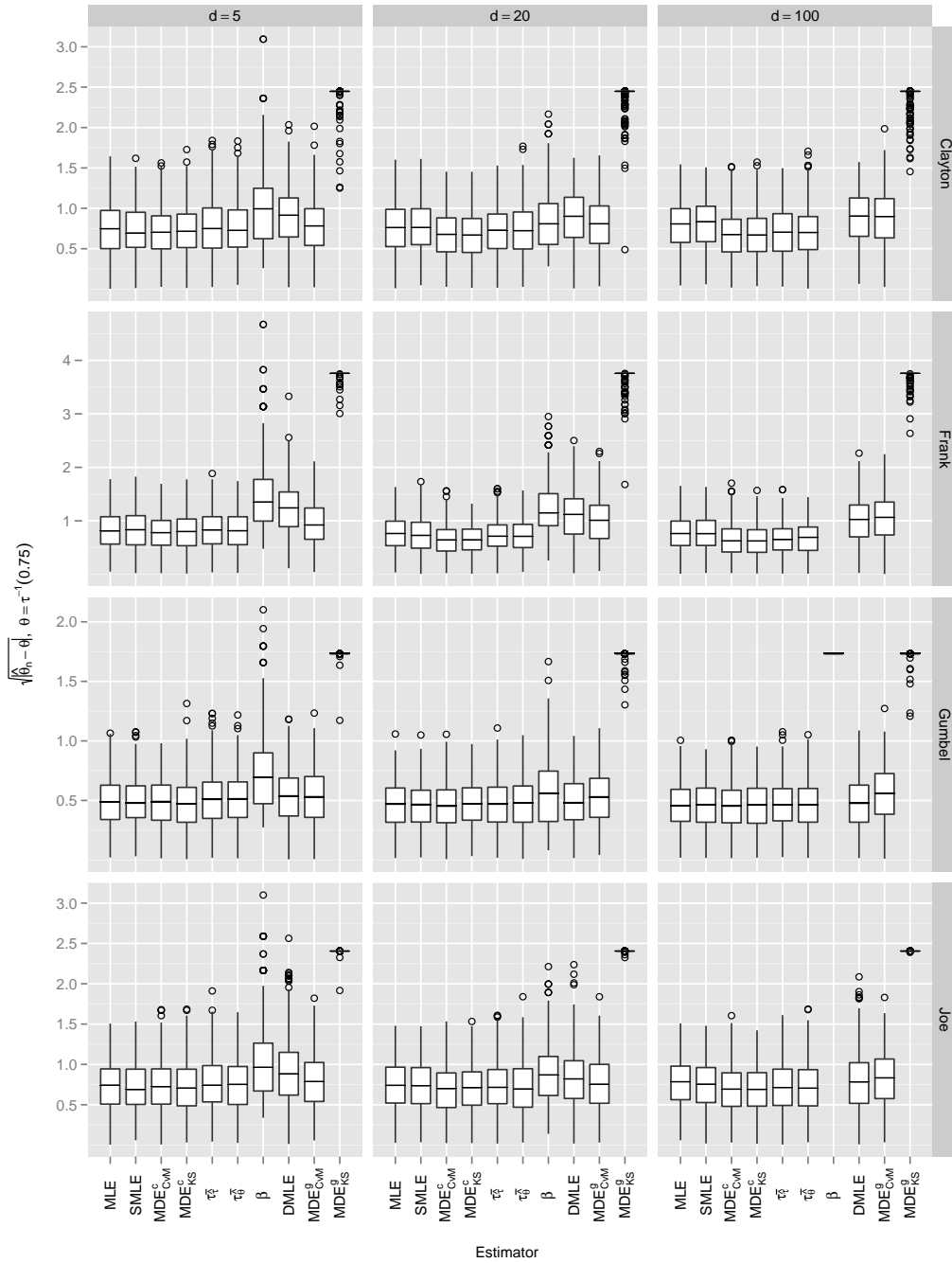


FIGURE 5. Box plots of the square root of the absolute error (under unknown margins) obtained from $N = 1000$ replications of sample size $n = 100$ for Kendall's tau equal to 0.75.

for moderate parameter values. The optimization of this distance carried out in the simulation study is done via \mathbb{R} 's `optimize`. It is indicated on the corresponding help page how this function proceeds. Based on the first two points in the optimization procedure, it is clear that the algorithm remains in the “flat part” of the distance function and thus returns wrong estimates.

The solution to this problem is simple and effective: By reparameterizing the distance one can carry out the optimization without problems. To see why, consider the right-hand side of Figure 6 which shows precisely the same distance as on the left-hand side of this figure, but now plotted in $\alpha = 1 - 1/\theta$. The advantage of this reparameterization is that the objective function is now a function of the bounded variable $\alpha \in (0, 1]$.

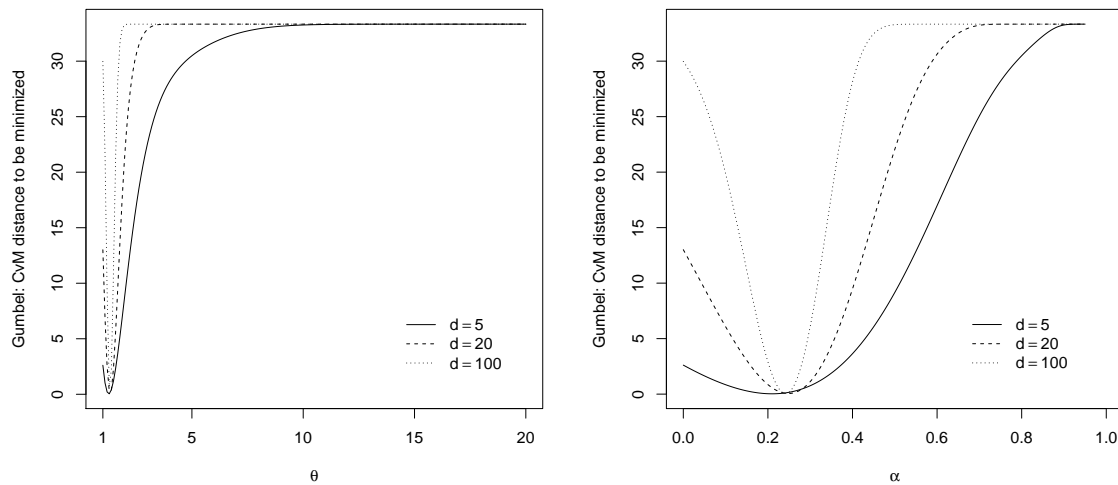


FIGURE 6. Plot of the Cramér-von Mises distances (based on the mapping to a χ^2 distribution) without (left) and with (right) reparameterization of the distance for the Gumbel copula with parameter $\theta = 4/3$ (Kendall's tau equals 0.25) based on a sample of size $n = 100$ in the dimensions indicated.

Similar transformations turn out to be convenient for the families of Clayton, Frank, and Joe as well. For the latter, we use the same reparameterization as for Gumbel, for Clayton and Frank we use $\alpha = 2 \arctan(\theta)/\pi$.

5.2. Simulated maximum-likelihood estimation

As can be seen from the results in Sections 4.3 and 4.4, the SMLE performs well except for Clayton's family under known margins. In this section we briefly investigate why. For this, recall that the SMLE is based on the approximation (10). The log-density approximated via this Monte Carlo method then involves

$$\log((-1)^d \psi^{(d)}(t)) \approx \log\left(\frac{1}{m} \sum_{k=1}^m V_k^d \exp(-V_k t)\right) = \log\left(\frac{1}{m} \sum_{k=1}^m \exp(b_k)\right), \tag{13}$$

where $b_k = d \log(V_k) - V_k t$.

For the SMLE to compute, we have to replace t in (13) by $\sum_{j=1}^d \psi^{-1}(u_j)$. For simplicity, let us assume that all components u_j are equal to u , so we consider the vector $\mathbf{u} = (u, \dots, u)^T \in [0, 1]^d$.

The corresponding value t for (13) is then $t = d\psi^{-1}(u) = d(u^{-\theta} - 1)$. Let us assume that $\theta = 2$, that is, the corresponding value of Kendall's tau is 0.5. If u is small, then t becomes large. The problem is now that the exponents b_k become quite small. Indeed, they are so small (depending on t) that the $\exp(b_k)$ become zero in computer arithmetic for many (again depending on t) of the $m = 10000$ sampled V_k 's. These zeros significantly affect the approximation in (13).

To give an example, let $\theta = 2$, $d = 5$, draw iid $V_k \sim \Gamma(1/\theta, 1)$, $k \in \{1, \dots, m\}$, for $m = 10000$ (using `set.seed(1)`), and compute the approximation in (13) at $t \in \{5 \cdot 10^{16}, 5 \cdot 10^{12}, 5 \cdot 10^8, 15\}$. The left-hand side (correct values) for these values of t are (roughly) -208.09, -157.44, -106.78, and -11.86, whereas the right-hand side gives -Inf, -622.62, -124.41, and -11.86. As one can see, for $t = 15$ both values agree, for $t = 5 \cdot 10^8$, the approximation is already quite far away from the corresponding true value. For large t this problem becomes more severe, with the extreme case being such a large t that $\exp(b_k)$ is zero in computer arithmetic for all V_k 's. This implies that $\log(0) = -\text{Inf}$ in (13). Note that this could be avoided by using an intelligent logarithm as given in Lemma 5.1 1 below. However, this does not solve the problem of a poor approximation to $\log((-1)^d \psi^{(d)}(t))$ (see also Figure 7 below), the problem being that all summands are zero, except the one being $\exp(b_k - b_{\max}) = \exp(0) = 1$.

Figure 7 shows, in log-log scale, the relative error of the approximation (13) for $\mathbf{u} = (u, \dots, u)^\top$ as a function of u , based on $m = 10000$, $d = 5$, and $\theta = 2$. As is clearly visible, the relative error of the approximation becomes much larger for smaller values of u . Since the Clayton copula has lower tail dependence, there is indeed a positive probability of obtaining random vectors with simultaneously small components. These (and only these) samples affect the likelihood approximation and lead to wrong SMLEs. Note that this problem vanishes for the SMLE based on pseudo-observations, see Figures 4 and 5, since each $\hat{u}_{ij} \in \{1/(n+1), \dots, n/(n+1)\}$ and thus the \hat{u}_{ij} 's are naturally bounded from below by $1/(n+1)$. Another option (not discussed here) could be to base the likelihood computation only on those samples not close to zero simultaneously.

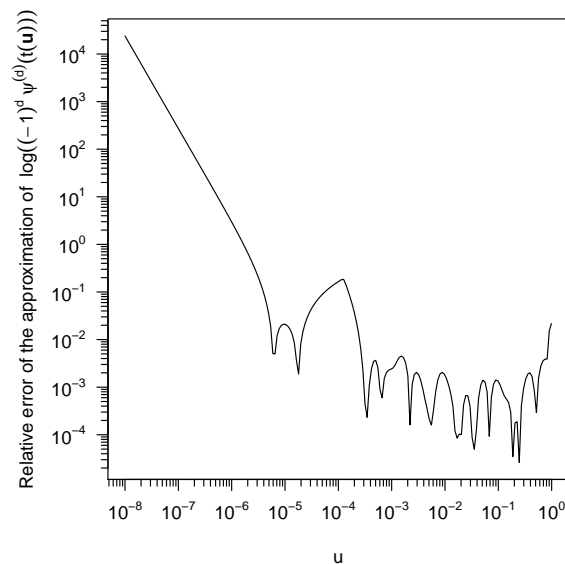


FIGURE 7. Relative error as a function of u based on $m = 10000$, $d = 5$, and $\theta = 2$.

5.3. Gumbel's and Joe's polynomial

For computing the log-likelihood for the Archimedean Gumbel or Joe copula, we need an efficient way of evaluating the logarithm of the density $c_\theta(\mathbf{u})$ as given in Section 3.4, Parts 4 and 5, respectively. The challenge is to evaluate the logarithm of the polynomials involved. For this the following auxiliary results are essential. Their proofs are straightforward and thus omitted.

Lemma 5.1

1. Let $x_i \geq 0$, $i \in \{1, \dots, n\}$, such that $\sum_{i=1}^n x_i > 0$. Furthermore, let $b_i = \log x_i$, $i \in \{1, \dots, n\}$, with $\log 0 = -\infty$, and let $b_{\max} = \max_{1 \leq i \leq n} b_i$. Then

$$\log \sum_{i=1}^n x_i = b_{\max} + \log \sum_{i=1}^n \exp(b_i - b_{\max}). \quad (14)$$

2. Let $x_i \in \mathbb{R}$, $i \in \{1, \dots, n\}$, such that $\sum_{i=1}^n x_i > 0$. Furthermore, let $s_i = \text{sign} x_i$, $b_i = \log |x_i|$, $i \in \{1, \dots, n\}$, with $\log 0 = -\infty$ and let $b_{\max} = \max_{1 \leq i \leq n} b_i$. Then

$$\log \sum_{i=1}^n x_i = b_{\max} + \log \left(\sum_{i=1}^n \exp(b_{(i)} - b_{\max}) - \sum_{i: s_i = -1} \exp(b_{(i)} - b_{\max}) \right), \quad (15)$$

where $b_{(i)}$ denotes the i th smallest value of b_i , $i \in \{1, \dots, n\}$.

The ideas behind Lemma 5.1 1 and 2 are implemented in the (non-exported) functions `lsum` and `lssum` in the R package `copula`.

Although mathematically straightforward, Lemma 5.1 has an important consequence for evaluating logarithms of polynomials such as $P_{d,\alpha}^G$ for Gumbel's or $P_{d,\alpha}^J$ for Joe's density. Depending on the evaluation point, it might happen that the value of the polynomial is not representable in computer arithmetic and thus one cannot first compute the value of the polynomial and take the logarithm afterwards. Instead, Formula (14) suggests a "intelligent" (numerically stable) logarithm to compute such polynomials (or sums). By taking out the maximum of the b_i , it is guaranteed that the exponentials which are summed up are all in $[0, 1]$ and thus the sum takes on values in $[0, n]$, representable in computer arithmetic. This trick solves the numerical issues for computing $P_{d,\alpha}^J$ and thus for computing the log-likelihood of a Joe copula. The evaluation of $P_{d,\alpha}^J$ is implemented as (non-exported) function `polyJ` in the R package `copula`. It is called with default method `log.poly` implementing the trick described above when evaluating the density of a Joe copula via the slot `dacopula`; two other, less efficient methods are also available, one of which is a straightforward polynomial evaluation (`poly`).

Formula (15) takes the above idea of an intelligent logarithm a step further, by dealing with possibly negative summands. The summands in each sum are ordered in increasing order to prevent cancellation. This formula is helpful in computing $P_{d,\alpha}^G$. However, the situation turns out to be more challenging for Gumbel's family. All in all, several different methods for the evaluation of $P_{d,\alpha}^G$ were implemented. They are based on the following results about $P_{d,\alpha}^G$ and described below, where here and in the following, $\alpha = 1/\theta \in (0, 1]$.

Lemma 5.2

Let

$$P_{d,\alpha}^G(x) = \sum_{k=1}^d a_{dk}^G(\alpha) x^k \quad (16)$$

for $\alpha \in (0, 1]$, where

$$a_{dk}^G(\alpha) = (-1)^{d-k} \sum_{j=k}^d \alpha^j s(d, j) S(j, k) \quad (17)$$

$$= \frac{d!}{k!} \sum_{j=1}^k \binom{k}{j} \binom{\alpha j}{d} (-1)^{d-j}, \quad k \in \{1, \dots, d\}. \quad (18)$$

Then

1. $a_{dk}^G(\alpha) = p_{01}^J(d; k) d! / k!$ for all $\alpha \in (0, 1]$, where $p_{01}^J(d; k) > 0$ denotes a probability mass function in $d \in \{k, k+1, \dots\}$;
2. $P_{d,\alpha}^G$ allows for the following representations:

$$P_{d,\alpha}^G(x) = (-1)^{d-1} x \sum_{j=1}^d \left(s(d, j) \sum_{k=0}^{j-1} S(j, k+1) (-x)^k \right) \alpha^j \quad (19)$$

$$= (-1)^{d-1} \alpha x \sum_{j=0}^{d-1} \left(s(d, j+1) \sum_{k=0}^{j-1} S(j, k+1) (-x)^k \right) \alpha^j \quad (20)$$

$$= \sum_{j=1}^d s_j \exp(\log |(\alpha j)_d| + j \log x + x - \log(j!) + \log F^{\text{Poi}(x)}(d-j)), \quad (21)$$

where, for all $j \in \{1, \dots, d\}$,

$$s_j = \begin{cases} (-1)^{j-\lceil \alpha j \rceil}, & \alpha j \notin \mathbb{N} \text{ or } (\alpha = 1, j = d), \\ 0, & \text{otherwise,} \end{cases} \quad (22)$$

and where $F^{\text{Poi}(x)}(\cdot)$ denotes the distribution function of a Poisson distribution with parameter x .

Proof

Part 1 of Lemma 5.2 follows from [Hofert, 2010, p. 99] (the probability mass function $p_{01}^J(d; k) > 0$ corresponds to the distribution function $F_{01}(\cdot; k)$ whose Laplace-Stieltjes transform is the generator $\psi_{01}(\cdot; k)$ appearing in a nested Joe copula). In particular, this equality implies that the coefficients $a_{dk}^G(\alpha)$ of $P_{d,\alpha}^G$ are positive. This allows one to apply (14) with $b_k = \log a_{dk}^G(\alpha) + k \log x$, $k \in \{1, \dots, d\}$ ($d = n$), to compute the logarithm of the polynomial $a_{dk}^G(\alpha)$ at x . Concerning Part 2, Equations (19) and (20) directly follow from interchanging the order of summation of (16) combined with (17). For Equation (21), note that interchanging the order of summation of (16) combined with (18) and rewriting the generalized binomial coefficient $\binom{\alpha j}{d}$ as $(\alpha j)_d / j!$ leads to

$$P_{d,\alpha}^G(x) = \sum_{j=1}^d (\alpha j)_d (-1)^{d-j} \frac{x^j \exp(x)}{j!} \sum_{k=0}^{d-j} \frac{x^k}{k!} \exp(-x).$$

Interpreting the second sum as $F^{\text{Poi}(x)}(d-j)$, pulling out the signs $s_j = \text{sign}((\alpha j)_d (-1)^{d-j})$, and bringing in an $\exp(\log(\cdot))$ leads to the result as stated. Concerning the formula for s_j , note that

$$\begin{aligned} s_j &= \text{sign}((\alpha j)_d (-1)^{d-j}) = (-1)^{d-j} \text{sign} \prod_{l=1}^{d-1} (\alpha j - l) = (-1)^{d-j} \prod_{l=1}^{d-1} \text{sign}(\alpha j - l) \\ &= (-1)^{d-j} \prod_{l=\lceil \alpha j \rceil}^{d-1} \text{sign}(\alpha j - l). \end{aligned} \quad (23)$$

First assume $\alpha j \notin \mathbb{N}$ and $\lceil \alpha j \rceil \leq d-1$. In this case $s_j = (-1)^{d-j} (-1)^{d-1-\lceil \alpha j \rceil+1} = (-1)^{j-\lceil \alpha j \rceil}$. This is also true if $\lceil \alpha j \rceil = d$ since then j has to be equal to d . Now consider $\alpha = 1$ and $j = d$. In this case, it is easily seen from (23) that $s_j = 1$ which is also true for (22). Finally, consider the second case in (22). It implies that $\alpha j \in \mathbb{N}$ and thus the first factor in (23) being zero due to $l = \lceil \alpha j \rceil = \alpha j$, so $s_j = 0$. Note the interesting fact that the formula for s_j is independent of d . \square

The results presented in Lemma 5.2 lead to the following different methods for computing the logarithm of $P_{d,\alpha}^G$, see the (non-exported) function `polyG` of the R package `copula`:

- `pois`, `pois.direct`: These methods are based on Representation (21), where `pois` applies the intelligent logarithm (15) to evaluate the sum and `pois.direct` computes the sum directly as given in (21);
- `stirling`, `stirling.horner`: Method `stirling` evaluates Representation (19) directly, where the polynomial $\sum_{k=0}^{j-1} S(j, k+1)(-x)^k$ in $-x$ is computed via Horner's scheme. Method `stirling.horner` is based on Representation (20), where Horner's scheme is applied to compute the polynomial in α with coefficients $s(d, j+1) \sum_{k=0}^{j-1} S(j, k+1)(-x)^k$ which, as before, are evaluated with Horner's scheme.
- `sort`, `horner`, `direct`, and `dsSib.*`: These methods all $P_{d,\alpha}^G$ with the intelligent logarithm (14) based on the logarithms of the coefficients $a_{dk}^G(\alpha)$ (note that the coefficients $a_{dk}^G(\alpha)$ of are all positive). The logarithmic coefficients can be obtained in different ways: `sort` computes them via (15); `horner` via Horner's scheme based on interpreting (17) as a polynomial in $-\alpha$; `direct` by directly computing the sum as given in (17); and `dsSib.*` by various different methods described on the help page of the function `dsumSibuya` (for example, `dsSib.log` uses (18) together with the intelligent logarithm as given in (15)).

Additionally, a method `default` is implemented which consists of a careful combination of the above methods based on numerical experiments. Note that all methods involved work with $\log x$ instead of x . For this reason, `polyG` requires as argument $\log x$ rather than x .

Finally, let us mention that the problem of evaluating sums of type

$$\sum_{k=0}^n \binom{n}{k} (-1)^k f_k \quad (24)$$

for sequences $(f_k)_k$ has a long history (note that (18) falls under this setup). They can be interpreted as forward differences and are known to be numerically challenging. Approximate (asymptotic for $n \rightarrow \infty$) formulas may be obtained by using methods from complex analysis; see, for example, [Flajolet and Sedgewick, 1995], and have been important, e.g., for estimating the complexity of computer algorithms.

However, these asymptotic formulas are not very accurate for finite n (note that in our case, $n = d$, the data dimension) and the only known way to accurately compute them, seems high precision arithmetic. See `sumBinomMpfr()` in R package `Rmpfr`, and its documentation for simple examples such as $f_k = f(k) = \sqrt{k}$.

5.4. Kendall's tau for Ali-Mikhail-Haq copulas

In Table 1, the population version of Kendall's tau for the Ali-Mikhail-Haq (A) family, as function of the parameter θ , is

$$\tau_A(\theta) = 1 - 2(\theta + (1 - \theta)^2 \log(1 - \theta)) / (3\theta^2). \tag{25}$$

When computing it for the Kendall's tau estimator (see Section 3.1), however, the simple formula (25) is not sufficient, notably not for small θ , see the left plot in Figure 8: Replacing $\log(1 - \theta)$ by its numerical accurate $\log1p(-\theta)$ helps down to around $\theta \approx 10^{-7}$, but then that formula breaks down as well, and indeed our `tauAMH()` (package `copula`), uses parts of the Taylor series $\tau_A(\theta) = \frac{2}{9}\theta(1 + \theta(\frac{1}{4} + \frac{\theta}{10}(1 + \theta(\frac{1}{2} + \theta\frac{2}{7})))) + O(\theta^6)$ ¹, as soon as $\theta \leq 10^{-2}$.

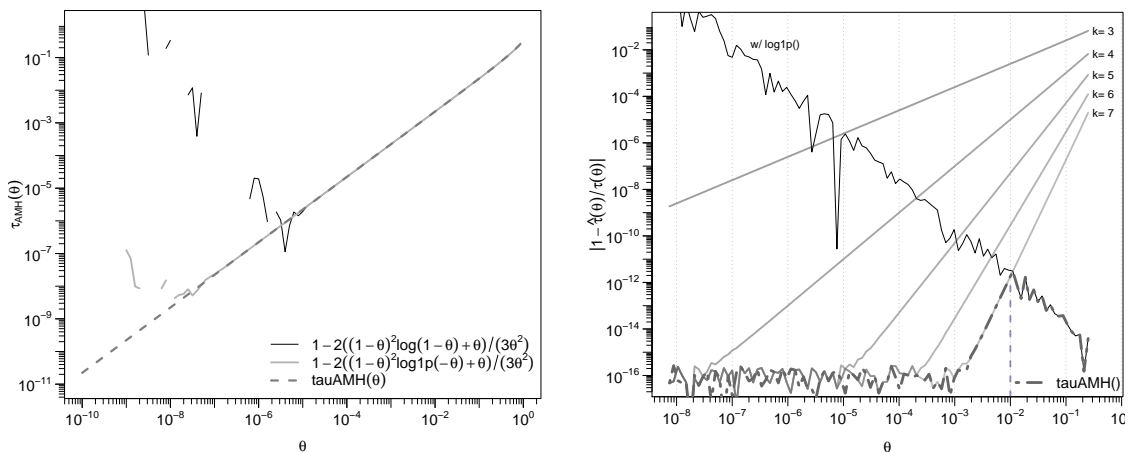


FIGURE 8. left: $\tau_A(\theta)$: direct form (breaking down for $\theta < 10^{-5}$), using `log1p()` (okay down to $\theta \approx 10^{-8}$), correct approximation as provided by `tauAMH()`; right: relative errors of `log1p()`, Taylor approximations, and our hybrid `tauAMH()`.

5.5. log1mexp

There are several situations, such as the one addressed in Section 5.6, where an accurate computation of

$$f(a) = \log1mexp(a) = \log(1 - \exp(-a)), \quad a \geq 0, \tag{26}$$

¹ replacing $\log(1 - \theta)$ by its Taylor expansion $-\sum_{k=1}^{\infty} \theta^k/k$ in (25) results in the expansion $\tau_A(\theta) = \frac{2}{9}\theta \sum_{k=0}^{\infty} \frac{6}{(k+1)(k+2)(k+3)} \theta^k$

is required. Note that this is numerically challenging in both situations, when $a \downarrow 0$ (hence $\exp(-a) \uparrow 1$ and cancellation of two almost equal terms in $1 - \exp(-a)$), and when $a \uparrow \infty$, as $\exp(-a) \downarrow 0$ and in $1 - \exp(-a)$, almost all accuracy of $\exp(-a)$ is lost when it is less than around 10^{-15} . Now, for the first case, we can make use of the **R** and **C** library function `expm1(x)` which computes $\exp(x) - 1$ accurately also for very small x , and for the second case, use the **R** and **C** library function `log1p(x)` which computes $\log(1 + x)$ accurately also for very small x . Our package `copula` provides the function `log1mexp()` which adapts to these two cases, in a sense, optimally by using a cutoff of $a = \log 2$; see [Mächler, 2012].

5.6. The density of the diagonal of Frank copulas

Computing the DMLE for Frank's copula family is one situation where the accurate computation of (26) is crucial. To compute the density δ'_θ of the diagonal (12) for Frank copulas with generator $\psi_\theta(t) = -\log(1 - (1 - \exp(-\theta))e^{-t})/\theta$, the functions

$$-\psi'_\theta(t) = \frac{(1 - e^{-\theta}) \exp(-t)}{\theta(1 - (1 - e^{-\theta}) \exp(-t))},$$

$$\psi_\theta^{-1}(u) = -\log\left(\frac{\exp(-u\theta) - 1}{e^{-\theta} - 1}\right), \quad \text{and } -(\psi_\theta^{-1})'(u) = \frac{\theta}{\exp(\theta u) - 1}$$

are involved. Numerical issues in computing $\delta'_\theta(u)$ arise for large θ and u close to 1. It is known that numerically, the computation of $e^x - 1$ suffers from cancellation when $0 < x \ll 1$. The first suspect is thus ψ_θ^{-1} which involves terms of this type. The left-hand side of Figure 9 displays $\log \delta'_\theta(u)$ for $\theta = 38$ and $d = 2$ for two different versions of computing ψ_θ^{-1} : `psiInv.0` uses only the **R** functions `log()` and `exp()`, whereas `psiInv.1` uses `log()` and `expm1()`. Either way, numerical issues appear due to the cancellation in the division of terms of type $e^x - 1$ when computing ψ_θ^{-1} . By rewriting ψ_θ^{-1} as

$$\psi_\theta^{-1}(u) = -\log\left(1 - \frac{\exp(-u\theta) - e^{-\theta}}{1 - e^{-\theta}}\right)$$

we can use **R**'s function `log1p(.)` to accurately compute $\log(1 + \cdot)$ and thus the generator inverse ψ_θ^{-1} via `-log1p((exp(-u*theta)-exp(-theta))/expm1(-theta))` which we denote by `psiInv.2`. The right-hand side of Figure 9 displays the effect of using `psiInv.2` in comparison to `psiInv.0` and `psiInv.1`.

Although this already looks promising, it is still not possible to compute the negative log-likelihood for the DMLE of Frank's copula family for a large range of parameters θ as one would like to do for the optimization. The left-hand side of Figure 10 shows the negative log-likelihood based on the diagonal of a five-dimensional (so rather low-dimensional) Frank copula, where computations are done in double precision and high-precision arithmetic with different significant bits (this was done with the **R** package `Rmpfr`). As it turns out, the problem is the evaluation of $-\psi'_\theta(t)$ for small t (equivalently, $t = \psi_\theta^{-1}(u)$ for large θ and u close to 1 as before). The solution

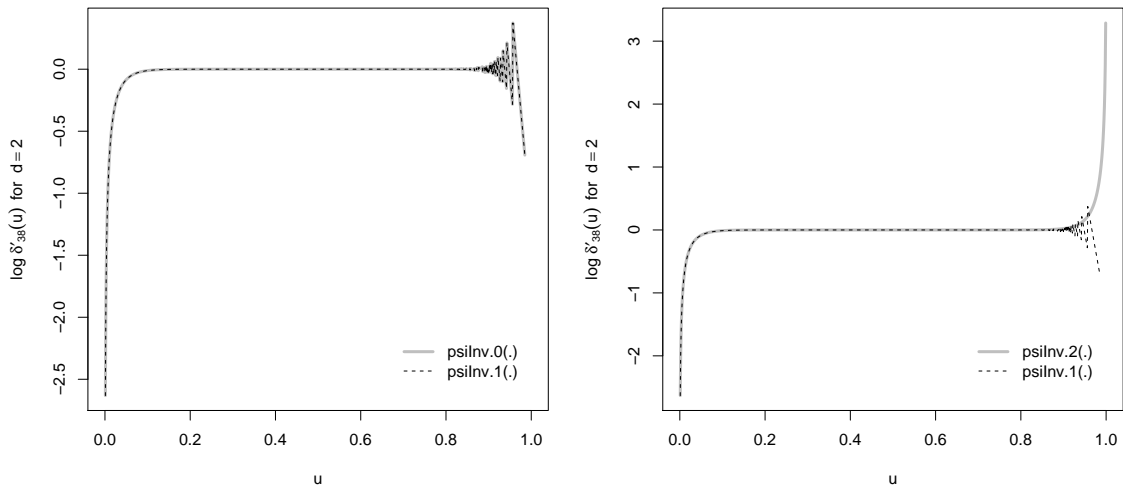


FIGURE 9. $\log \delta'_\theta(u)$ for $\theta = 38$ and $d = 2$ for computing ψ_θ^{-1} via R's `log()` and `exp()` functions (`psiInv.0.`), a version with `log()` and `expm1()` (`psiInv.1.`), and a version using `log1p()` and `expm1()` (`psiInv.2.`).

is to rewrite the logarithm of $-\psi'_\theta(t)$ via

$$\begin{aligned}
 \log(-\psi'_\theta(t)) &= \log(1 - e^{-\theta}) - t - \log(\theta) - \log(1 - (1 - e^{-\theta}) \exp(-t)) \\
 &= \log 1 \text{mexp}(\theta) - t - \log(\theta) - \log 1 \text{mexp}(-\log((1 - e^{-\theta}) \exp(-t))) \\
 &= \log 1 \text{mexp}(\theta) - t - \log(\theta) - \log 1 \text{mexp}(t - \log(1 - e^{-\theta})) \\
 &= w - \log(\theta) - \log 1 \text{mexp}(-w),
 \end{aligned}$$

where $w = \log 1 \text{mexp}(\theta) - t$. By computing `log1mexp` via `log1mexp()` as described in Section 5.5, one can then accurately compute the negative log-likelihood for the DMLE for Frank's copula family; see the right-hand side of Figure 10. For more details we refer the interested reader to [Mächler, 2011]².

6. Conclusion

Motivated by applications in quantitative risk management, we introduced and compared different parametric estimators for Archimedean copula families with focus on large dimensions (up to $d = 100$). In particular, estimators based on Kendall's tau, Blomqvist's beta, minimum distance estimators, the maximum-likelihood estimator, a simulated maximum-likelihood estimator, and a maximum-likelihood estimator based on the copula diagonal were investigated both under known and unknown margins (pseudo-observations). Several of these estimation methods were newly introduced and investigated in this context.

Under known margins, the best performance according to precision was shown by the maximum-likelihood estimator. To our surprise, the maximum-likelihood estimator also performed well according to numerical stability (being of similar numerical stability as the pairwise Kendall's

² As this is a vignette of R package `copula`, all its figures are completely reproducible via R code in the file `Frank-Rmpfr.Rnw` which is part of the package source.

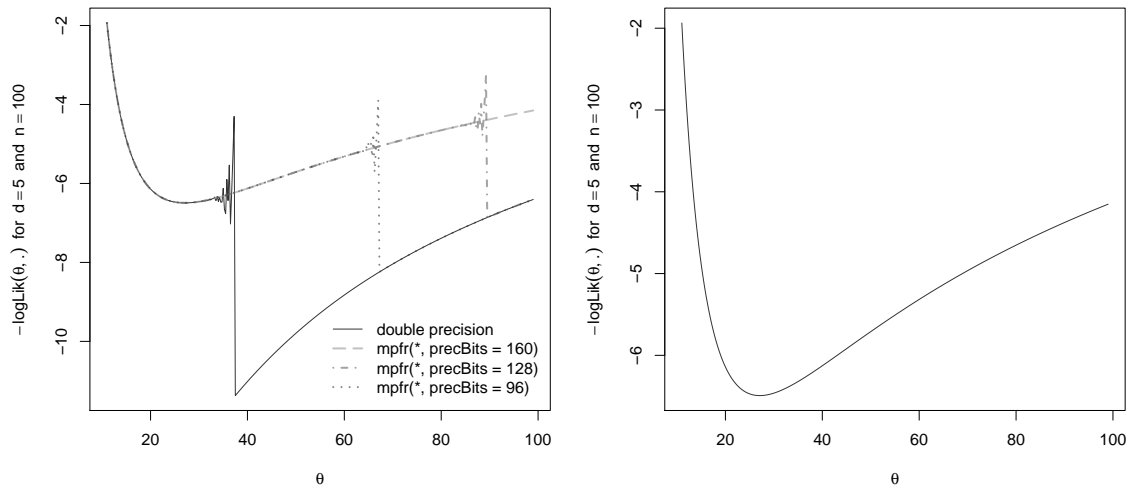


FIGURE 10. Negative log-likelihood based on the diagonal of a five-dimensional Frank copula (sample size $n = 100$) with various kinds of precision (left) and an accurate version in double precision based on $\log 1mexp$ (right).

tau estimators) and run time (being only outperformed by the diagonal maximum likelihood estimator). Under unknown margins, the MLE still performed best, but the differences in precision between the various estimators are much less clear-cut and the rate of improvement in d is not as high as under known margins.

Our work specifically addressed the challenges of inference in large dimensions which is important for practical applications. Large dimensions up to $d = 100$ were tackled for the first time and numerical challenges when working in such large dimensions were addressed in detail. Moreover, a detailed implementation of the presented estimation methods in the R package *copula* creates transparency and allows the reader to access and verify our results.

Archimedean copulas in high dimensions are often used for modeling dependence in large portfolios of various kinds, for example, if tail dependence is of interest or only few data is available. Furthermore, Archimedean copulas serve as building blocks for several extensions which are not limited by exchangeability and thus allow for a wider range of statistical applications. The numerical solutions studied in this paper extend to these models. Furthermore, important statistical quantities used for sampling or goodness-of-fit testing such as distributions of radial parts or the Kendall distribution function require accurate evaluation procedures, which are now available even in high dimensions.

Appendix A: Simulation tables

TABLE 2. Biases (under known margins) multiplied by 1000. The numbers in parentheses denote the factors of the corresponding entries with respect to the performance of the MLE.

τ	Fam.	d	Estimator														
			τ_{β}	τ_{γ}	τ_{δ}	β	MDExCM	MDExS	MDExM	MDExS	MLE	SMLE	DMLE				
0.25	A	5	-4.3 (1.6)	-22.9 (8.7)	-10.1 (3.9)	-12.2 (4.6)	-7.4 (2.8)	-54.3 (20.6)	-824.7 (313.7)	-2.6 (1.0)	-7.0 (2.7)	-35.8 (13.6)					
		20	-5.5 (-15.2)	-20.2 (-55.5)	-42.9 (-117.7)	-1.9 (-5.3)	-3.8 (-10.4)	-38.3 (-105.0)	-831.0 (-2277.7)	0.4 (1.0)	-0.4 (-1.0)	-27.9 (-76.5)					
		100	0.9 (-9.5)	-21.9 (-241.5)	171.3 (-1888.9)	-0.0 (0.5)	0.1 (-1.0)	-39.2 (-432.7)	-827.3 (9122.9)	-0.1 (1.0)	-0.6 (-6.7)	-26.5 (-292.4)					
	C	5	4.4 (1.2)	18.9 (5.3)	4.9 (1.4)	-2.1 (-0.6)	-185.7 (-51.5)	4.6 (1.3)	-666.0 (-184.7)	3.6 (1.0)	256.2 (71.1)	-0.6 (-0.2)					
		20	3.1 (2.1)	19.7 (13.3)	-18.6 (-12.6)	-258.8 (-91.9)	-258.8 (-175.4)	-12.1 (-8.2)	-663.3 (-449.5)	1.5 (1.0)	526.0 (356.4)	1.8 (1.2)					
		100	-1.3 (321.1)	24.1 (-6023.6)	37143.5 (-9284172.1)	-418.9 (104693.9)	-351.5 (87871.1)	-21.6 (5406.7)	-661.9 (165436.2)	-0.0 (1.0)	742.1 (-185481.5)	-4.6 (1146.6)					
	F	5	0.3 (0.0)	35.1 (4.6)	8.2 (1.1)	0.7 (0.1)	17.9 (2.3)	6.3 (0.8)	-2363.5 (-308.0)	7.7 (1.0)	5.3 (0.7)	-32.5 (-4.2)					
		20	10.9 (12.2)	36.5 (40.9)	-14.8 (-40.9)	3.8 (4.3)	10.1 (11.3)	9.6 (10.7)	-2357.3 (-2642.6)	0.9 (1.0)	-7.5 (-8.4)	-127.2 (-142.6)					
		100	2.3 (-3.4)	27.1 (-39.6)	60976.9 (-89236.7)	-5.3 (7.8)	3.7 (-5.3)	-38.6 (56.5)	-2356.9 (3449.2)	-0.7 (1.0)	3.3 (-4.9)	-208.3 (304.8)					
	G	5	5.2 (2.4)	10.7 (4.9)	8.4 (3.8)	1.6 (0.7)	0.1 (0.1)	12.8 (5.8)	-331.4 (-150.7)	2.2 (1.0)	7.7 (3.5)	1.6 (0.7)					
20		2.9 (-1.7)	10.3 (-6.2)	2.4 (-1.4)	0.3 (-0.2)	-0.2 (0.1)	7.2 (-4.3)	-326.2 (194.4)	-1.7 (1.0)	2.0 (-1.2)	0.3 (-0.2)						
100		4.1 (12.0)	10.2 (29.7)	-33.3 (-96.9)	-0.2 (-0.6)	-0.1 (-0.2)	10.8 (31.4)	-324.7 (-944.9)	0.3 (1.0)	0.9 (2.5)	-0.9 (-2.7)						
J	5	9.0 (10.5)	17.9 (20.8)	4.6 (5.3)	4.9 (5.7)	3.9 (4.5)	28.8 (33.5)	-594.0 (-689.9)	0.9 (1.0)	17.8 (20.6)	13.6 (15.8)						
	20	9.8 (10.2)	19.6 (20.4)	18.1 (18.9)	-0.7 (-0.8)	1.9 (2.0)	44.2 (46.1)	-591.9 (-617.7)	1.0 (1.0)	9.9 (10.4)	4.4 (4.6)						
	100	2.8 (8.2)	20.1 (58.9)	33980.2 (99368.1)	0.4 (1.2)	0.2 (0.7)	60.2 (175.9)	-594.9 (-1739.7)	0.3 (1.0)	-0.0 (-0.0)	0.7 (2.2)						
0.75	C	5	74.6 (6.8)	164.8 (15.1)	197.1 (18.1)	-3988.0 (-365.3)	-4140.0 (-379.3)	-1632.3 (-149.5)	-6000.0 (-549.7)	10.9 (1.0)	119.7 (11.0)	719.9 (66.0)					
		20	118.9 (-164.8)	145.1 (-201.1)	176.7 (-245.0)	-5849.9 (8109.5)	-5521.4 (7654.1)	-1714.1 (2376.2)	-5999.8 (8317.5)	-0.7 (1.0)	99.1 (-137.4)	281.3 (-389.9)					
		100	24.1 (20.4)	106.0 (89.8)	31810.2 (26967.5)	-5976.2 (-5066.4)	-5946.2 (-5041.0)	-1650.7 (-1599.4)	-5996.4 (-5083.5)	1.2 (1.0)	61.2 (51.8)	161.5 (136.9)					
	F	5	82.2 (5.3)	138.2 (8.9)	532.9 (34.3)	25.4 (1.6)	-3.9 (-0.3)	161.0 (10.4)	-14089.0 (-906.6)	15.5 (1.0)	195.6 (12.6)	643.6 (41.4)					
		20	-0.7 (-0.0)	201.5 (11.3)	260.1 (14.6)	9.6 (0.5)	-7.6 (-0.4)	95.8 (5.4)	-14043.0 (-786.5)	17.9 (1.0)	77.9 (4.4)	-58.3 (-3.3)					
		100	31.9 (-6.0)	173.8 (-32.8)	49210.4 (-9293.0)	-2.3 (0.4)	8.2 (-1.6)	37.8 (-7.1)	-14057.7 (2654.7)	-5.3 (1.0)	24.1 (-4.6)	-194.7 (36.8)					
	G	5	57.0 (40.8)	42.2 (30.3)	140.8 (100.9)	3.7 (2.7)	-0.3 (-0.2)	27.3 (19.5)	-2998.7 (-2148.7)	1.4 (1.0)	-15.3 (-11.0)	199.0 (142.6)					
		20	16.1 (-9.9)	68.5 (-42.2)	52.3 (-32.2)	1.5 (-0.9)	2.1 (-1.3)	40.3 (-24.8)	-2996.6 (1843.4)	-1.6 (1.0)	-11.4 (7.0)	53.6 (-33.0)					
		100	19.6 (-14.8)	57.5 (-43.5)	-2999.9 (2268.3)	2.0 (-1.5)	-0.2 (0.2)	30.2 (-22.8)	-2997.8 (2266.6)	-1.3 (1.0)	-9.9 (7.5)	3.4 (-2.6)					
	J	5	24.0 (1.3)	123.7 (6.7)	236.8 (12.9)	15.2 (0.8)	-0.6 (-0.0)	105.2 (5.7)	-5781.1 (-3144.9)	18.4 (1.0)	-27.2 (-1.5)	219.0 (11.9)					
20		86.0 (15.9)	133.4 (24.7)	82.0 (15.2)	4.4 (0.8)	6.8 (1.3)	98.6 (18.3)	-5776.6 (-1070.5)	5.4 (1.0)	-0.4 (-0.1)	-0.1 (-0.0)						
100		109.8 (33.0)	121.0 (36.3)	28759.0 (8635.4)	0.2 (0.1)	0.2 (0.1)	63.5 (19.1)	-5782.3 (-1736.2)	3.3 (1.0)	8.8 (2.7)	-18.1 (-5.4)						

TABLE 3. RMSE (under known margins) multiplied by 1000. The numbers in parentheses denote the factors of the corresponding entries with respect to the performance of the MLE.

τ	Family	d	Estimator											
			$\tau_{\hat{\epsilon}}$	$\tau_{\hat{\theta}}$	β	$MDE_{\hat{\alpha}}^{CM}$	$MDE_{\hat{\alpha}}^{KS}$	$MDE_{\hat{\alpha}}^{CM}$	$MDE_{\hat{\alpha}}^{KS}$	MLE	SMLE	DMLE		
0.25	A	5	77.2 (1.7)	81.5 (1.8)	125.3 (2.7)	80.8 (1.8)	80.4 (1.8)	213.2 (4.6)	831.5 (18.1)	45.9 (1.0)	48.2 (1.0)	115.7 (2.5)		
		20	51.5 (3.1)	54.4 (3.3)	160.1 (9.7)	35.3 (2.3)	37.5 (2.3)	167.7 (10.2)	834.3 (50.7)	16.5 (1.0)	17.5 (1.1)	84.3 (5.1)		
		100	45.9 (10.0)	49.4 (10.8)	171.3 (37.4)	15.5 (3.4)	16.1 (3.5)	158.7 (34.6)	832.6 (181.6)	4.6 (1.0)	5.1 (1.1)	80.6 (17.6)		
	C	5	138.6 (1.8)	139.7 (1.8)	168.3 (2.2)	120.6 (1.6)	367.5 (4.8)	281.6 (3.7)	666.3 (8.7)	76.8 (1.0)	2127.0 (27.7)	217.4 (2.8)		
		20	100.1 (2.9)	100.6 (3.0)	172.3 (5.1)	306.1 (9.0)	416.3 (12.3)	262.8 (7.7)	666.0 (19.6)	33.9 (1.0)	2881.6 (84.9)	161.2 (4.7)		
		100	89.0 (6.1)	97.7 (6.6)	37143.5 (2525.6)	528.5 (35.9)	484.3 (32.9)	224.8 (15.3)	663.7 (45.1)	14.7 (1.0)	3563.2 (242.3)	147.1 (10.0)		
	F	5	349.5 (1.2)	350.8 (1.3)	497.6 (1.8)	398.4 (1.4)	406.4 (1.5)	1028.9 (3.7)	2368.7 (8.5)	279.6 (1.0)	292.2 (1.0)	623.4 (2.2)		
		20	251.5 (2.0)	251.6 (2.0)	538.4 (4.3)	185.1 (1.5)	191.4 (1.5)	961.1 (7.7)	2364.4 (18.9)	125.3 (1.0)	129.4 (1.0)	423.4 (3.4)		
		100	217.4 (5.0)	218.3 (5.0)	60976.9 (1399.6)	86.7 (2.0)	82.0 (1.9)	903.3 (20.7)	2364.0 (54.3)	43.6 (1.0)	50.2 (1.2)	398.9 (9.2)		
	G	5	71.0 (1.5)	73.2 (1.5)	96.7 (2.0)	59.0 (1.2)	57.6 (1.2)	165.7 (3.5)	332.7 (7.0)	47.7 (1.0)	50.4 (1.1)	109.2 (2.3)		
		20	57.8 (2.5)	62.0 (2.7)	110.1 (4.7)	26.8 (1.2)	28.0 (1.2)	150.6 (6.5)	335.3 (14.5)	23.2 (1.0)	23.8 (1.0)	59.1 (2.6)		
		100	57.9 (5.6)	57.7 (5.6)	333.3 (32.3)	12.6 (1.2)	12.5 (1.2)	145.8 (14.1)	338.2 (32.8)	10.3 (1.0)	12.3 (1.2)	37.6 (3.6)		
J	5	139.5 (1.7)	146.9 (1.8)	165.5 (2.0)	98.9 (1.2)	104.7 (1.3)	286.8 (3.5)	596.5 (7.3)	81.6 (1.0)	90.5 (1.1)	185.2 (2.3)			
	20	132.2 (3.6)	126.4 (3.4)	214.4 (5.8)	46.1 (1.2)	46.4 (1.2)	306.7 (8.3)	597.0 (16.1)	37.2 (1.0)	47.9 (1.3)	92.3 (2.5)			
	100	124.2 (8.2)	126.7 (8.3)	33980.2 (2237.0)	21.4 (1.4)	21.6 (1.4)	352.1 (23.2)	595.4 (39.2)	15.2 (1.0)	22.5 (1.5)	53.7 (3.5)			
0.75	C	5	871.6 (3.1)	869.8 (3.1)	1456.9 (5.1)	4897.1 (17.2)	4983.8 (17.5)	3234.4 (11.3)	6000.0 (21.0)	285.2 (1.0)	1430.2 (5.0)	3545.5 (12.4)		
		20	770.1 (6.0)	773.6 (6.0)	1040.2 (8.1)	5924.6 (46.0)	5755.2 (44.7)	3296.7 (25.6)	5999.8 (46.6)	128.7 (1.0)	1284.5 (10.0)	1706.7 (13.3)		
		100	698.6 (12.4)	722.2 (12.8)	31810.2 (563.2)	5988.0 (106.0)	5972.9 (105.7)	3221.1 (57.0)	5997.2 (106.2)	56.5 (1.0)	1019.0 (18.0)	1172.5 (20.8)		
F	5	1014.1 (1.5)	1005.7 (1.5)	3232.6 (4.8)	766.6 (1.1)	807.3 (1.2)	2124.4 (3.2)	14095.8 (21.1)	668.6 (1.0)	793.1 (1.2)	4105.2 (6.1)			
	20	702.8 (2.4)	760.2 (2.6)	2169.0 (7.3)	354.9 (1.2)	361.7 (1.2)	2081.9 (7.0)	14067.2 (47.3)	297.2 (1.0)	347.8 (1.2)	2023.9 (6.8)			
	100	639.0 (4.9)	692.9 (5.3)	49210.4 (380.0)	158.2 (1.2)	160.9 (1.2)	1943.9 (15.0)	14071.0 (108.6)	129.5 (1.0)	147.7 (1.1)	1273.0 (9.8)			
G	5	385.1 (2.4)	383.6 (2.4)	763.5 (4.8)	173.0 (1.1)	177.2 (1.1)	462.3 (2.9)	2998.8 (18.8)	159.7 (1.0)	185.5 (1.2)	1238.1 (7.8)			
	20	339.4 (4.7)	337.0 (4.7)	509.8 (7.1)	80.8 (1.1)	83.3 (1.1)	471.9 (6.5)	2997.0 (41.5)	72.2 (1.0)	105.1 (1.5)	568.8 (7.9)			
	100	327.3 (10.2)	339.3 (10.6)	2999.9 (93.9)	37.4 (1.2)	38.5 (1.2)	452.8 (14.2)	2998.2 (93.9)	31.9 (1.0)	77.8 (2.4)	353.3 (11.1)			
J	5	853.2 (2.9)	851.6 (2.8)	1366.3 (4.6)	341.0 (1.1)	341.5 (1.1)	947.5 (3.2)	5781.2 (19.3)	299.2 (1.0)	369.0 (1.2)	1821.4 (6.1)			
	20	785.7 (6.0)	823.0 (6.3)	874.6 (6.7)	156.9 (1.2)	158.3 (1.2)	910.4 (7.0)	5777.9 (44.2)	130.6 (1.0)	185.3 (1.4)	757.8 (5.8)			
	100	751.6 (12.7)	784.0 (13.2)	28780.2 (485.8)	73.5 (1.2)	72.6 (1.2)	922.3 (15.6)	5782.3 (97.6)	59.2 (1.0)	127.3 (2.1)	445.9 (7.5)			

TABLE 4. Mean user run times (under known margins) in milliseconds. The numbers in parentheses denote the factors of the corresponding entries with respect to the performance of the MLE.

τ	Family	d	Estimator									
			$\tau_{\hat{\tau}}$	$\tau_{\hat{\theta}}$	β	MDE $_{\hat{\tau}}^{\text{CM}}$	MDE $_{\hat{\tau}}^{\text{KS}}$	MDE $_{\hat{\tau}}^{\text{CM}}$	MDE $_{\hat{\tau}}^{\text{KS}}$	MLE	SMLE	DMLE
0.25	A	5	6.4 (0.3)	10.2 (0.5)	3.1 (0.2)	64.7 (3.3)	90.1 (4.6)	70.1 (3.6)	96.5 (4.9)	19.5 (1.0)	3950.4 (202.1)	3.4 (0.2)
		20	58.6 (2.3)	142.5 (5.6)	3.8 (0.1)	277.5 (11.0)	389.7 (15.4)	299.1 (11.8)	416.0 (16.5)	25.2 (1.0)	3598.9 (142.6)	3.8 (0.2)
		100	1340.6 (19.1)	3498.9 (49.7)	6.8 (0.1)	2511.6 (35.7)	3534.4 (50.2)	3272.5 (46.5)	4743.4 (67.4)	70.4 (1.0)	4202.8 (59.7)	4.7 (0.1)
	C	5	5.7 (0.3)	5.7 (0.3)	2.3 (0.1)	225.4 (9.9)	241.3 (10.6)	212.0 (9.3)	157.1 (6.9)	22.8 (1.0)	10362.1 (454.7)	2.8 (0.1)
		20	58.2 (2.0)	58.2 (2.0)	2.8 (0.1)	958.9 (33.1)	887.9 (30.6)	963.9 (33.3)	666.7 (23.0)	29.0 (1.0)	10179.1 (351.2)	2.4 (0.1)
		100	1342.1 (18.5)	1344.8 (18.5)	6.6 (0.1)	8379.0 (115.4)	8890.5 (122.4)	10115.4 (139.3)	6551.3 (90.2)	72.6 (1.0)	10425.8 (143.6)	3.3 (0.0)
	F	5	7.3 (0.3)	13.8 (0.5)	5.0 (0.2)	172.2 (6.6)	201.3 (7.8)	180.2 (6.9)	196.9 (7.6)	26.0 (1.0)	11106.5 (427.7)	11.7 (0.5)
		20	59.2 (1.4)	218.8 (5.1)	4.9 (0.1)	777.9 (18.2)	919.1 (21.5)	729.7 (17.0)	883.8 (20.6)	42.8 (1.0)	11860.2 (277.0)	12.1 (0.3)
		100	1346.3 (10.9)	5289.8 (42.8)	10.0 (0.1)	7247.7 (58.7)	8489.3 (68.7)	7266.4 (58.8)	8234.5 (66.6)	123.6 (1.0)	11279.1 (91.3)	12.0 (0.1)
	G	5	6.0 (0.1)	5.9 (0.1)	2.4 (0.0)	45.0 (0.6)	79.1 (1.0)	52.3 (0.7)	85.8 (1.1)	78.4 (1.0)	10081.0 (128.5)	1.4 (0.0)
20		58.5 (0.5)	58.6 (0.5)	2.8 (0.0)	226.7 (1.9)	314.2 (2.6)	225.3 (1.9)	423.3 (3.5)	119.8 (1.0)	9596.7 (80.1)	1.8 (0.0)	
100		1336.6 (3.6)	1341.9 (3.6)	4.4 (0.0)	2657.2 (7.1)	3701.2 (9.9)	2617.7 (7.0)	4376.9 (11.7)	373.7 (1.0)	9739.1 (26.1)	2.5 (0.0)	
J	5	7.2 (0.2)	16.5 (0.4)	4.6 (0.1)	48.5 (1.2)	85.0 (2.1)	56.8 (1.4)	92.1 (2.2)	41.3 (1.0)	11156.2 (270.1)	6.3 (0.2)	
	20	59.6 (1.0)	331.9 (5.5)	4.8 (0.1)	237.7 (4.0)	311.1 (5.2)	245.2 (4.1)	402.1 (6.7)	59.9 (1.0)	11234.0 (187.4)	8.0 (0.1)	
	100	1377.6 (9.1)	8047.2 (53.1)	10.0 (0.1)	2198.1 (14.5)	3486.4 (23.0)	2802.5 (18.5)	4158.8 (27.5)	151.5 (1.0)	9975.0 (65.9)	9.0 (0.1)	
0.75	C	5	5.6 (0.3)	5.8 (0.3)	2.8 (0.2)	175.7 (10.1)	177.0 (10.2)	216.9 (12.5)	158.3 (9.1)	17.3 (1.0)	10301.2 (593.8)	2.6 (0.1)
		20	56.0 (2.2)	56.5 (2.2)	3.0 (0.1)	636.4 (25.3)	674.6 (26.8)	940.4 (37.3)	630.2 (25.0)	25.2 (1.0)	10317.8 (409.4)	2.4 (0.1)
		100	1275.8 (18.0)	1280.0 (18.1)	7.0 (0.1)	7541.1 (106.4)	6855.5 (96.8)	9827.8 (138.7)	7086.2 (100.0)	70.8 (1.0)	9795.6 (138.2)	3.4 (0.0)
	F	5	6.7 (0.3)	16.2 (0.7)	4.9 (0.2)	201.8 (8.5)	203.6 (8.5)	210.8 (8.8)	217.1 (9.1)	22.9 (1.0)	11028.6 (462.4)	10.9 (0.5)
		20	56.9 (1.6)	247.0 (6.8)	5.2 (0.1)	847.6 (23.2)	905.7 (24.8)	841.4 (23.1)	796.7 (21.8)	36.5 (1.0)	11474.2 (314.6)	11.4 (0.3)
		100	1306.4 (15.2)	6880.2 (79.9)	9.4 (0.1)	8067.5 (93.7)	8728.4 (101.4)	8501.9 (98.8)	8625.5 (100.2)	86.1 (1.0)	10596.2 (123.1)	12.8 (0.1)
	G	5	5.7 (0.1)	5.8 (0.1)	2.5 (0.0)	60.0 (1.1)	74.9 (1.4)	52.0 (1.0)	90.2 (1.7)	53.1 (1.0)	9570.5 (180.3)	1.4 (0.0)
		20	56.0 (0.6)	56.4 (0.6)	2.9 (0.0)	248.6 (2.8)	313.8 (3.5)	267.7 (3.0)	401.0 (4.5)	88.5 (1.0)	9368.8 (105.8)	1.7 (0.0)
		100	1279.3 (4.4)	1275.5 (4.4)	4.7 (0.0)	2924.9 (10.0)	3379.0 (11.5)	2705.9 (9.2)	4126.4 (14.1)	297.6 (1.0)	9533.3 (32.6)	2.4 (0.0)
	J	5	7.3 (0.2)	18.4 (0.6)	4.8 (0.2)	64.6 (2.0)	97.7 (3.1)	66.6 (2.1)	103.1 (3.3)	31.7 (1.0)	10136.8 (320.2)	5.5 (0.2)
20		57.4 (1.3)	357.9 (8.3)	6.2 (0.1)	308.9 (7.2)	371.5 (8.6)	289.5 (6.7)	431.1 (10.0)	43.0 (1.0)	10660.1 (247.7)	7.7 (0.2)	
100		1279.2 (9.9)	8342.0 (64.9)	9.9 (0.1)	2921.9 (22.7)	4094.5 (31.8)	3137.8 (24.4)	4657.5 (36.2)	128.6 (1.0)	9835.1 (76.5)	8.9 (0.1)	

TABLE 5. Biases (under unknown margins) multiplied by 1000. The numbers in parentheses denote the factors of the corresponding entries with respect to the performance of the MLE.

τ	Fam.	d	Estimator																			
			τ_{ξ}	τ_{θ}	β	MDE_{χ}^{CVM}	MDE_{χ}^{KS}	MDE_{χ}^{VM}	MDE_{χ}^{KS}	MLE	SMLE	DMLE										
0.25	A	5	-4.3	(-1.6)	-22.9	(-8.4)	-13.8	(-5.1)	34.9	(12.8)	41.8	(15.3)	10.4	(3.8)	-828.8	(-303.2)	2.7	(1.0)	-1.3	(-0.5)	-33.7	(-12.3)
		20	-1.9	(-0.6)	-20.1	(-6.1)	-64.5	(-19.6)	48.5	(14.7)	48.1	(14.6)	9.2	(2.8)	-826.7	(-251.1)	3.3	(1.0)	4.5	(1.4)	-32.4	(-9.8)
		100	-0.5	(0.1)	-22.3	(3.6)	125.2	(-20.0)	50.6	(-8.1)	53.1	(-8.5)	6.1	(-1.0)	-825.7	(131.6)	-6.3	(1.0)	-5.8	(0.9)	-37.9	(6.0)
	C	5	4.4	(0.1)	18.9	(0.4)	1.5	(0.0)	84.2	(1.7)	82.8	(1.7)	61.8	(1.3)	-657.5	(-13.5)	48.8	(1.0)	58.0	(1.2)	-33.4	(-0.7)
		20	5.3	(0.1)	24.1	(0.5)	-24.7	(-0.6)	87.5	(2.0)	92.5	(2.1)	49.9	(1.1)	-633.3	(-14.2)	44.6	(1.0)	42.3	(0.9)	-44.6	(-1.0)
		100	9.9	(0.2)	20.3	(0.4)	37143.5	(779.0)	85.9	(1.8)	84.3	(1.8)	20.6	(0.4)	-638.4	(-13.4)	47.7	(1.0)	43.1	(0.9)	-55.0	(-1.2)
	F	5	0.3	(0.0)	35.1	(1.5)	-17.1	(-0.7)	234.6	(10.1)	258.9	(11.1)	173.6	(7.5)	-2358.6	(-101.5)	23.2	(1.0)	31.4	(1.4)	-98.4	(-4.2)
		20	1.3	(0.0)	29.9	(1.1)	-77.1	(-2.9)	285.2	(10.7)	290.9	(10.9)	231.8	(8.7)	-2345.6	(-87.9)	26.7	(1.0)	12.9	(0.5)	-174.2	(-6.5)
		100	17.0	(-2.3)	39.6	(-5.4)	60976.9	(-8308.9)	326.8	(-44.5)	310.1	(-42.3)	252.8	(-34.4)	-2367.5	(322.6)	-7.3	(1.0)	2.6	(-0.4)	-163.0	(22.2)
	G	5	5.2	(0.2)	10.7	(0.4)	3.1	(0.1)	39.1	(1.3)	37.0	(1.2)	33.4	(1.1)	-332.0	(-10.9)	30.4	(1.0)	38.1	(1.3)	14.1	(0.5)
		20	1.7	(0.1)	10.1	(0.3)	-0.9	(-0.0)	40.2	(1.2)	42.9	(1.3)	43.5	(1.3)	-329.1	(-9.8)	33.5	(1.0)	37.8	(1.1)	24.5	(0.7)
		100	1.8	(0.0)	9.9	(0.2)	-333.3	(-7.9)	43.1	(1.0)	45.7	(1.1)	54.4	(1.3)	-318.0	(-7.6)	42.0	(1.0)	43.3	(1.0)	60.2	(1.4)
J	5	7.2	(0.2)	20.8	(0.5)	7.7	(0.2)	67.9	(1.6)	62.6	(1.5)	69.7	(1.6)	-592.7	(-13.8)	43.1	(1.0)	50.1	(1.2)	32.8	(0.8)	
	20	5.7	(0.2)	18.9	(0.5)	4.0	(0.1)	63.3	(1.8)	68.6	(2.0)	93.4	(2.7)	-594.1	(-17.1)	34.7	(1.0)	37.0	(1.1)	39.7	(1.1)	
	100	-0.8	(-0.0)	21.1	(0.8)	33980.2	(1278.4)	71.0	(2.7)	68.9	(2.6)	104.8	(3.9)	-593.1	(-22.3)	26.6	(1.0)	24.9	(0.9)	67.1	(2.5)	
0.75	C	5	74.6	(-0.2)	164.8	(-0.5)	65.6	(-0.2)	-9.3	(0.0)	-14.0	(0.0)	-170.4	(0.5)	-5959.7	(17.4)	-343.0	(1.0)	-310.4	(0.9)	-358.3	(1.0)
		20	74.4	(-0.1)	121.8	(-0.2)	-121.2	(0.2)	-88.6	(0.2)	-104.0	(0.2)	-342.3	(0.6)	-5942.6	(11.1)	-536.3	(1.0)	-535.6	(1.0)	-681.6	(1.3)
		100	37.8	(-0.1)	127.7	(-0.2)	31810.2	(-54.2)	-44.6	(0.1)	-98.5	(0.2)	-591.6	(1.0)	-5931.5	(10.1)	-587.4	(1.0)	-614.7	(1.0)	-708.5	(1.2)
F	5	82.2	(-0.2)	138.2	(-0.4)	81.6	(-0.2)	34.4	(-0.1)	39.2	(-0.1)	-115.8	(0.3)	-14115.6	(39.2)	-360.5	(1.0)	-292.8	(0.8)	-883.0	(2.4)	
	20	40.3	(-0.1)	171.8	(-0.3)	-175.5	(0.3)	-42.7	(0.1)	-18.6	(0.0)	-373.4	(0.7)	-14052.1	(25.4)	-554.3	(1.0)	-469.4	(0.8)	-695.1	(1.3)	
	100	-0.8	(0.0)	147.8	(-0.3)	49210.4	(-87.5)	-158.8	(0.3)	-129.6	(0.2)	-546.4	(1.0)	-14085.0	(25.0)	-562.6	(1.0)	-533.3	(0.9)	-537.9	(1.0)	
G	5	16.0	(-0.4)	52.1	(-1.2)	51.5	(-1.2)	-7.8	(0.2)	4.7	(-0.1)	-21.2	(0.5)	-2997.9	(70.0)	-42.8	(1.0)	-74.0	(1.7)	-85.2	(2.0)	
	20	30.5	(-0.3)	67.4	(-0.6)	-6.8	(0.1)	-36.3	(0.3)	-51.7	(0.5)	-107.9	(1.0)	-2994.4	(28.4)	-105.4	(1.0)	-103.6	(1.0)	-1.9	(0.0)	
	100	32.5	(-0.2)	48.0	(-0.4)	-2999.9	(22.2)	-66.9	(0.5)	-69.2	(0.5)	-174.5	(1.3)	-2994.4	(22.1)	-135.4	(1.0)	-127.6	(0.9)	42.1	(-0.3)	
J	5	83.3	(-0.3)	106.6	(-0.4)	166.7	(-0.6)	-40.2	(0.1)	-58.0	(0.2)	-89.0	(0.3)	-5779.8	(20.0)	-289.6	(1.0)	-290.1	(1.0)	7.6	(-0.0)	
	20	40.4	(-0.1)	122.1	(-0.3)	15.9	(-0.0)	-117.4	(0.3)	-117.1	(0.3)	-234.0	(0.5)	-5781.6	(13.5)	-428.3	(1.0)	-421.0	(1.0)	19.7	(-0.0)	
	100	98.2	(-0.2)	171.2	(-0.3)	28514.8	(-53.6)	-74.0	(0.1)	-119.7	(0.2)	-357.3	(0.7)	-5782.2	(10.9)	-532.0	(1.0)	-487.3	(0.9)	91.5	(-0.2)	

TABLE 6. RMSE (under unknown margins) multiplied by 1000. The numbers in parentheses denote the factors of the corresponding entries with respect to the performance of the MLE.

τ	Family	d	1000 RMSE																			
			τ_f	τ_g	β	MD τ τ	MD τ β	MD τ β	MD τ τ	MD τ β	MLE	SMLE	DMLE									
0.25	A	5	77.2	(1.5)	81.5	(1.5)	127.9	(2.4)	70.7	(1.3)	74.6	(1.4)	101.4	(1.9)	833.8	(15.7)	53.0	(1.0)	53.0	(1.0)	93.7	(1.8)
		20	52.3	(1.9)	54.3	(2.0)	185.6	(6.7)	60.8	(2.2)	59.7	(2.2)	832.2	(30.2)	27.6	(1.0)	29.3	(1.1)	29.3	(1.1)	83.5	(3.0)
		100	44.5	(2.0)	49.2	(2.2)	125.2	(5.6)	58.5	(2.6)	60.1	(2.7)	93.1	(4.2)	831.3	(37.4)	22.2	(1.0)	24.7	(1.1)	75.8	(3.4)
		5	138.6	(1.0)	139.7	(1.0)	180.0	(1.3)	163.9	(1.2)	160.9	(1.2)	207.0	(1.5)	665.6	(4.9)	136.9	(1.0)	149.8	(1.1)	160.0	(1.2)
	C	20	98.1	(0.9)	105.9	(0.9)	172.5	(1.5)	133.8	(1.2)	135.4	(1.2)	184.4	(1.6)	659.2	(5.7)	115.0	(1.0)	112.8	(1.0)	137.4	(1.2)
		100	98.2	(0.9)	96.5	(0.8)	37143.5	(324.7)	126.7	(1.1)	124.7	(1.1)	178.8	(1.6)	655.0	(5.7)	114.4	(1.0)	110.7	(1.0)	128.1	(1.1)
		5	349.5	(1.1)	350.8	(1.1)	494.9	(1.5)	423.6	(1.3)	444.1	(1.4)	603.3	(1.8)	2366.1	(7.2)	328.4	(1.0)	332.8	(1.0)	487.3	(1.5)
		20	243.9	(1.1)	253.2	(1.1)	486.0	(2.1)	366.2	(1.6)	374.3	(1.6)	589.9	(2.6)	2358.4	(10.2)	230.4	(1.0)	235.8	(1.0)	411.1	(1.8)
	F	100	215.0	(1.0)	222.4	(1.0)	60976.9	(276.4)	385.4	(1.7)	372.8	(1.7)	705.9	(3.2)	2369.2	(10.7)	220.6	(1.0)	219.5	(1.0)	359.0	(1.6)
		5	71.0	(0.9)	73.2	(0.9)	96.6	(1.2)	89.2	(1.1)	85.8	(1.0)	113.3	(1.4)	332.9	(4.0)	82.9	(1.0)	83.6	(1.0)	88.2	(1.1)
		20	61.7	(0.9)	60.5	(0.9)	98.9	(1.4)	76.9	(1.1)	78.8	(1.1)	117.5	(1.7)	336.3	(4.9)	68.8	(1.0)	73.1	(1.1)	84.1	(1.2)
		100	55.3	(0.8)	60.2	(0.8)	333.3	(4.6)	75.8	(1.0)	77.1	(1.1)	132.4	(1.8)	350.3	(4.8)	72.8	(1.0)	74.4	(1.0)	108.6	(1.5)
G	5	137.4	(0.9)	143.5	(1.0)	178.1	(1.2)	165.6	(1.1)	166.8	(1.1)	202.9	(1.3)	595.1	(3.9)	150.8	(1.0)	159.4	(1.1)	162.4	(1.1)	
	20	126.0	(0.9)	129.2	(0.9)	179.2	(1.3)	149.8	(1.1)	152.3	(1.1)	228.2	(1.7)	595.4	(4.4)	136.7	(1.0)	138.2	(1.0)	157.0	(1.1)	
	100	119.6	(1.0)	126.3	(1.1)	33980.2	(286.8)	152.5	(1.3)	149.0	(1.3)	270.0	(2.3)	596.7	(5.0)	118.5	(1.0)	123.4	(1.0)	174.6	(1.5)	
	5	871.6	(1.1)	859.8	(1.1)	1468.2	(1.8)	739.8	(0.9)	756.9	(0.9)	868.2	(1.1)	5968.9	(7.4)	804.3	(1.0)	764.2	(1.0)	1101.4	(1.4)	
C	20	754.7	(0.9)	788.9	(1.0)	1043.1	(1.3)	658.6	(0.8)	651.6	(0.8)	897.7	(1.1)	5953.8	(7.2)	826.4	(1.0)	842.4	(1.0)	1043.6	(1.3)	
	100	744.6	(0.9)	723.6	(0.9)	31810.2	(37.8)	660.3	(0.8)	672.4	(0.8)	1038.5	(1.2)	5943.0	(7.1)	842.5	(1.0)	887.1	(1.1)	1024.2	(1.2)	
	5	1014.1	(1.0)	1005.7	(1.0)	3215.4	(3.2)	908.2	(0.9)	940.7	(0.9)	1363.9	(1.4)	14118.1	(14.2)	991.5	(1.0)	1017.7	(1.0)	2024.5	(2.0)	
	20	746.2	(0.9)	756.3	(0.9)	2035.2	(2.4)	622.9	(0.7)	605.7	(0.7)	1420.8	(1.7)	14066.0	(16.4)	855.8	(1.0)	838.6	(1.0)	1691.0	(2.0)	
F	100	626.4	(0.7)	654.6	(0.8)	49210.4	(58.1)	662.2	(0.8)	606.7	(0.7)	1580.1	(1.9)	14091.2	(16.6)	846.5	(1.0)	868.7	(1.0)	1415.8	(1.7)	
	5	379.0	(1.1)	380.4	(1.1)	757.3	(2.2)	334.1	(1.0)	330.8	(1.0)	416.8	(1.2)	2998.3	(8.9)	337.4	(1.0)	349.6	(1.0)	414.2	(1.2)	
	20	332.6	(1.1)	337.7	(1.1)	528.0	(1.7)	305.6	(1.0)	316.7	(1.0)	407.8	(1.3)	2995.1	(9.6)	312.5	(1.0)	300.5	(1.0)	345.4	(1.1)	
	100	319.7	(1.0)	318.4	(1.0)	2999.9	(9.7)	307.8	(1.0)	309.1	(1.0)	435.7	(1.4)	2995.4	(9.7)	310.0	(1.0)	307.1	(1.0)	337.0	(1.1)	
J	5	860.7	(1.1)	853.7	(1.1)	1611.9	(2.1)	770.0	(1.0)	760.6	(1.0)	906.1	(1.2)	5780.2	(7.5)	771.5	(1.0)	741.5	(1.0)	1226.6	(1.6)	
	20	775.1	(1.0)	770.2	(1.0)	1149.1	(1.5)	687.8	(0.9)	707.2	(0.9)	867.3	(1.1)	5781.6	(7.5)	770.5	(1.0)	769.9	(1.0)	1000.3	(1.3)	
	100	768.6	(1.0)	767.2	(1.0)	28683.7	(35.5)	711.4	(0.9)	680.8	(0.8)	950.6	(1.2)	5782.2	(7.2)	807.0	(1.0)	778.3	(1.0)	950.6	(1.2)	
	5	871.6	(1.1)	859.8	(1.1)	1468.2	(1.8)	739.8	(0.9)	756.9	(0.9)	868.2	(1.1)	5968.9	(7.4)	804.3	(1.0)	764.2	(1.0)	1101.4	(1.4)	

TABLE 7. Mean user run times (under unknown margins) in milliseconds. The numbers in parentheses denote the factors of the corresponding entries with respect to the performance of the MLE.

τ	Family	d	Estimator										
			$\tau_{\hat{\tau}}$	$\tau_{\hat{\theta}}$	β	MDE_{τ}^{CVM}	MDE_{τ}^{KS}	MDE_{τ}^{CVM}	MDE_{τ}^{KS}	MLE	SMLE	DMLE	
0.25	A	5	6.5 (0.3)	10.1 (0.5)	3.1 (0.1)	63.0 (3.0)	87.7 (4.1)	68.3 (3.2)	90.9 (4.3)	21.3 (1.0)	3966.3 (186.0)	3.5 (0.2)	
		20	59.0 (2.4)	139.6 (5.6)	4.1 (0.2)	253.8 (10.2)	404.9 (16.3)	275.9 (11.1)	446.4 (18.0)	24.8 (1.0)	3685.1 (148.6)	3.9 (0.2)	
		100	1339.5 (23.0)	3502.7 (60.2)	7.1 (0.1)	2666.4 (45.8)	3816.6 (65.6)	3336.2 (57.4)	4271.6 (73.4)	58.2 (1.0)	3964.5 (68.2)	4.4 (0.1)	
	C	5	6.1 (0.3)	6.1 (0.3)	2.5 (0.1)	59.7 (3.0)	84.6 (4.2)	60.1 (3.0)	90.6 (4.5)	20.1 (1.0)	10270.0 (510.2)	2.9 (0.1)	
		20	58.3 (2.2)	58.3 (2.2)	3.3 (0.1)	251.5 (9.6)	333.9 (12.7)	273.8 (10.4)	410.7 (15.7)	26.2 (1.0)	9752.5 (371.7)	2.7 (0.1)	
		100	1336.1 (18.2)	1337.0 (18.2)	6.7 (0.1)	2343.2 (31.9)	3581.7 (48.8)	2797.5 (38.1)	4590.4 (62.5)	73.4 (1.0)	9413.1 (128.2)	3.3 (0.0)	
	F	5	6.9 (0.2)	13.6 (0.5)	4.4 (0.2)	79.6 (2.9)	115.8 (4.2)	91.8 (3.3)	126.6 (4.5)	27.9 (1.0)	10895.9 (391.0)	13.3 (0.5)	
		20	59.0 (1.4)	224.7 (5.2)	5.3 (0.1)	317.8 (7.4)	438.5 (10.2)	343.8 (8.0)	496.9 (11.5)	43.1 (1.0)	10908.1 (253.0)	11.7 (0.3)	
		100	1345.8 (11.4)	5565.1 (47.1)	9.0 (0.1)	3325.2 (28.1)	4142.6 (35.0)	3412.2 (28.9)	4576.6 (38.7)	118.2 (1.0)	11207.6 (94.8)	12.3 (0.1)	
	G	5	5.7 (0.1)	6.1 (0.1)	2.4 (0.0)	41.1 (0.5)	80.8 (1.0)	49.3 (0.6)	89.9 (1.1)	78.6 (1.0)	9999.5 (127.2)	1.5 (0.0)	
		20	58.0 (0.4)	58.4 (0.4)	2.7 (0.0)	205.1 (1.5)	355.8 (2.6)	240.0 (1.8)	387.3 (2.9)	135.3 (1.0)	9642.7 (71.3)	1.6 (0.0)	
		100	1367.5 (3.6)	1377.2 (3.6)	4.3 (0.0)	2222.0 (5.8)	3302.3 (8.7)	2665.2 (7.0)	4455.5 (11.7)	380.7 (1.0)	9115.7 (23.9)	2.3 (0.0)	
J	5	6.7 (0.2)	16.4 (0.4)	4.3 (0.1)	52.8 (1.2)	81.6 (1.9)	54.3 (1.3)	94.2 (2.2)	42.5 (1.0)	10620.3 (249.9)	5.5 (0.1)		
	20	59.2 (1.1)	312.0 (5.6)	4.5 (0.1)	227.5 (4.1)	315.3 (5.7)	238.1 (4.3)	401.7 (7.2)	55.7 (1.0)	10349.2 (185.8)	7.7 (0.1)		
	100	1346.0 (8.8)	7328.9 (47.7)	9.5 (0.1)	2248.2 (14.6)	3549.6 (23.1)	2723.1 (17.7)	4469.4 (29.1)	153.6 (1.0)	9479.6 (61.7)	9.1 (0.1)		
0.75	C	5	5.7 (0.3)	6.0 (0.3)	2.4 (0.1)	70.2 (3.8)	95.8 (5.1)	70.8 (3.8)	103.3 (5.5)	18.6 (1.0)	9531.7 (511.5)	2.3 (0.1)	
		20	55.8 (2.2)	56.7 (2.3)	3.1 (0.1)	258.1 (10.3)	352.1 (14.0)	308.5 (12.3)	425.8 (17.0)	25.1 (1.0)	10392.1 (414.6)	2.6 (0.1)	
		100	1278.9 (21.2)	1282.7 (21.3)	6.6 (0.1)	2757.5 (45.8)	4207.0 (69.9)	3338.1 (55.5)	4328.8 (71.9)	60.2 (1.0)	9734.6 (161.7)	3.0 (0.0)	
F	5	6.9 (0.3)	16.7 (0.6)	4.8 (0.2)	95.0 (3.6)	107.9 (4.0)	98.0 (3.7)	118.0 (4.4)	26.7 (1.0)	11284.8 (422.0)	11.5 (0.4)		
	20	57.0 (1.5)	244.1 (6.4)	5.1 (0.1)	333.4 (8.7)	459.7 (12.0)	368.6 (9.6)	517.6 (13.5)	38.3 (1.0)	11185.6 (292.3)	11.3 (0.3)		
	100	1307.8 (13.1)	5910.5 (59.4)	9.7 (0.1)	3265.4 (32.8)	3848.2 (38.7)	4031.5 (40.5)	4563.2 (45.9)	99.5 (1.0)	10563.6 (106.2)	12.3 (0.1)		
G	5	6.0 (0.1)	5.6 (0.1)	2.9 (0.1)	56.1 (1.0)	75.5 (1.4)	53.8 (1.0)	88.8 (1.6)	55.3 (1.0)	8612.3 (155.6)	1.5 (0.0)		
	20	56.0 (0.6)	56.7 (0.6)	2.9 (0.0)	239.9 (2.7)	364.1 (4.1)	270.5 (3.0)	403.7 (4.5)	89.1 (1.0)	9264.3 (103.9)	1.8 (0.0)		
	100	1291.9 (4.8)	1279.1 (4.7)	4.2 (0.0)	2332.0 (8.6)	3676.8 (13.5)	2574.3 (9.5)	4633.4 (17.1)	271.4 (1.0)	9082.9 (33.5)	2.6 (0.0)		
J	5	7.3 (0.2)	16.6 (0.5)	4.8 (0.1)	62.6 (1.9)	92.5 (2.9)	70.1 (2.2)	103.4 (3.2)	32.4 (1.0)	10673.4 (329.2)	5.7 (0.2)		
	20	57.0 (1.1)	308.1 (6.1)	5.3 (0.1)	260.4 (5.1)	382.4 (7.5)	274.4 (5.4)	428.4 (8.4)	50.9 (1.0)	9682.6 (190.2)	7.3 (0.1)		
	100	1276.6 (11.0)	8129.0 (70.0)	9.5 (0.1)	2545.1 (21.9)	3540.6 (30.5)	2918.5 (25.1)	3915.4 (33.7)	116.1 (1.0)	9530.5 (82.1)	9.6 (0.1)		

References

- [Barbe et al., 1996] Barbe, P., Genest, C., Ghouli, K., and Rémillard, B. (1996). On Kendall's process. *Journal of Multivariate Analysis*, 58:197–229.
- [Berg, 2009] Berg, D. (2009). Copula goodness-of-fit testing: an overview and power comparison. *The European Journal of Finance*.
- [Berg and Aas, 2009] Berg, D. and Aas, K. (2009). Models for construction of multivariate dependence – a comparison study. *The European Journal of Finance*, 15(7):639–659.
- [Brahimi and Necir, 2011] Brahimi, B. and Necir, A. (2011). A semiparametric estimation of copula models based on the method of moments.
- [Charpentier et al., 2007] Charpentier, A., Fermanian, J.-D., and Scaillet, O. (2007). The estimation of copulas: Theory and practice. In Rank, J., editor, *Copulas: From Theory to Applications in Finance*, pages 35–62. Risk Books.
- [Chavez-Demoulin et al., 2013] Chavez-Demoulin, V., Embrechts, P., and Hofert, M. (2013). An extreme value approach for modeling operational risk losses depending on covariates.
- [D'Agostino and Stephens, 1986] D'Agostino, R. B. and Stephens, M. A. (1986). *Goodness-of-fit techniques*. Dekker.
- [Dimitrova et al., 2008] Dimitrova, D. S., Kaishev, V. K., and Penev, S. I. (2008). GeD spline estimation of multivariate Archimedean copulas. *Computational Statistics & Data Analysis*, 52:3570–3582.
- [Donnelly and Embrechts, 2010] Donnelly, C. and Embrechts, P. (2010). The devil is in the tails: actuarial mathematics and the subprime mortgage crisis. *ASTIN Bulletin*, 40(1):1–33.
- [Duchateau and Janssen, 2008] Duchateau, L. and Janssen, P. (2008). *The Frailty Model*. Springer.
- [Embrechts and Hofert, 2011] Embrechts, P. and Hofert, M. (2011). Comments on: Inference in multivariate Archimedean copula models. *TEST*, 20(2):263–270.
- [Feller, 1971] Feller, W. (1971). *An Introduction to Probability Theory and Its Applications*, volume 2. Wiley, 2 edition.
- [Flajolet and Sedgewick, 1995] Flajolet, P. and Sedgewick, R. (1995). Mellin transforms and asymptotics: Finite differences and Rice's integrals. *Theoretical Computer Science*, 144:101–124.
- [Genest et al., 1995] Genest, C., Ghouli, K., and Rivest, L.-P. (1995). A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82(3):543–552.
- [Genest et al., 2011] Genest, C., Nešlehová, J., and Ziegel, J. (2011). Inference in multivariate Archimedean copula models. *TEST*. in press.
- [Genest et al., 2009] Genest, C., Rémillard, B., and Beaudoin, D. (2009). Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics*, 44:199–213.
- [Genest and Rivest, 1993] Genest, C. and Rivest, L.-P. (1993). Statistical inference procedures for bivariate Archimedean copulas. *Journal of the American Statistical Association*, 88(423):1034–1043.
- [Griewank and Walther, 2003] Griewank, A. and Walther, A. (2003). Introduction to automatic differentiation. *Proceedings of GAMM 2002*, 2(1):45–49.
- [Hering, 2011] Hering, C. (2011). *Estimation Techniques and Goodness-of-fit Tests for Certain Copula Classes in Large Dimensions*. PhD thesis, Ulm University.
- [Hering and Hofert, 2013] Hering, C. and Hofert, M. (2013). Goodness-of-fit tests for Archimedean copulas in large dimensions.
- [Hofert, 2010] Hofert, M. (2010). *Sampling Nested Archimedean Copulas with Applications to CDO Pricing*. Südwestdeutscher Verlag für Hochschulschriften AG & Co. KG. PhD thesis.
- [Hofert, 2012] Hofert, M. (2012). A stochastic representation and sampling algorithm for nested Archimedean copulas. *Journal of Statistical Computation and Simulation*, 82(9):1239–1255.
- [Hofert and Mächler, 2011] Hofert, M. and Mächler, M. (2011). Nested Archimedean copulas meet R: The nacopula package. *Journal of Statistical Software*, 39(9):1–20.
- [Hofert et al., 2012] Hofert, M., Mächler, M., and McNeil, A. J. (2012). Likelihood inference for Archimedean copulas in high dimensions under known margins. *Journal of Multivariate Analysis*, 110:133–150.
- [Hofert and Pham, 2013] Hofert, M. and Pham, D. (2013). Densities of nested Archimedean copulas. *Journal of Multivariate Analysis*.

- [Hofert and Scherer, 2011] Hofert, M. and Scherer, M. (2011). CDO pricing with nested Archimedean copulas. *Quantitative Finance*, 11(5):775–787.
- [Hofert and Vrins, 2013] Hofert, M. and Vrins, F. (2013). Sibuya copulas. *Journal of Multivariate Analysis*, 114:318–337.
- [Hougaard, 2000] Hougaard, P. (2000). *Analysis of Multivariate Survival Data*. Springer.
- [Joe, 1997] Joe, H. (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall/CRC.
- [Kojadinovic and Yan, 2010] Kojadinovic, I. and Yan, J. (2010). Modeling multivariate distributions with continuous margins using the copula r package. *Journal of Statistical Software*, 34(9):1–20.
- [Lambert, 2007] Lambert, P. (2007). Archimedean copula estimation using Bayesian splines smoothing techniques. *Computational Statistics & Data Analysis*, 51:6307–6320.
- [Mächler, 2011] Mächler, M. (2011). Numerically stable Frank copula functions via multiprecision: R package ‘rmpfr’.
- [Mächler, 2012] Mächler, M. (2012). Accurately computing $\log(1 - \exp(-|a|))$.
- [McNeil et al., 2005] McNeil, A. J., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques, Tools*. Princeton University Press.
- [McNeil and Nešlehová, 2009] McNeil, A. J. and Nešlehová, J. (2009). Multivariate Archimedean copulas, d -monotone functions and l_1 -norm symmetric distributions. *The Annals of Statistics*, 37(5b):3059–3097.
- [Nelsen, 2006] Nelsen, R. B. (2006). *An Introduction to Copulas*. Springer.
- [Qu et al., 2010] Qu, X., Zhou, J., and Shen, X. (2010). Archimedean copula estimation and model selection via l_1 -norm symmetric distribution. *Insurance: Mathematics and Economics*, 46:406–414.
- [Savu and Trede, 2010] Savu, C. and Trede, M. (2010). Hierarchies of Archimedean copulas. *Quantitative Finance*, 10(3):295–304.
- [Scarsini, 1984] Scarsini, M. (1984). On measures of concordance. *Stochastica*, 8(3):201–218.
- [Schmid and Schmidt, 2007] Schmid, F. and Schmidt, R. (2007). Nonparametric inference on multivariate versions of Blomqvist’s beta and related measures of tail dependence. *Metrika*, 66:323–354.
- [Schönbucher and Schubert, 2001] Schönbucher, P. J. and Schubert, D. (2001). Copula-dependent default risk in intensity models.
- [Sibuya, 1959] Sibuya, M. (1959). Bivariate extreme statistics, i. *Annals of the Institute of Statistical Mathematics*, 11(2):195–210.
- [Stephenson, 2009] Stephenson, A. G. (2009). High-dimensional parametric modelling of multivariate extreme events. *Australian & New Zealand Journal of Statistics*, 51(1):77–88.
- [Tsukahara, 2005] Tsukahara, H. (2005). Semiparametric estimation in copula models. *The Canadian Journal of Statistics*, 33(3):357–375.
- [Weiß, 2010] Weiß, G. N. F. (2010). Copula parameter estimation: numerical considerations and implications for risk management. *The Journal of Risk*, 13(1):17–53.