Structure Determination and Estimation of Hierarchical Archimedean Copulas Based on Kendall Correlation Matrix

Jan Górecki[1], Martin Holeňa[2]

[1] Department of Informatics
SBA in Karvina, Silesian University in Opava
Karvina, Czech Republic
gorecki@opf.slu.cz
[2] Institute of Computer Science
Academy of Sciences of the Czech Republic
Praha, Czech Republic
martin@cs.cas.cz

Abstract. An estimation method for the copula of a continuous multivariate distribution is proposed. A popular class of copulas, namely the class of hierarchical Archimedean copulas, is considered. The proposed method is based on the close relationship of the copula structure and the values of Kendall's tau computed on all its bivariate margins. A generalized measure based on Kendall's tau adapted for purposes of the estimation is introduced. A simple algorithm implementing the method is provided and its effectiveness is shown in several experiments including its comparison to other available methods. The results show that the proposed method can be regarded as a suitable alternative to existing methods in the terms of goodness of fit and computational efficiency.

Keywords: copula, hierarchical Archimedean copula, copula estimation, structure determination, Kendall's correlation coefficient

1 Introduction

Studying relationships among random quantities is a crucial task in the field of knowledge discovery and data mining (KDDM). Having a dataset collected, the relationships among the observed variables can be studied by means of an appropriate measure of stochastic dependence. Under assumption of the multivariate continuous distribution of the variables, the famous Sklar's theorem [29] can be used to decompose the distribution in two components. While the first component describes the distributions of the univariate margins, the second component describes the copula of the distribution containing the whole information about the relationship among the variables. Thus, studying dependencies among the random variables can be restricted without any loss of generality to studying the copula.

Despite the fact that copulas have most success in finance, they are increasingly adopted also in KDDM, where they are used due to their effective mathematical ability to capture even very complex dependence structures among variables. We can see applications of copulas in water-resources and hydro-climatic analysis [4,13,14,17,19], gene analysis [18,31], cluster analysis [3,15,26] or in evolution algorithms, particularly in the estimation of distribution algorithms [7,30]. For an illustrative example, we refer to [13], where the task for anomaly detection in climate that incorporates complex spatio-temporal dependencies is solved using copulas.

Hierarchical Archimedean copulas (HACs) are a frequently used alternative to the most popular Gaussian copulas due to their flexibility and conveniently limited number of parameters. Despite their popularity, feasible techniques for HAC estimation are addressed only in few papers. Most of them assume in the estimation process a given structure of a copula, which is motivated trough applications in economy, see [27,28]. There exists only one recently published paper [23], which addresses the estimation technique generally, i.e., the estimation also concerns the proper structure determination of the HAC.

The mentioned paper describes a multi-stage procedure, which is used both for the structure determination and the estimation of the parameters. The authors devote mainly to the estimation of the parameters using the maximum-likelihood (ML) technique and briefly mention its alternative, which uses for the parameters estimation the relationship between the copula parameter and the value of Kendall's tau computed on a bivariate margin of the copula (shortly, $\theta-\tau$ relationship). The authors present six approaches denoted as $\tau_{\Delta\tau>0}$, τ_{binary}, Chen, θ_{binary}, $\theta_{binary\ aggr.}$ and θ_{RML} to the structure determination. The first two approaches are based on the $\theta-\tau$ relationship, the third approach is based on the Chen test statistics [2] and the last three approaches are based on the ML technique. The first five approaches lead to biased estimators, what can be seen in the results of the attached simulation study, and the the sixth (θ_{RML}) is used for re-estimation and thus for better approximation of the parameters of the true copula. θ_{RML} shows the best goodness-of-fit (measured by Kullback-Leibler divergence) of the resulting estimates. However, the best approximation of the true parameters with θ_{RML} is possible only in the cases, when the structure is properly determined (the estimated structure equals the true structure). But, as θ_{RML} is based on the biased $\theta_{binary\ aggr.}$, which often does not return the true structure due to the involved bias, θ_{RML} also cannot return close approximation of the true parameters in the cases, when the structure is determined improperly. Moreover, the number of those cases rapidly increases with the increasing data dimension, as we show later in Section 4.

In our paper, we propose the construction of the estimator for HACs that approximates the parameters of the true copula better than the previously mentioned methods, and thus also increases the ratio of properly determined structures. Avoiding the need of re-estimation, we also gain high computational efficiency. The included experiments on simulated data show that our approach outperforms all the other above mentioned methods in the sense of goodness-of-fit,

the properly determined structures ratio and also in the time consumption, which is even slightly lower than the most efficient binary methods $\tau_{binary}, \theta_{binary}$.

The paper is structured as follows. The next section summarizes some necessary theoretical concepts concerning Archimedean copulas (ACs) and HACs. Section 3 presents the new approach to the HAC estimation. Section 4 describes the experiments and their results and Section 5 concludes this paper.

2 Preliminaries

2.1 Copulas

Definition 1. *For every $d \geq 2$, a* d-dimensional copula *(shortly,* d-copula*) is a d-variate distribution function on $\mathbb{I}^d$ ($\mathbb{I}$ is the unit interval), whose univariate margins are uniformly distributed on $\mathbb{I}$.*

Copulas establish a connection between general joint distribution functions (d.f.s) and its univariate margins (in text below we use only *margin* for term *univariate margin*), as can be seen in the following theorem.

Theorem 1. ***(Sklar's Theorem)*** *[29] Let H be a d-variate d.f. with univariate margins $F_1, ..., F_d$. Let A_j denote the range of F_j, $A_j := F_j(\overline{\mathbb{R}})(j = 1, ..., d), \overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$. Then there exists a copula C such for all $(x_1, ..., x_d) \in \overline{\mathbb{R}}^d$,*

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

Such a C is uniquely determined on $A_1 \times ... \times A_d$ and, hence, it is unique if $F_1, ..., F_d$ are all continuous. Conversely, if $F_1, ..., F_d$ are univariate d.f.s, and if C is any d-copula, then the function $H : \overline{\mathbb{R}}^d \to \mathbb{I}$ defined by (1) is a d-dimensional distribution function with margins $F_1, ..., F_d$.

Through the Sklar's theorem, one can derive for any d-variate d.f. its copula C using (1). In case that the margins $F_1, ..., F_d$ are all continuous, the copula C is given by $C(u_1, ..., u_d) = H(F_1^-(u_1), ..., F_d^-(u_d))$, where $F_i^-, i \in \{1, ..., d\}$ denotes pseudo-inverse of F_i given by $F_i^-(s) = \inf\{t|\ F_i(t) \geq s\}, s \in \mathbb{I}$. Many classes of copulas are derivable in this way from popular joint d.f.s, e.g., the most popular class of Gaussian copulas is derived using H corresponding to a d-variate Gaussian distribution. But, using this process often results in copulas not expressible in closed form, what can bring difficulties in some applications.

2.2 Archimedean Copulas

This drawback is overcame while using (exchangeable) Archimedean copulas, due to their different construction process. ACs are not constructed using the Sklar's theorem, but instead of it, one starts with a given functional form and asks for properties needed to obtain a proper copula. As a result of such a construction, ACs are always expressed in closed form, which is one of the main advantages of this class of copulas [10]. To construct ACs, we need the notion of an *Archimedean generator* and of *complete monotonicity*.

Definition 2. Archimedean generator *(shortly,* generator*) is a continuous, non-increasing function* $\psi : [0, \infty] \to [0, 1]$*, which satisfies* $\psi(0) = 1, \psi(\infty) = \lim_{t\to\infty} \psi(t) = 0$ *and is strictly decreasing on* $[0, \inf\{t : \psi(t) = 0\}]$*. We denote the set of all generators as* Ψ*.*

Definition 3. *A function f is called* completely monotone *(shortly, c.m.) on* $[a, b]$*, if* $(-1)^k f^{(k)}(x) \geq 0$ *holds for every* $k \in \mathbb{N}_0, x \in (a, b)$*. We denote the set of all completely monotonous generators as* Ψ_∞*.*

Definition 4. *Any d-copula C is called* Archimedean copula *(we denote it d-AC), if it admits the form*

$$C(\mathbf{u}) := C(\mathbf{u}; \psi) := \psi(\psi^{-1}(u_1) + ... + \psi^{-1}(u_d)), \boldsymbol{u} \in \mathbb{I}^d, \tag{2}$$

where $\psi \in \Psi$ *and the* $\psi^{-1} : [0, 1] \to [0, \infty]$ *is defined* $\psi^{-1}(s) = \inf\{t : \psi(t) = s\}, s \in \mathbb{I}$*.*

For verifying whether function C given by (2) is a proper copula, we can use the property stated in Definiton 3. A condition sufficient for C to be a copula is stated as follows.

Theorem 2. *[21] If* $\psi \in \Psi$ *is completely monotone, then the function C given by (2) is a copula.*

We can see from Definition 4 and from the properties of generators that having a random vector $\mathbf{U}$ distributed according to some AC, all its k-dimensional ($k < d$) marginal copulas have the same marginal distribution. It implies that all multivariate margins of the same dimension are equal, thus, e.g., the dependence among all pairs of components is identical. This symmetry of ACs is often considered to be a rather strong restriction, especially in high dimensional applications.

Given the number of variables, to derive the explicit form of an AC to work with, we need the explicit form of generators. The reader can find many explicit forms of the generators in, e.g., [22]. In this paper, we use and present only the Clayton generator, defined $(1 + t)^{-1/\theta}$, which corresponds to the family of the Clayton copulas. Copulas based on this generator have been used, e.g., to study correlated risks, because they exhibit strong left tail dependence and relatively weak right tail dependence. The explicit parametric form of a bivariate Clayton copula is $C(u_1, u_2; \psi) = \left({u_1}^{-\theta} + {u_2}^{-\theta} - 1\right)^{-\frac{1}{\theta}}$ [22].

2.3 Hierarchical Archimedean Copulas

To allow for asymmetries, one may consider the class of HACs (often also called *nested Archimedean copulas*), recursively defined as follows.

Definition 5. *[11] A d-dimensional copula C is called* hierarchical Archimedean copula *if it is an AC with arguments possibly replaced by other hierarchical Archimedean copulas. If C is given recursively by (2) for* $d = 2$ *and*

$$C(\mathbf{u}; \psi_1, ..., \psi_{d-1}) = \psi_1(\psi_1^{-1}(u_1) + \psi_1^{-1}(C(u_2, ..., u_d; \psi_2, ..., \psi_{d-1}))), \boldsymbol{u} \in \mathbb{I}^d, \tag{3}$$

for $d \geq 2$, C is called fully-nested hierarchical Archimedean copula (FHAC)[3] *with $d-1$* nesting levels*. Otherwise C is called* partially-nested hierarchical Archimedean copula (PHAC)[4]*.*

Remark 1. We denote a d-dimensional HAC as d-HAC, and analogously d-FHAC and d-PHAC.

From the definition, we can see that ACs are special cases of HACs. The most simple proper 3-PHAC is with two nesting levels. The copula is given by

$$\begin{aligned} C(\mathbf{u};\psi_1,\psi_2) &= C(u_1, C(u_2,u_3;\psi_2);\psi_1) \\ &= \psi_1(\psi_1^{-1}(u_1) + \psi_1^{-1}(\psi_2(\psi_2^{-1}(u_2) + \psi_2^{-1}(u_3)))), \mathbf{u} \in \mathbb{I}^3. \end{aligned} \quad (4)$$

As in the case of ACs, we can ask for necessary and sufficient condition for the function C given by (3) to be a proper copula. Partial answer to this question in form of sufficient condition is contained in the following theorem.

Theorem 3. ***(McNeil (2008))*** *[20] If $\psi_j \in \Psi_\infty, j \in \{1,...,d-1\}$ such that $\psi_k^{-1} \circ \psi_{k+1}$ have completely monotone derivatives for all $k \in \{1,...,d-2\}$, then $C(\boldsymbol{u};\psi_1, ...,\psi_{d-1})$, $\boldsymbol{u} \in \mathbb{I}^d$, given by (3) is a copula.*

McNeil's theorem is stated only for fully-nested HACs, but it can be easily translated also for use with partially-nested HACs (for more see [20]). The condition for $(\psi_1^{-1} \circ \psi_2)'$ to be compete monotone is often called the *nesting condition*.

A d-HAC structure, which is given by the recursive nesting in the definition, can be expressed as a tree with $k \leq d-1$ non-leaf nodes (shortly, nodes), which correspond to the generators $\psi_1, ..., \psi_k$, and d leafs, which correspond to the variables $u_1, ..., u_d$. If the structure corresponds to a binary tree, then $k = d-1$. In other case $k < d-1$. Thus, a HAC structure is viewed as a tree in the next text. Also, for the sake of simplicity, we assume only binary HAC structures.

Let s be the structure of a d-HAC. Each 2-AC is determined just by its corresponding generator, and, if we identify each node in s with one generator, we have always nodes $\psi_1, ..., \psi_{d-1}$. For a node ψ denote as $\mathcal{D}_n(\psi)$ the set of all descendant nodes of ψ, $\mathcal{P}(\psi)$ the parent node of ψ, $\mathcal{H}_l(\psi)$ the left child of ψ and $\mathcal{H}_r(\psi)$ the right child of ψ.

For simplicity, a d-HAC structure s is denoted as a sequence of reordered indices $\{1,...,d\}$ using parentheses to mark the variables with the same parent node. For example, the structure of the copula given by (4) is denoted as (1(23)). The inner parenthesis corresponds to the fact that for the variables u_2, u_3 is $\mathcal{P}(u_2) = \mathcal{P}(u_3) = \psi_2$. As u_2, u_3 are connected through their parent, we can introduce a new variable denoted as z_{23}, which represents the variables u_2, u_3 and is defined as $z_{23} = C(u_2, u_3; \psi_2)$. Then (4) turns in $\psi_1(\psi_1^{-1}(u_1) + \psi_1^{-1}(z_{23})) = C(u_1, z_{23}; \psi_1)$, and thus the outer parenthesis in the notation of the structure corresponds to the fact that for the variables u_1, z_{23} is $\mathcal{P}(u_1) = \mathcal{P}(z_{23}) = \psi_1$.

[3] sometimes called *fully-nested Archimedean copula*

[4] sometimes called *partially-nested Archimedean copula*

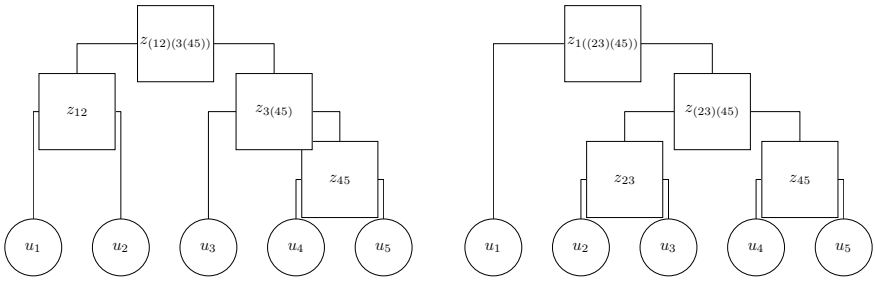

Fig. 1. On the left side is depicted the 5-PHAC structure denoted as ((12)(3(45))) and on the right side is depicted the 5-PHAC structure denoted as (1((23)(45))).

The structure of the 4-FHAC given as in Definition 5 is denoted as (1(2(34))), for 5-FHAC, it is (1(2(3(45)))), etc. Analogously, for PHACs, ((12)(3(45))) and (1((23)(45))) denote the structures depicted on the left and the right side in Figure 1.

When using HACs in applications, there exist, for example for $d = 10$, more than 280 millions of possible HAC structures (including also non-binary ones) and each 10-HAC can incorporate up to 9 parameters (using only one-parametric generators) in generators from possibly different families. If choosing the model that the best fit the data, this is much more complex situation relative to the case when using ACs, which have just one structure, one parameter and one Archimedean family.

To derive the explicit parametric form a d-HAC C, we need the explicit parametric forms of its generators $\psi_1, ..., \psi_{d-1}$, which involve the parameters $\theta_1, ..., \theta_{d-1}$ (θ_i corresponds to the generator $\psi_i, i = 1, ..., d-1$), and its structure s. Due to this, the copula C is also denoted as $C(\psi, \theta; s)(u_1, ..., u_d)$ in the rest of the text. For example, the 3-HAC that is given by (4) and assuming both of its generators ψ_1, ψ_2 to be Clayton generators, can be denoted as $C(\psi_1, \psi_2, \theta_1, \theta_2; (1(23)))$ and its parametric form is given as

$$C(\psi_1, \psi_2, \theta_1, \theta_2; (1(23))) = \left(\left({u_2}^{-\theta_2} + {u_3}^{-\theta_2} - 1 \right)^{-\frac{1}{\theta_2}} \right)^{-\theta_1} + {u_1}^{-\theta_1} - 1 \Bigg)^{-\frac{1}{\theta_1}}. \tag{5}$$

2.4 Kendall's tau and its generalization

The standard definition of Kendall's tau for two random variables X, Y is given as follows. Let (X_1, Y_1) and (X_2, Y_2) be independent random vectors with the same distribution as (X, Y). Then the population version of Kendall's tau is defined as the probability of concordance minus the probability of discordance, i.e.,

$$\tau = \tau_{XY} = P((X_1 - X_2)(Y_1 - Y_2) > 0) - P((X_1 - X_2)(Y_1 - Y_2) < 0). \tag{6}$$

As we are interested in Kendall's tau relationship to a general bivariate copula, we use its definition given by (as in [4])

$$\tau(C) = 4 \int_{\mathbb{I}^2} C(u_1, u_2) dC(u_1, u_2) - 1. \tag{7}$$

If C is a 2-AC based on a generator ψ, and ψ depends on the parameter $\theta \in \mathbb{R}$, then (7) states an explicit relationship between θ and τ, which can be often expressed in a closed form. For example, if C is a Clayton copula, we get $\tau = \theta/(\theta+2)$ (the relationship between θ and τ for other generators can be found, e.g., in[10]). The inversion of this relationship establish an estimator of the parameter θ, which can be based on the empirical version of τ given by (as in [4])

$$\tau_n = \frac{4}{n(n-1)} \sum_{i=1,j=1}^{n} \mathbf{1}_{\{(u_{i1}-u_{j1})(u_{i2}-u_{j2})>0\}}, \tag{8}$$

where $(u_{\bullet 1}, u_{\bullet 2})$ denotes the realizations of r.v.s $(U_1, U_2) \sim C$.

This estimation method was introduced in [5] as a method-of-moments estimator for bivariate one-parameter Archimedean copulas. The copula parameter $\theta \in \Theta \subseteq \mathbb{R}$ is estimated by $\hat{\theta}_n$ such that $\tau(\hat{\theta}_n) = \tau_n$, where $\tau(\theta)$ denotes Kendall's tau of the corresponding Archimedean family viewed as a function of the parameter $\theta \in \Theta \subseteq \mathbb{R}$, i.e., that $\hat{\theta}_n = \tau^{-1}(\tau_n)$, assuming the inverse τ^{-1} of τ exists. If the equation has no solution, this estimation method does not lead to an estimator. Unless there is an explicit form for τ^{-1}, $\hat{\theta}_n$ is computed by numerical root finding[12].

This estimation method can also be generalized for ACs when $d > 2$, see [1,12,16,28]. The generalized method is using pairwise sample version of Kendall's tau. If τ_{ij}^n denotes the sample version of Kendall's tau between the i-th and j-th data column, then θ is estimated by

$$\hat{\theta}_n = \tau^{-1}\left(\binom{d}{2}^{-1} \sum_{1 \le i \le j \le d} \tau_{ij}^n \right). \tag{9}$$

As can be seen, the parameter is chosen such that Kendall's tau equals the average over all pairwise sample versions of Kendall's tau. Properties of this estimator are not known and also not easy to derive since the average is taken over dependent data columns [12]. However, $\binom{d}{2}^{-1} \sum_{1 \le i \le j \le d} \tau_{ij}^n$ is an unbiased estimator of $\tau(\theta)$. This is an important property and we transfer it later to the estimator that we use for the structure determination, which we base on appropriately selected pairwise sample versions of Kendall's tau.

To use the generalized method mentioned in the previous paragraph with HACs, we define a generalization of τ for m (possibly > 2) random variables (r.v.s). For simplification denote the set of pairs of r.v.s as $\mathbf{U}_{IJ} = \{(U_i, U_j) | (i,j) \in I \times J\}$, where $I, J \subset \{1, ..., d\}, I \neq \emptyset \neq J, (U_1, ..., U_d) \sim C$, C is a d-HAC.

Definition 6. *Let τ be the Kendall's tau, $g : [0,1]^k \to [0,1], k \in \mathbb{N}$, be an aggregation function (like, e.g., max, min or mean), which has the following*

properties: 1) $g(u,...,u) = u$ *for all* $u \in \mathbb{I}$ *and 2)* $g(u_{p_1},...,u_{p_k}) = g(u_1,...,u_k)$ *for all* $u_1,...,u_k \in \mathbb{I}$ *and all permutations* p *of* $\{1,...,k\}$*. Then define an* aggregated *Kendall's tau* τ^g *as*

$$\tau^g(\mathbf{U}_{IJ}) = \begin{cases} \tau(U_i, U_j) & \text{if } I = \{i\}, J = \{j\} \\ g(\tau(U_{i_1}, U_{j_1}), \tau(U_{i_1}, U_{j_2}), ..., \tau(U_{i_l}, U_{j_q})), & \text{else,} \end{cases} \tag{10}$$

where $I = \{i_1,...,i_l\}, J = \{j_1,...,j_q\}$ *are non-empty disjoint subsets of* $\{1,...,d\}$*.*

As the aggregated τ^g depends only on the pairwise τ and the aggregation function g, we can easily derive its empirical version τ_n^g just by substituting τ in τ^g by its empirical version τ_n given by (8). Then, analogously to the case of ACs, the parameter is estimated as $\hat{\theta}_n = \tau^{-1}(\tau_n^g)$. But, as all bivariate margins of a HAC are not assumed to be identical, each estimate is computed just on some appropriately selected ones. This is later explained by Remark 2.

2.5 Okhrin's algorithm for the structure determination of HAC

We recall the algorithm presented in [24] for the structure determination of HAC, which returns for some unknown HAC C its structure using only the known forms of its bivariate margins. The algorithm uses the following definition.

Definition 7. *Let* C *be a* d*-HAC with generators* $\psi_1,...,\psi_{d-1}$ *and* $(U_1,...,U_d) \sim C$*. Then denote as* $\mathcal{U}_C(\psi_k), k = 1,...,d-1$*, the set of indexes* $\mathcal{U}_C(\psi_k) = \{i | (\exists U_j)\ (U_i, U_j) \sim C(\cdot;\psi_k) \vee (U_j, U_i) \sim C(\cdot;\psi_k), 1 \le i < j \le d\}, k = 1,...,d-1$*.*

Proposition 1. *[24] Defining* $\mathcal{U}_C(u_i) = \{i\}$ *for the leaf* $i, 1 \le i \le d$*, there is an unique disjunctive decomposition of* $\mathcal{U}_C(\psi_k)$ *given by*

$$\mathcal{U}_C(\psi_k) = \mathcal{U}_C(\mathcal{H}_l(\psi_k)) \cup \mathcal{U}_C(\mathcal{H}_r(\psi_k)). \tag{11}$$

For an unknown d-HAC C, knowing all its bivariate margins, its structure can be easily determined using Algorithm 1. We start from the sets $\mathcal{U}_C(u_1),...,\mathcal{U}_C(u_d)$ joining them together through (11) until we reach the node ψ for which $\mathcal{U}_C(\psi) = \{1,...,d\}$.

We illustrate the Algorithm 1 for a 5-HAC given by $C(C(u_1,u_2;\psi_2), C(u_3, C(u_4,u_5;\psi_4);\psi_3);\psi_1) = C(\psi_1,...,\psi_4;((12)(3(45))))(u_1,...,u_5)$. The structure of this copula is depicted on the left side in Figure 1 and its bivariate margins are:

$(U_1,U_2) \sim C(\cdot;\psi_2)$ $(U_1,U_3) \sim C(\cdot;\psi_1)$ $(U_1,U_4) \sim C(\cdot;\psi_1)$ $(U_1,U_5) \sim C(\cdot;\psi_1)$
$(U_2,U_3) \sim C(\cdot;\psi_1)$ $(U_2,U_4) \sim C(\cdot;\psi_1)$ $(U_2,U_5) \sim C(\cdot;\psi_1)$ $(U_3,U_4) \sim C(\cdot;\psi_3)$
$(U_3,U_5) \sim C(\cdot;\psi_3)$ $(U_4,U_5) \sim C(\cdot;\psi_4)$

Now assume that the structure is unknown and only the bivariate margins are known. We see that $\mathcal{U}_C(\psi_1) = \{1,2,3,4,5\}$, $\mathcal{U}_C(\psi_2) = \{1,2\}$, $\mathcal{U}_C(\psi_3) =$

Algorithm 1 The HAC structure determination [24]

Input:
1) $\mathcal{U}_C(\psi_1), ..., \mathcal{U}_C(\psi_{d-1})$,
2) $\mathcal{I} = \{1, ..., d-1\}$

while $\mathcal{I} \neq \emptyset$ **do**
1. $k = \text{argmin}_{i \in \mathcal{I}}(\#\mathcal{U}_C(\psi_i))$, if there are more minima, then choose as k one of them arbitrarily.
2. Find the nodes ψ_l, ψ_r, for which $\mathcal{U}_C(\psi_k) = \mathcal{U}_C(\psi_l) \cup \mathcal{U}_C(\psi_r)$.
3. $\mathcal{H}_l(\psi_k) := \psi_l, \mathcal{H}_r(\psi_k) := \psi_r$.
4. Set $\mathcal{I} := \mathcal{I} \backslash \{k\}$.

end while

Output:
The structure stored in $\mathcal{H}_l(\psi_k), \mathcal{H}_r(\psi_k), k = 1, ..., d-1$

$\{3,4,5\}, \mathcal{U}_C\ (\psi_4) = \{4,5\}$. For leafs $u_1, ..., u_5$, it is defined $\mathcal{U}_C(u_i) = \{i\}, i = 1, ..., 5$. In step 1., there are two minima: $k = 2$ and $k = 4$. We choose arbitrarily $k = 4$. As $\mathcal{U}_C(\psi_4) = \mathcal{U}_C(u_4) \cup \mathcal{U}_C(u_5)$, we set in step 3. $\mathcal{H}_l(\psi_4) := u_4$ and $\mathcal{H}_r(\psi_4) := u_5$. In step 4., we set $\mathcal{I} = \{1,2,3\}$. In the second loop, $k = 2$. As $\mathcal{U}_C(\psi_2) = \mathcal{U}_C(u_1) \cup \mathcal{U}_C(u_2)$, we set in step 3. $\mathcal{H}_l(\psi_2) := u_1$ and $\mathcal{H}_r(\psi_2) := u_2$. In the third loop, we have $k = 3$. As $\mathcal{U}_C(\psi_3) = \mathcal{U}_C(u_3) \cup \mathcal{U}_C(\psi_4)$, we set in step 3. $\mathcal{H}_l(\psi_3) := u_3$ and $\mathcal{H}_r(\psi_3) := \psi_4$. In the last loop, we have $k = 1$. As $\mathcal{U}_C(\psi_1) = \mathcal{U}_C(\psi_2) \cup \mathcal{U}_C(\psi_3)$, we set in step 3. $\mathcal{H}_l(\psi_1) := \psi_2$ and $\mathcal{H}_r(\psi_1) := \psi_3$. Observing the original copula form and Figure 1, we see that we have determined the correct structure, which is stored in $\mathcal{H}_l(\psi_k), \mathcal{H}_r(\psi_k), k = 1, ..., 4$.

3 Our Approach

3.1 HAC structure determination

Recalling Theorem 3, the sufficient condition for C to be a proper copula is that the nesting condition must hold for each generator and its parent in a HAC structure. As this is the only known condition that assures that C is a proper copula, we deal in our work only with the copulas that fulfill this condition. The nesting condition results in constraints on the parameters θ_1, θ_2 of the involved generators ψ_1, ψ_2 (see [10,11]). As $\theta_i, i = 1, 2$ is closely related to τ through (7), there is also an important relationship between the values of τ and the HAC tree structure following from the nesting condition. This relationship is described for the fully-nested 3-HAC given by the form (4) in Remark 2.3.2 in [10]. There, it is shown that if the nesting condition holds for the parent-child pair (ψ_1, ψ_2), then $0 \leq \tau(\psi_1) \leq \tau(\psi_2)$ (as we deal only with HACs with binary structures, which are fully determined by its generator, we use as the domain of τ the set Ψ instead of the usually used set of all 2-copulas). We generalize this statement, using our notation, as follows.

Algorithm 2 The HAC structure determination based on τ

Input:
1) $\mathcal{I} = \{1, ..., d\}$,
2) $(U_1, ..., U_d) \sim C$,
3) τ^g ... an aggregated Kendall's tau,
4) $z_k = u_k, \mathcal{U}_C(z_k) = \mathcal{U}_C(u_k) = \{k\}, k = 1, ..., d$

The structure determination:
for $k = 1, ..., d-1$ **do**
1. $(i, j) := \underset{i^* < j^*, i^* \in \mathcal{I}, j^* \in \mathcal{I}}{\operatorname{argmax}} \tau^g(\mathbf{U}_{\mathcal{U}_C(z_{i^*})\mathcal{U}_C(z_{j^*})})$
2. $\mathcal{U}_C(z_{d+k}) := \mathcal{U}_C(z_i) \cup \mathcal{U}_C(z_j)$
3. $\mathcal{I} := \mathcal{I} \cup \{d+k\} \backslash \{i, j\}$

end for

Output:
$\mathcal{U}_C(z_{d+k}), k = 1, ..., d-1$

Proposition 2. *Let C be a d-HAC with the structure t and the generators $\psi_1, ..., \psi_{d-1}$, where each parent-child pair satisfy the nesting condition. Then $\tau(\psi_i) \leq \tau(\psi_j)$, where $\psi_j \in \mathcal{D}_n(\psi_i)$, holds for each $\psi_i, i = 1, ..., d-1$.*

Proof. As $\psi_j \in \mathcal{D}_n(\psi_i)$, there exists a unique sequence $\psi_{k_1}, ..., \psi_{k_l}$, where $1 \leq k_m \leq d-1, m = 1, ..., l, l \leq d-1, \psi_{k_1} = \psi_i, \psi_{k_l} = \psi_j$ and $\psi_{k-1} = \mathcal{P}(\psi_k)$ for $k = 2, ..., l$. Applying the above mentioned remark for each pair $(\psi_{k-1}, \psi_k), k = 2, ..., l$, we get $\tau(\psi_{k_1}) \leq ... \leq \tau(\psi_{k_l})$. □

Thus, having a branch from t, all its nodes are uniquely ordered according to their value of τ assuming unequal values of τ for all parent-child pairs. This provides an alternative algorithm for the HAC structure determination. We have to assign the generators with the highest values of τ to the lowest levels of the branches in the structure and ascending to higher levels we assign the generators with lower values of τ.

Remark 2. $\tau(\psi_k) = \tau^g(\mathbf{U}_{\mathcal{U}_C(\mathcal{H}_l(\psi_k))\mathcal{U}_C(\mathcal{H}_r(\psi_k))})$ for a d-HAC C and for each $k = 1, ..., d-1$. This is because the bivariate margins $C_{ij}, (i, j) \in \mathcal{U}_C(\mathcal{H}_l(\psi_k)) \times \mathcal{U}_C(\mathcal{H}_r(\psi_k))$ of C are all equal and $g(u, ..., u) = u$ for all $u \in \mathbb{I}$. Thus $\tau(\psi_k)$ depends only on the population version of Kendall correlation matrix.

Computing $\tau(\psi_k), k = 1, ..., d-1$ using Remark 2 and following Proposition 2 leads to the alternative algorithm for HAC structure determination. The algorithm is summarized in Algorithm 2 and can be used for arbitrary $d > 2$ (see [8] for more details including an example for $d = 4$). It returns the sets $\mathcal{U}_C(z_{d+k+1})$ corresponding to the sets $\mathcal{U}_C(\psi_k), k = 1, ..., d-1$. Passing them to Algorithm 1, we avoid their computation from Definition 7 and we get the requested d-HAC structure without a need of knowing the forms of the bivariate margins. Assuming a family for each ψ_k, $\theta - \tau$ relationship for the given family can be used to

obtain the parameters, i.e., $\theta_k = \tau_\theta^{-1}(\tau(\psi_k)), k = 1, ..., d-1$, where τ_θ^{-1} denotes the $\theta - \tau$ relationship, e.g., for Clayton family $\tau_\theta^{-1}(\tau) = 2\tau/(1-\tau)$. Hence we get together with the structure the whole copula.

3.2 HAC estimation

Using τ_n^g instead of τ^g, we can easily derive the empirical version of the structure determination process represented by Algorithms 1, 2. In this way, we base the structure determination only on the values of the pairwise τ. This is an essential property of our approach. Using the $\theta-\tau$ relationship established through (7) for some selected Archimedean family, whole HAC, including its structure and its parameters, can be estimated just from Kendall correlation matrix computed for the realizations of $(U_1, ..., U_d)$, assuming all the generators to be from a selected Archimedean family.

The proposed empirical approach is summarized in Algorithm 3. The Kendall correlation matrix (τ_{ij}^n) is computed for the realizations of the pairs $(U_i, U_j), 1 \leq i < j \leq d$ using (8). The algorithm returns the parameters $\hat{\theta}_1, ..., \hat{\theta}_{d-1}$ of the estimate $\hat{C}$ and the sets $\mathcal{U}_{\hat{C}}(z_{d+k})$ corresponding to the sets $\mathcal{U}_{\hat{C}}(\psi_k), k = 1, ..., d-1$. Passing the sets to Algorithm 1 we get the requested $\hat{C}$ structure.

If g is set to be the average function, and as $\tau_n^{avg}(\theta_k) = g((\tau_{\tilde{i}\tilde{j}}^n)_{(\tilde{i},\tilde{j})\in\mathcal{U}_{\hat{C}}(z_i)\times\mathcal{U}_{\hat{C}}(z_j)})$, where i, j are the indices found in step 1. of the algorithm, then $\tau_n^{avg}(\theta_k)$ is an unbiased estimator of $\tau(\theta_k)$, and thus the structure determination is based only on unbiased estimates, what is another favourable property of the proposed method.

Due to the nesting condition, the parameter $\hat{\theta}_k$ is trimmed in step 3. in order to obtain the resulting estimate as a proper d-HAC. Note that if we allow the generators to be from different Archimedean families, the task is much more complex, and we do not concern it in the paper due to space limitations and refer the reader to [9,10].

Note that the proposed algorithm is just a variation of another famous algorithm, namely the algorithm for agglomerative hierarchical clustering (AHC). Defining $\delta_{ij} = 1 - \tau_{ij}^n$ we establish δ_{ij} to be a standardly used distance between the random variables U_i, U_j. Setting g to be the aggregation function min, avg or max, the algorithm results in (due to $\delta_{ij} = 1-\tau_{ij}^n$) complete-linkage, average-linkage or single-linkage AHC, respectively. As most of statistical softwares include an implementation of AHC, the implementation of the proposed algorithm is straightforward. Moreover, adding the dendrogram obtained during AHC makes the result even more understandable to the user.

4 Experiments

We performed a large number of different experiments on simulated data involving different data dimensions, HAC structures, generators and parameters. Due to space limitations we present only one experiment, where we compare

Algorithm 3 The HAC estimation

Input:

1) (τ_{ij}^n) {...Kendall correlations matrix},
2) g {...an aggregation function},
3) $\mathcal{I} = \{1, ..., d\}$,
4) $z_i = u_i, i = 1, ..., d$,
5) Archimedean family based on generator ψ and corresponding τ_θ^{-1}

Estimation:

for $k = 1, ..., d-1$ **do**

1. $(i, j) := \underset{\tilde{i}<\tilde{j}, \tilde{i}\in\mathcal{I}, \tilde{j}\in\mathcal{I}}{\operatorname{argmax}} g((\tau^n_{\tilde{i}\tilde{j}})_{(\tilde{i},\tilde{j})\in\mathcal{U}_{\hat{C}}(z_{\tilde{i}})\times\mathcal{U}_{\hat{C}}(z_{\tilde{j}})})$
2. $\hat{\theta}_k := \tau_\theta^{-1}\big(g((\tau^n_{\tilde{i}\tilde{j}})_{(\tilde{i},\tilde{j})\in\mathcal{U}_{\hat{C}}(z_i)\times\mathcal{U}_{\hat{C}}(z_j)})\big)$
3. $\hat{\theta}_k := \min(\hat{\theta}_k, \hat{\theta}_i, \hat{\theta}_j)$
4. $z_{d+k} := C(u_i, u_j; \psi)$
5. $\mathcal{U}_{\hat{C}}(z_{d+k}) := \mathcal{U}_{\hat{C}}(z_i) \cup \mathcal{U}_{\hat{C}}(z_j)$
6. $\mathcal{I} := \mathcal{I} \cup \{d+k\} \backslash \{i, j\}$

end for

Output:

$\hat{\theta}_k, \mathcal{U}_{\hat{C}}(k), k = 1, ..., d-1$

the proposed method with the other previously mentioned methods on simulated data for $d = 5, 6, 7, 9$. We simulate 100 samples of size 500, i.e., 500 rows and d columns of simulated data for each sample, according to [11] for 4 copula models based on the Clayton generator. The first considered model is $((12)_{\frac{3}{4}}(3(45)_{\frac{4}{4}})_{\frac{3}{4}})_{\frac{2}{4}}$. The natural numbers in the model notation (as in [23]) are the indexes of the copula variables, i.e., 1,...5, the parentheses correspond to each $\mathcal{U}_C(\cdot)$ of individual copulas, i.e., $\mathcal{U}_C(\psi_1) = \{1, 2\}$,$\mathcal{U}_C(\psi_2) = \{4, 5\}$,$\mathcal{U}_C(\psi_3) = \{3, 4, 5\}$,$\mathcal{U}_C(\psi_4) = \{1, 2, 3, 4, 5\}$, and the subscripts are the model parameters, i.e, $(\theta_1, \theta_2, \theta_3, \theta_4) = (\frac{2}{4}, \frac{3}{4}, \frac{3}{4}, \frac{4}{4})$. Note that the indexes of the 4 generators could be permuted arbitrarily and the particular selection of their ordering serves just for better illustration. The other 3 models are given with analogous notation as $(1((23)_{\frac{5}{4}}(4(56)_{\frac{6}{4}})_{\frac{5}{4}})_{\frac{4}{4}})_{\frac{2}{4}}$, $(1((23)_{\frac{5}{4}}(4(5(67)_{\frac{7}{4}})_{\frac{6}{4}})_{\frac{5}{4}})_{\frac{4}{4}})_{\frac{2}{4}}$ and $((1(2(34)_{\frac{5}{4}})_{\frac{4}{4}})_{\frac{3}{4}}$ $((56)_{\frac{4}{4}}(7(89)_{\frac{5}{4}})_{\frac{4}{4}})_{\frac{3}{4}})_{\frac{2}{4}}$. The smallest difference between the parameters is set to $\frac{1}{4}$. As we revealed, while we experimented with different parameterizations, a larger difference in the parameters could hide the impact of the bias of the concerned methods on the structure determination, and the results obtained by different methods can be similar in some of those cases. Setting it to $\frac{1}{4}$ fully reveals the impact of the bias and clearly shows the difference among the methods.

The results for each model are shown in Table 1 and are separated by the double lines. As we are interested in binary copulas, we choose for the comparison the methods θ_{binary}, θ_{RML}, τ_{binary}, which return binary copula structures as their results. The first 2 methods are based on ML estimation technique, whereas the third method is based on the $\theta - \tau$ relationship. To get the results we used

their R implementation described in [25]. Our method, implemented in Matlab, is denoted as τ^{avg}_{binary}, i.e., the involved function g is selected to be the avg function due to the previously mentioned reasons. As θ_{RML} failed in most cases for $d \geq 7$, the results for the method for those dimensions are not presented.

Firstly, we assess the ability of the methods to determine the true copula structure correctly. This can be seen from the third and the fourth column. The third column shows 3 the most frequent structures obtained by the method (if the true structure was not the one of the 3 most frequent structures, then we show the 2 most frequent structures and the true structure) with average parameter values. The true structure is emphasized by bold text. The fourth column shows the frequency of the structures. τ^{avg}_{binary} clearly dominates in all four cases ($d = 5, 6, 7, 9$). The other methods show very poor ability to detect the correct structure, especially for $d \geq 7$, where, e.g., θ_{binary} did not return the correct structure for any among all 100 samples used.

Next, we assess the methods by means of goodness-of-fit. The results can be seen in the fifth and the sixth column, where the statistics $S^{(K)}, S^{(C)}$ (described in [6]) are computed on all bivariate margins and their maximum (the $S^{(K)}, S^{(C)}$ for the worst fitted bivariate margin) is shown. τ^{avg}_{binary} also dominates in all four cases. θ_{RML} shows also good results, but its time consumption for comparable results is considerably higher. The remaining methods show poor results, what is additionally illustrated by the discrepancy between the estimated average parameter values shown in the third column and the true parameter values.

The next two columns show the average Frobenius norm of the difference between the Kendall correlation matrix for the true model and the Kendall correlation matrix for the estimated model and the average Frobenius norm of the difference between the matrix of lower tail coefficients (cf. [22]) for the true model and the the matrix of lower tail coefficients for the estimated model (as in [23]). The comparison results are similar to the goodness-of-fit comparison. θ_{RML} shows slightly better results than τ^{avg}_{binary} and the remaining methods show significant discrepancy between the theoretical and the empirical quantities.

The last column shows the average computing times needed for a single data sample. τ^{avg}_{binary} is slightly better that the binary methods $\theta_{binary}, \tau_{binary}$, whereas θ_{RML} shows significantly higher time consumption, particularly for $d = 6$.

5 Conclusion

Copulas are a feasible tool for the modeling of complex patters. A popular alternative to Gaussian copulas, the hierarchical Archimedean copulas, are convenient copula models even in high dimensions due to their flexibility and rather limited number of parameters. Despite their popularity, a general approach for their estimation has been addressed only in one recently published paper[23], which proposes several methods for the estimation task.

We propose an alternative approach to structure determination and estimation of a hierarchical Archimedean copula, which combines the advantages and avoids the disadvantages of the previously mentioned methods in the terms of

Table 1. The results for the copula models for $d = 5, 6, 7, 9$. The columns contain method names; the 3 most frequent estimated structures with average parameter values; goodness-of-fit statistics $S^{(K)}, S^{(C)}$ (described in [6]); the Frobenius norms of the differences between estimated and true Kendall matrices and lower tail indices; the estimation time in s. The values in parenthesis are the corresponding standard deviations.

d	Method	Structure(s)	%	$S_n^{(K)}$	$S_n^{(C)}$	Avg. error in τ	λ_L	time (in s)
5	θ_{binary}	$(3((12)_{0.77}(45)_{1.01})_{0.76})_{0.24}$	79	2.1478 (0.5)	0.7206 (0.3)	0.3101 (0.025)	0.6306 (0.04)	0.1517 (0.04)
		$((\mathbf{12})_{\mathbf{0.69}}(\mathbf{3}(\mathbf{45})_{\mathbf{1.01}})_{\mathbf{0.72}})_{\mathbf{0.68}}$	**18**	0.4897 (0.21)	0.4089 (0.21)	0.1426 (0.024)	0.2893 (0.05)	
		$((12)_{0.61}(4(35)_{0.85})_{0.71})_{0.61}$	2	0.5546 (0.22)	0.2843 (0.04)	0.1208 (0.02)	0.2346 (0.04)	
	θ_{RML}	$((\mathbf{12})_{\mathbf{0.71}}(\mathbf{3}(\mathbf{45})_{\mathbf{1.00}})_{\mathbf{0.77}})_{\mathbf{0.54}}$	**52**	0.2102 (0.08)	0.2426 (0.11)	0.0511 (0.02)	0.1016 (0.05)	0.3616 (0.06)
		$((45)_{1.01}(3(12)_{0.79})_{0.72})_{0.62}$	43	0.4959 (0.28)	0.3290 (0.14)	0.1339 (0.018)	0.2704 (0.03)	
		$((12)_{0.80}(4(35)_{0.93})_{0.81})_{0.52}$	3	0.3090 (0.12)	0.2992 (0.09)	0.0973 (0.026)	0.1743 (0.05)	
	τ_{binary}	$((\mathbf{12})_{\mathbf{0.81}}(\mathbf{3}(\mathbf{45})_{\mathbf{1.04}})_{\mathbf{0.93}})_{\mathbf{0.89}}$	**46**	1.2082 (0.3)	0.5333 (0.22)	0.2751 (0.06)	0.5234 (0.11)	0.3055 (0.018)
		$(1(2(3(45)_{1.02})_{0.92})_{0.78})_{0.85}$	23	0.9928 (0.29)	0.4469 (0.18)	0.2332 (0.07)	0.4494 (0.12)	
		$(2(1(3(45)_{0.99})_{0.92})_{0.79})_{0.88}$	21	0.9659 (0.2)	0.4022 (0.16)	0.2443 (0.04)	0.4709 (0.08)	
	τ_{binary}^{avg}	$((\mathbf{12})_{\mathbf{0.76}}(\mathbf{3}(\mathbf{45})_{\mathbf{1.01}})_{\mathbf{0.75}})_{\mathbf{0.49}}$	**92**	0.1719 (0.06)	0.2372 (0.1)	0.0627 (0.028)	0.1208 (0.06)	0.1631 (0.0007)
		$((12)_{0.68}(5(34)_{0.95})_{0.87})_{0.52}$	3	0.1826 (0.05)	0.2141 (0.06)	0.0778 (0.016)	0.1362 (0.028)	
		$((12)_{0.74}(4(35)_{0.93})_{0.85})_{0.50}$	3	0.2106 (0.05)	0.3107 (0.14)	0.0829 (0.011)	0.1513 (0.019)	
6	θ_{binary}	$(1(4((23)_{1.28}(56)_{1.53})_{1.28})_{0.55})_{0.18}$	49	2.1014 (0.4)	0.8661 (0.34)	0.4078 (0.03)	0.7367 (0.05)	0.2674 (0.08)
		$(\mathbf{1}((\mathbf{23})_{\mathbf{1.16}}(\mathbf{4}(\mathbf{56})_{\mathbf{1.53}})_{\mathbf{1.24}})_{\mathbf{1.15}})_{\mathbf{0.21}}$	**25**	1.1039 (0.3)	0.4969 (0.27)	0.2507 (0.04)	0.4839 (0.05)	
		$((14)_{0.56}((23)_{1.24}(56)_{1.49})_{1.24})_{0.56}$	22	1.7606 (0.4)	0.7776 (0.27)	0.3101 (0.018)	0.5375 (0.03)	
	θ_{RML}	$(\mathbf{1}((\mathbf{23})_{\mathbf{1.19}}(\mathbf{4}(\mathbf{56})_{\mathbf{1.53}})_{\mathbf{1.28}})_{\mathbf{1.00}})_{\mathbf{0.50}}$	**48**	0.1965 (0.07)	0.2945 (0.12)	0.0506 (0.019)	0.0884 (0.04)	3.4299 (2.13)
		$(1((56)_{1.52}(4(23)_{1.29})_{1.21})_{1.08})_{0.51}$	44	0.3149 (0.13)	0.3055 (0.14)	0.1026 (0.02)	0.1617 (0.04)	
		$(1(2(3(4(56)_{1.68})_{1.40})_{1.12})_{1.04})_{0.56}$	2	0.2016 (0.08)	0.3781 (0.05)	0.1006 (0.04)	0.1601 (0.08)	
	τ_{binary}	$(1(2(3(4(56)_{1.56})_{1.49})_{1.39})_{1.39})_{0.70}$	40	0.6187 (0.16)	0.4378 (0.16)	0.2478 (0.06)	0.3970 (0.1)	0.4983 (0.02)
		$(1(3(2(4(56)_{1.53})_{1.48})_{1.41})_{1.40})_{0.71}$	32	0.6652 (0.17)	0.4294 (0.15)	0.2541 (0.05)	0.4073 (0.07)	
		$(\mathbf{1}((\mathbf{23})_{\mathbf{1.37}}(\mathbf{4}(\mathbf{56})_{\mathbf{1.57}})_{\mathbf{1.52}})_{\mathbf{1.36}})_{\mathbf{0.73}}$	**11**	0.6411 (0.13)	0.4015 (0.13)	0.2474 (0.06)	0.4077 (0.1)	
	τ_{binary}^{avg}	$(\mathbf{1}((\mathbf{23})_{\mathbf{1.27}}(\mathbf{4}(\mathbf{56})_{\mathbf{1.54}})_{\mathbf{1.25}})_{\mathbf{1.00}})_{\mathbf{0.51}}$	**84**	0.1753 (0.06)	0.2749 (0.19)	0.0745 (0.029)	0.1263 (0.05)	0.2470 (0.06)
		$(1((23)_{1.21}(5(46)_{1.49})_{1.36})_{1.04})_{0.50}$	4	0.1535 (0.05)	0.3090 (0.12)	0.1017 (0.04)	0.1640 (0.08)	
		$(1(3(2(4(56)_{1.62})_{1.38})_{1.20})_{1.06})_{0.54}$	3	0.1657 (0.01)	0.1743 (0.05)	0.1174 (0.029)	0.1738 (0.04)	
7	θ_{binary}	$((14)_{0.52}((23)_{1.24}(5(67)_{1.74})_{1.41})_{1.24})_{0.52}$	48	2.3349 (0.5)	1.0978 (0.6)	0.3810 (0.03)	0.6637 (0.06)	0.3827 (0.03)
		$(1(4((23)_{1.25}(5(67)_{1.77})_{1.43})_{1.24})_{0.48})_{0.14}$	18	2.7023 (0.4)	1.2764 (0.6)	0.5236 (0.04)	0.9294 (0.06)	
		$(1((45)_{1.17}((23)_{1.35}(67)_{1.77})_{1.34})_{1.16})_{0.19}$	16	1.3054 (0.4)	0.5234 (0.2)	0.3388 (0.03)	0.6068 (0.03)	
	τ_{binary}	$(1(2(3(4(5(67)_{1.79})_{1.73})_{1.63})_{1.46})_{1.45})_{0.70}$	45	0.8215 (0.19)	0.4797 (0.17)	0.3173 (0.07)	0.4776 (0.11)	0.7435 (0.021)
		$(1(3(2(4(5(67)_{1.81})_{1.76})_{1.66})_{1.47})_{1.46})_{0.72}$	32	0.8420 (0.2)	0.5341 (0.19)	0.3333 (0.07)	0.5047 (0.1)	
		$(\mathbf{1}((\mathbf{23})_{\mathbf{1.48}}(\mathbf{4}(\mathbf{5}(\mathbf{67})_{\mathbf{1.85}})_{\mathbf{1.85}})_{\mathbf{1.67}})_{\mathbf{1.48}})_{\mathbf{0.67}}$	**3**	0.8633 (0.1)	0.4852 (0.11)	0.3373 (0.14)	0.5019 (0.2)	
	τ_{binary}^{avg}	$(\mathbf{1}((\mathbf{23})_{\mathbf{1.27}}(\mathbf{4}(\mathbf{5}(\mathbf{67})_{\mathbf{1.80}})_{\mathbf{1.52}})_{\mathbf{1.25}})_{\mathbf{1.00}})_{\mathbf{0.50}}$	**77**	0.1877 (0.05)	0.3065 (0.15)	0.0895 (0.04)	0.1472 (0.07)	0.3255 (0.07)
		$(1((23)_{1.26}(4(7(56)_{1.65})_{1.55})_{1.28})_{1.02})_{0.49}$	6	0.1854 (0.05)	0.3338 (0.2)	0.0902 (0.018)	0.1394 (0.04)	
		$(1((23)_{1.25}(4(6(57)_{1.55})_{1.42})_{1.25})_{1.02})_{0.50}$	5	0.2094 (0.08)	0.4709 (0.26)	0.0951 (0.027)	0.1514 (0.06)	
9	θ_{binary}	$((17)_{0.51}((2(34)_{1.25})_{0.90}((56)_{1.02}(89)_{1.26})_{1.02})_{0.89})_{0.51}$	58	1.6487 (0.4)	0.7410 (0.26)	0.4771 (0.04)	0.9144 (0.08)	0.7862 (0.06)
		$(1((2(34)_{1.25})_{0.86}((56)_{0.96}(7(89)_{1.33})_{1.01})_{0.96})_{0.86})_{0.13}$	11	3.5263 (0.6)	0.9699 (0.3)	0.6364 (0.03)	1.1800 (0.05)	
		$(1((56)_{0.91}((2(34)_{1.32})_{0.96}(7(89)_{1.30})_{0.99})_{0.94})_{0.72})_{0.13}$	10	4.1839 (0.5)	1.2621 (0.4)	0.6296 (0.024)	1.1628 (0.05)	
	τ_{binary}	$((1(2(34)_{1.34})_{1.22})_{1.06}(6(5(7(89)_{1.28})_{1.22})_{1.06})_{1.06})_{1.11}$	15	2.6079 (0.4)	0.9986 (0.29)	0.7463 (0.09)	1.3381 (0.13)	1.4654 (0.02)
		$((1(2(34)_{1.31})_{1.24})_{1.12}(5(6(7(89)_{1.29})_{1.23})_{1.12})_{1.11})_{1.12}$	13	2.3948 (0.3)	0.9770 (0.28)	0.7620 (0.11)	1.3583 (0.15)	
		$((\mathbf{1}(\mathbf{2}(\mathbf{34})_{\mathbf{1.21}})_{\mathbf{1.17}})_{\mathbf{1.04}}((\mathbf{56})_{\mathbf{1.06}}(\mathbf{7}(\mathbf{89})_{\mathbf{1.13}})_{\mathbf{1.10}})_{\mathbf{1.00}})_{\mathbf{1.05}}$	**4**	2.3784 (0.29)	0.8742 (0.3)	0.6753 (0.15)	1.2305 (0.23)	
	τ_{binary}^{avg}	$((\mathbf{1}(\mathbf{2}(\mathbf{34})_{\mathbf{1.27}})_{\mathbf{0.99}})_{\mathbf{0.75}}((\mathbf{56})_{\mathbf{1.00}}(\mathbf{7}(\mathbf{89})_{\mathbf{1.28}})_{\mathbf{1.01}})_{\mathbf{0.75}})_{\mathbf{0.50}}$	**81**	0.2491 (0.07)	0.3328 (0.12)	0.1134 (0.04)	0.2096 (0.09)	0.4851 (0.0019)
		$((1(3(24)_{1.17})_{1.07})_{0.72}((56)_{0.97}(7(89)_{1.27})_{0.99})_{0.76})_{0.49}$	4	0.2264 (0.06)	0.1860 (0.04)	0.1264 (0.06)	0.2400 (0.14)	
		$((1(2(34)_{1.41})_{1.07})_{0.84}((56)_{1.05}(9(78)_{1.26})_{1.12})_{0.83})_{0.56}$	3	0.1921 (0.03)	0.3401 (0.21)	0.1444 (0.022)	0.2576 (0.04)	

the correctly determined structures ratio, the goodness-of-fit of the estimates, and computation time. This is confirmed in the experiments on simulated data performed for different dimensions and copula models. The proposed method should be preferred to the other mentioned methods and is particularly attractive in applications, where a good approximation and computational efficiency are both crucial issues.

Acknowledgment

The research reported in this paper has been supported by the Czech Science Foundation (GA ČR) grant 13-17187S.
This work has been originally published in New Frontiers in Mining Complex Patterns, Second International Workshop, NFMCP 2013, Held in Conjunction with ECML-PKDD 2013, Prague, Czech Republic, September 27, 2013, Revised Selected Papers, pages 132-147, 2014, see `http://link.springer.com/chapter/10.1007%2F978-3-319-08407-7_9#`. For a BibTex citation, see `http://suzelly.opf.slu.cz/~gorecki/en/research.php`.

References

1. D. Berg. Copula goodness-of-fit testing: an overview and power comparison. *The European Journal of Finance*, 15(7-8):675–701, 2009.
2. X. Chen, Y. Fan, and A. J. Patton. Simple tests for models of dependence between multiple financial time series, with applications to us equity returns and exchange rates. 2004. Discussion paper 483, Financial Markets Group, London School of Economics.
3. E. Cuvelier and M. Noirhomme-Fraitur. Clayton copula and mixture decomposition. In *Applied Stochastic Models and Data Analysis*, ASMDA'05, Brest, 2005.
4. C. Genest and A. Favre. Everything you always wanted to know about copula modeling but were afraid to ask. *Hydrol. Eng.*, 12:347 – 368, 2007.
5. C. Genest and L.-P. Rivest. Statistical inference procedures for bivariate archimedean copulas. *Journal of the American statistical Association*, 88(423):1034–1043, 1993.
6. C. Genest, B. Rémillard, and D. Beaudoin. Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics*, 44(2):199 – 213, 2009.
7. Y. González-Fernández and M. Soto. copulaedas: An r package for estimation of distribution algorithms based on copulas. *CoRR*, abs/1209.5429, 2012.
8. J. Górecki and M. Holeňa. An alternative approach to the structure determination of hierarchical archimedean copulas. 31st International Conference on Mathematical Methods in Economics 2013, pages 201 – 206, Jihlava, 2013.
9. M. Hofert. Construction and sampling of nested archimedean copulas. In P. Jaworski, F. Durante, W. K. Hardle, and T. Rychlik, editors, *Copula Theory and Its Applications*, volume 198 of *Lecture Notes in Statistics*, pages 147–160. Springer Berlin Heidelberg, 2010.
10. M. Hofert. *Sampling Nested Archimedean Copulas with Applications to CDO Pricing*. PhD thesis, Ulm University, 2010.

11. M. Hofert. Efficiently sampling nested archimedean copulas. *Computational Statistics and Data Analysis*, 55(1):57 – 70, 2011.
12. M. Hofert, M. Mächler, and A. J. McNeil. Estimators for archimedean copulas in high dimensions: A comparison. *arXiv preprint arXiv:1207.1708*, 2012.
13. S.-C. Kao, A. R. Ganguly, and K. Steinhaeuser. Motivating complex dependence structures in data mining: A case study with anomaly detection in climate. *Data Mining Workshops, International Conference on*, 0:223–230, 2009.
14. S.-C. Kao and R. S. Govindaraju. Trivariate statistical analysis of extreme rainfall events via plackett family of copulas. *Water Resour. Res.*, 44, 2008.
15. I. Kojadinovic. Hierarchical clustering of continuous variables based on the empirical copula process and permutation linkages. *Computational Statistics & Data Analysis*, 54(1):90 – 108, 2010.
16. I. Kojadinovic and J. Yan. Modeling multivariate distributions with continuous margins using the copula r package. *Journal of Statistical Software*, 34(9):1–20, 2010.
17. G. Kuhn, S. Khan, A. R. Ganguly, and M. L. Branstetter. Geospatial-temporal dependence among weekly precipitation extremes with applications to observations and climate model simulations in south america. *Adv. Water Resour.*, 30(12):2401–2423, 2007.
18. F. Lascio and S. Giannerini. A copula-based algorithm for discovering patterns of dependent observations. *Journal of Classification*, 29:50–75, 2012.
19. R. Maity and D. N. Kumar. Probabilistic prediction of hydroclimatic variables with nonparametric quantification of uncertainty. *J. Geophys. Res.*, 113, 2008.
20. A. J. McNeil. Sampling nested archimedean copulas. *Journal of Statistical Computation and Simulation*, 78(6):567–581, 2008.
21. A. J. McNeil and J. Nešlehová. "multivariate archimedean copulas, d-monotone functions and l1-norm symmetric distributions. *The Annals of Statistics*, 37:3059 – 3097, 2009.
22. R. Nelsen. *An Introduction to Copulas*. Springer, 2nd edition, 2006.
23. O. Okhrin, Y. Okhrin, and W. Schmid. On the structure and estimation of hierarchical archimedean copulas. *Journal of Econometrics*, 173(2):189 – 204, 2013.
24. O. Okhrin, Y. Okhrin, and W. Schmid. Properties of hierarchical archimedean copulas. *Statistics & Risk Modeling*, 30(1):21–54, 2013.
25. O. Okhrin and A. Ristig. Hierarchical Archimedean copulae: The HAC package. Discussion paper 2012, 036, CRC 649, Economic Risk, 2012.
26. M. Rey and R. V. Copula mixture model for dependency-seeking clustering. In *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, Edinburgh, Scotland, UK, 2012.
27. C. Savu and M. Trede. Goodness-of-fit tests for parametric families of archimedean copulas. *Quantitative Finance*, 8(2):109–116, 2008.
28. C. Savu and M. Trede. Hierarchies of archimedean copulas. *Quantitative Finance*, 10:295–304, 2010.
29. A. Sklar. Fonctions de répartition a n dimensions et leurs marges. *Publ. Inst. Stat. Univ. Paris*, 8:229 – 231, 1959.
30. L. Wang, X. Guo, Z. J., and Y. Hong. Copula estimation of distribution algorithms based on exchangeable archimedean copula. *International Journal of Computer Applications in Technology*, 43:13 – 20, 2012.
31. A. Yuan, G. Chen, Z.-C. Zhou, G. Bonney, and C. Rotimi. Gene copy number analysis for family data using semiparametric copula model. *Bioinform Biol Insights.*, 2.:343–355, 2008.