SFB 649 Discussion Paper 2012-036

Hierarchical Archimedean Copulae: The HAC Package

Ostap Okhrin*
Alexander Ristig*

* Humboldt-Universität zu Berlin, Germany

This research was supported by the Deutsche Forschungsgemeinschaft through the SFB 649 "Economic Risk".

http://sfb649.wiwi.hu-berlin.de
ISSN 1860-5664

SFB 649, Humboldt-Universität zu Berlin
Spandauer Straße 1, D-10178 Berlin

Hierarchical Archimedean Copulae: The HAC Package

Ostap Okhrin[*]
Humboldt-Universität zu Berlin

Alexander Ristig[*]
Humboldt-Universität zu Berlin

Abstract

This paper aims at explanation of the R-package **HAC**, which provides user friendly methods for dealing with high-dimensional hierarchical Archimedean copulae (HAC). A computationally efficient estimation procedure allows to recover the structure and the parameters of HACs from data. In addition, arbitrary HACs can be constructed to sample random vectors and to compute the values of the corresponding cumulative distribution as well as density functions. Accurate graphics of the important characteristics of the package's object `hac` can be produced by the generic `plot` function.

JEL classification: C51, C87.
Keywords: copula, R, hierarchical Archimedean copula (HAC).

1. Introduction

The success of copulae in applied statistics started at the end of the 90th, when Embrechts, McNeil, and Straumann (1999) introduced copula to empirical finance in the context of risk management. Nowadays, quantitative orientated sciences like biostatistics and hydrology were doing attempts of measuring the dependence of random variables with copulae, e.g., Lakhal-Chaieb (2010); Acar, Craiu, and Yao (2011); Bárdossy (2006); Genest and Favre (2007); Bárdossy and Li (2008). In finance, copulae became a standard tool, explicitly on VaR measurement and in valuation of structured credit portfolios, see Mendes and Souza (2004); Junker and May (2005) and Li (2000) respectively. This paper targets to provide the necessary tools for academics and practitioners for simple and effective use of HAC in their analysis.

Copula is the function splitting the multivariate distribution into the margins and a pure dependency component. Formally copulae were introduced in Sklar (1959) stating that if F is an arbitrary d-dimensional continuous distribution function of the random variables $X_1, \dots, X_d$, then the associated copula is unique and defined as a continuous function $C : [0,1]^d \to [0,1]$ which satisfies the equality

$$C(u_1, \dots, u_d) = F\{F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)\}, \quad u_1, \dots, u_d \in [0,1],$$

where $F_1^{-1}(\cdot), \dots, F_d^{-1}(\cdot)$ are the quantile functions of the corresponding marginal distributions $F_1(x_1), \dots, F_d(x_d)$. For an overview and recent developments of copulae we refer to

[*]The financial support from the Deutsche Forschungsgemeinschaft via SFB 649 Ökonomisches Risiko, Humboldt-Universität zu Berlin is gratefully acknowledged.

Nelsen (2006), Cherubini, Luciano, and Vecchiato (2004), Joe (1997) and Jaworski, Durante, Härdle, and Rychlik (2010). If F belongs to the class of elliptical distributions, then C is an elliptical copula, which in most cases cannot be given explicitly, because the distribution function F and the inverse marginal distributions F_i usually have integral representations. One of the classes that overcomes this drawback of elliptical copulae is the class of Archimedean copulae, which however is very restrictive even for moderate dimensions. Among other packages dealing with Archimedean copula, we would like to mention the **copula** and the **fCopulae** package, c.f. Yan (2007), Kojadinovic and Yan (2010) and Wuertz *et al.* (2009). HAC generalizes the concept of simple Archimedean copulae by substituting (a) marginal distribution(s) by a further HAC. This class is thoroughly analyzed in Whelan (2004); Savu and Trede (2010); Embrechts, Lindskog, and McNeil (2003); Hofert (2011). The first sampling algorithms for special HAC structures were provided by the **QRMlib** package of McNeil and Ulman (2011), which last available version on CRAN is 1.4.5.1, which is not updated anymore. Hofert and Mächler (2011) presented the comprehensive **nacopula** package which among other features allows to sample from arbitrary HAC and was integrated into the package **copula** from version 0.8-1. The central contribution of the **HAC** package is the estimation of the parameter and the structure for this class of copulae, as discussed in Okhrin, Okhrin, and Schmid (2011a), including a simple and intuitive representation of HACs as R-objects of the class `hac`. The main estimation procedure relies on a multi-stage Maximum Likelihood (ML) procedure, which determines the parameter and the structure simultaneously. This elegant procedure endows the estimator with the usual asymptotic properties but avoids the computationally intensive one-step ML estimation, which is also implemented for a predetermined structure. Besides, the package offers functions to produce graphics of the copula's structure, to sample random vectors from a given copula and to compute values of the corresponding distribution and density.

The paper is organized as follows. The next section describes shortly the theoretical aspects of HAC and its estimation. Section 3 describes the functions of the **HAC** package and section 4 presents a simulation study. Section 5 concludes.

2. Hierarchical Archimedean copulae

As mentioned above, the large class of copulae, which can describe tail dependency, non-ellipticity, and what is most important, has close form representation

$$C(u_1, \dots, u_d; \theta) = \phi_\theta\{\phi_\theta^{-1}(u_1) + \cdots + \phi_\theta^{-1}(u_d)\}, \quad u_1, \dots, u_d \in [0,1], \tag{1}$$

where $\phi_\theta \in \mathfrak{L} = \{\phi_\theta : [0;\infty) \to [0,1] \,|\, \phi_\theta(0) = 1,\ \phi_\theta(\infty) = 0;\ (-1)^j \phi_\theta^{(j)} \geq 0;\ j = 1, \dots, d-2\}$ and $(-1)^{d-2}\phi_\theta^{(d-2)}(x)$ being non-decreasing and convex on $[0,\infty)$, is the class of Archimedean copulae. The function ϕ is called the generator of the copula and commonly depends on a single parameter θ. For example, the Gumbel generator is given by $\phi_\theta = \exp(-x^{1/\theta})$ for $0 \leq x < \infty$, $1 \leq \theta < \infty$. Detailed reviews of the properties of Archimedean copulae can be found in McNeil and Nešlehová (2009) as well as in Joe (1996).

A disadvantage of Archimedean copulae is the fact that the multivariate dependency structure is very restricted, since it typically depends on a single parameter of the generator function ϕ. Moreover, the rendered dependency is symmetric with respect to the permutation of variables, i.e., the distribution is exchangeable. HACs (also called nested Archimedean copulae) overcome this problem by considering the compositions of simple Archimedean copulae. For example, the special case of a four-dimensional HAC can be given by

$$\begin{aligned} C(u_1,\dots,u_4) &= C_1\{C_2(u_1,\dots,u_3),u_4\} = \phi_3\{\phi_3^{-1}\circ C_2(u_1,\dots,u_3)+\phi_3^{-1}(u_4)\} \\ &= \phi_3\{\phi_3^{-1}\circ\phi_2[\phi_2^{-1}\{C_3(u_1,u_2)\}+\phi_2^{-1}(u_3)]+\phi_3^{-1}(u_4)\}. \qquad (2) \end{aligned}$$

The form (2) is called fully nested HAC. The composition can be applied recursively using different segmentations of variables leading to more complex HACs. For notational convenience we denote the structure of a HAC by $s = \{(\dots(i_1\dots i_{j_1})\dots(\dots)\dots)\}$, where $i_\ell \in \{1,\dots,d\}$ is a reordering of the indices of the variables and s_j denotes the structure of subcopulae with $s_d = s$. Further, let the d-dimensional HAC be denoted by $C(u_1,\dots,u_d; s, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes the vector of feasible dependency parameters. Thus, the fully nested HAC, given in (2), can be expressed as

$$\begin{aligned} &C(u_1,\dots,u_4; s=(((12)3)4),\boldsymbol{\theta}) = C\{u_1,\dots,u_4;((s_3)4),(\theta_1,\dots,\theta_3)^\top\} \\ =\ &\phi_{\theta_3}(\phi_{\theta_3}^{-1}\circ C\{u_1,\dots,u_3;((s_2)(3)),(\theta_1,\theta_2)^\top\}+\phi_{\theta_3}^{-1}(u_4)). \end{aligned}$$

Figure 1 presents the four-dimensional fully and partially nested Archimedean copula.

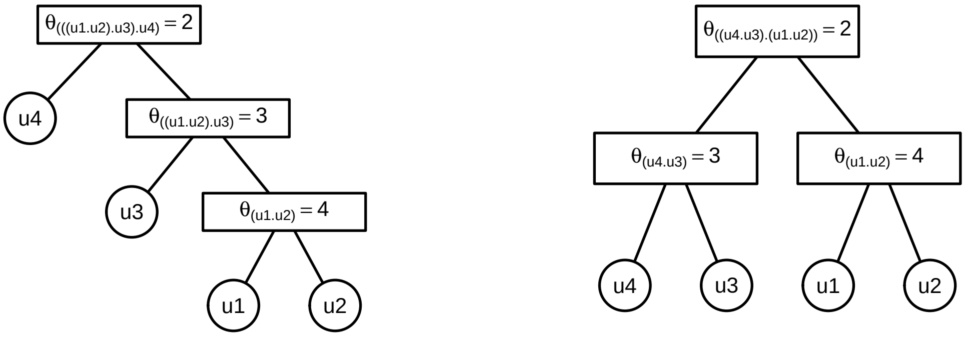

Figure 1: Fully and partially nested Archimedean copulae of dimension $d = 4$ with structures $s = (((12)3)4)$ on the left and $s = ((43)(12))$ on the right.

HACs can adopt arbitrary elaborate structures s. This makes it a very flexible and simultaneously parsimonious distribution model. The generators ϕ_{θ_i} within a HAC can come either from a single generator family or from different generator families. If the ϕ_{θ_i}'s belong to the same family, then the required complete monotonicity of $\phi_{\theta_{i+1}}^{-1}\circ\phi_{\theta_i}$ usually imposes some constraints on the parameters $\theta_1,\dots,\theta_{d-1}$. Theorem 4.4 of McNeil (2008) provides sufficient conditions on the generator functions to guarantee that C is a copula. It holds that if $\phi_{\theta_i} \in \mathfrak{L}$, for $i = 1,\dots,d-1$, and $\phi_{\theta_{i+1}}^{-1}\circ\phi_{\theta_i}$ have completely monotone derivatives, then C is a copula for $d \geq 2$. For the majority of generators a feasible HAC requires decreasing parameters from the highest to the lowest hierarchical level. However, in the case of different families within a single HAC, the condition of complete monotonicity is not always fulfilled, see Hofert (2011). In our study, we consider HAC only with generators from the same family. If we use the same single-parameter generator function on each level, but with a different value of θ, we may specify the whole distribution with at most $d-1$ parameters. From this point of view, the HAC approach can be seen as an alternative to covariance driven models. But for each HAC not only the parameters are unknown, but also the structure has to be determined. One possible procedure is to enumerate and to estimate all possible HACs. Using a suitable goodness-of-fit test, the optimal structure then can be determined. This approach is however unrealistic in practice, because the variety of different structures is enormously large even in moderate dimensions. Okhrin *et al.* (2011a) suggest a computationally efficient procedure, which allows to estimate HACs recursively. The **HAC** package provides this method for estimating the HAC parameters and structure in a user-friendly way.

2.1. Estimation of HAC

In the most cases the discussion is constrained to binary copulae, i.e., at each level of the hierarchy only two variables are joined together. The whole procedure can be written in the recursive way, where at the first iteration step we fit a bivariate copula to every couple of the variables. The couple of variables with strongest dependency is selected. We denote the respective estimator of the parameter at the first level by $\hat{\theta}_1$ and the set of indices of the variables by I_1. The selected couple is joined together to define the pseudo-variable $Z_{I_1} \overset{\text{def}}{=} C\{(I_1); \hat{\theta}_1, \phi_1\}$. At the next step, we proceed in the same way by considering the remaining variables and the new pseudo-variable as the new set of variables. This procedure allows us to determine the estimated structure of the copula and if the restrictions on the parameters are fulfilled always leads to a feasible copula function with $d-1$ parameters. Nevertheless, if the true copula is not binary, the procedure might return a slightly misspecified structure. Despite of a difference in the structures, the difference in the distribution functions is in general minor. To allow more sophisticated structures, we aggregate the variables of the estimated copula afterwards, if the absolute value of the difference of two successive nodes is smaller than a fixed small threshold, i.e., $\theta_1 - \theta_2 < \varepsilon$, with $\theta_1 > \theta_2$, as suggested by Okhrin *et al.* (2011a).

For better understanding, let us consider a three-dimensional example with u_j, $j = 1\ldots,3$, being uniformly distributed on $[0,1]$. All possible pairs $C_{(12)}(u_1, u_2, \hat{\theta}_{(12)})$, $C_{(13)}(u_1, u_3, \hat{\theta}_{(13)})$ and $C_{(23)}(u_2, u_3, \hat{\theta}_{(23)})$ are estimated by regular ML, see Franke, Härdle, and Hafner (2011). To compare the strengths of the fit one can use goodness-of-fit tests, which are however computationally complicated and do not necessarily lead to a function which will be a copula on the final level of aggregation due to the restrictions on θ. For that reason we compare simply the parameters $\hat{\theta}_{(12)}$, $\hat{\theta}_{(13)}$ and $\hat{\theta}_{(23)}$. This is due to the fact that for the most Archimedean copulae, the larger the parameters the stronger is the dependency (the larger is the parameter the larger is Kendalls τ correlation coefficient). Let the strongest dependence be in the first pair $\hat{\theta}_1 \overset{\text{def}}{=} \hat{\theta}_{(12)} = \max\{\hat{\theta}_{(12)}, \hat{\theta}_{(13)}, \hat{\theta}_{(23)}\}$, then $I_1 = \{1,2\}$ and we introduce the pseudo-variable $Z_1 \overset{\text{def}}{=} C_1(I_1; \hat{\theta}_1) = C_1(u_1, u_2; \hat{\theta}_{(12)})$. On the next and final step for this example we join together u_3 and Z_1. The theoretical validation is also reported by Proposition 1 of Okhrin, Okhrin, and Schmid (2011b) stating that HAC can be uniquely recovered from the marginal distribution functions and all bivariate copula functions.

In practice, the marginal distributions F_j, $j = 1, \ldots, d$, are either parametrically or non-parametrically estimated in advance, whereby $\widehat{F}_j(\cdot)$ is an estimator of the marginal cdf F_j. Accordingly, the marginal densities $\hat{f}_j(\cdot), j = 1, \ldots, d$, are estimated by an appropriate kernel density estimator. If we estimate the margins parametrically then $\widehat{F}_j(\cdot) = F_j(\cdot, \hat{\boldsymbol{\alpha}}_j)$, where $\boldsymbol{\alpha}_j$ denotes the vector of parameters of the j-th margin.

The estimation of the copula parameters on each step of the iteration can be sketched as follows: at the first stage, we estimate the parameter of the copula at the first hierarchical level assuming that the marginal distributions are known. At further stages the next level copula parameter is estimated assuming that the margins as well as the copula parameters at lower levels are known. Let $\mathbf{X} = \{x_{ij}\}^\top$ be the respective sample, for $i = 1, \dots, n$, $j = 1, \dots, d$, and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{d-1})^\top$ be the parameters of the copula starting with the lowest up to the highest level. The multi-stage ML estimator $\widehat{\boldsymbol{\theta}}$ solves the system

$$\left(\frac{\partial \mathcal{L}_1}{\partial \theta_1}, \dots, \frac{\partial \mathcal{L}_{d-1}}{\partial \theta_{d-1}}\right)^\top = \mathbf{0}, \tag{3}$$

$$\begin{aligned} \text{where} \quad \mathcal{L}_j &= \sum_{i=1}^{n} l_j(\mathbf{X}_i), \ \text{ for } j = 1, \dots, d-1, \\ l_j(\mathbf{X}_i) &= \log \left\{ c\big[\{\widehat{F}_m(x_{im})\}_{m \in s_j};\, s_j, \theta_j\big] \prod_{m \in s_j} \hat{f}_m(x_{im}) \right\} \\ & \quad \text{for } j = 1, \dots, d-1,\ i = 1, \dots, n, \end{aligned}$$

where s_j is referred to the two (pseudo)-variables considered at the j-th estimation stage. Note, a d-dimensional density f can be split in the copula density c and the product of the marginal densities. Chen and Fan (2006) and Okhrin *et al.* (2011a) provide asymptotic behaviour of the estimates. As long as the structure is determined through grouping binary structures, it seems to be appropriate to estimate Kendall's τ at each step of the iteration and exploit the bivariate relationship between Archimedean copulae and Kendall's $\tau(\cdot)$, implied through Proposition 1.1 of Genest and Rivest (1993), see table 2. On the other hand, the asymptotic theory for Kendall's τ is usually restricted to the two-dimensional case and cannot be carried over to a higher-dimensional framework as necessary for the considered purpose. Moreover, the copula parameters θ_j,$j = 1, \dots, d-1$, estimated with Kendall's τ cannot be guaranteed to be increasing from the lowest to the highest hierarchical level and therefore, the estimated copula can fail to be a properly defined cdf. In the ML setup, this problem is tackled by shortening the feasible parameter space.

3. Applications of HAC

Core of the **HAC** package is the function `estimate.copula` estimating the parameters and determining the structure for given data. Let us consider a dataset from Yahoo! Finance consisting of the log-returns of four oil corporations: Chevron Corporation (`CVX`), Exxon Mobil Corporation (`XOM`), Royal Dutch Shell (`RDSA`) and Total (`FP`), covering $n = 283$ observations from 20110202 to 20120319. Time dependencies are removed by usual ARMA-GARCH models, whose standardized residuals are employed as `sample` in the subsequent analysis.

```
> library(HAC)
> t      = Sys.time()
> result = estimate.copula(sample, margins = "edf")
> Sys.time() - t
```

```
Time difference of 0.04680014 secs
```

		$z_{i,(\texttt{CVX.XOM})} \overset{\text{def}}{=} \widehat{C}\{\widehat{F}_{\texttt{CVX}}(x_{i,\texttt{CVX}}), \widehat{F}_{\texttt{XOM}}(x_{i,\texttt{XOM}})\}$		$z_{i,(\texttt{FP.RDSA})} \overset{\text{def}}{=} \widehat{C}\{\widehat{F}_{\texttt{FP}}(x_{i,\texttt{FP}}), \widehat{F}_{\texttt{RDSA}}(x_{i,\texttt{RDSA}})\}$
$(\texttt{CVX.FP}) \rightsquigarrow \hat{\theta}_{(\texttt{CVX.FP})}$				
$(\texttt{CVX.XOM}) \rightsquigarrow \hat{\theta}_{(\texttt{CVX.XOM})}$	best fit $(\texttt{CVX.XOM})$ $\Rightarrow$	$(\texttt{CVX.XOM})\texttt{FP} \rightsquigarrow \hat{\theta}_{(\texttt{CVX.XOM})\texttt{FP}}$	best fit $(\texttt{FP.RDSA})$ $\Rightarrow$	$((\texttt{CVX.XOM})(\texttt{FP.RDSA}))$
$(\texttt{FP.RDSA}) \rightsquigarrow \hat{\theta}_{(\texttt{FP.RDSA})}$		$(\texttt{CVX.XOM})\texttt{RDSA} \rightsquigarrow \hat{\theta}_{(\texttt{CVX.XOM})\texttt{RDSA}}$		$\rightsquigarrow \hat{\theta}_{((\texttt{CVX.XOM})(\texttt{FP.RDSA}))}$
$(\texttt{FP.XOM}) \rightsquigarrow \hat{\theta}_{(\texttt{FP.XOM})}$		$(\texttt{FP.RDSA}) \rightsquigarrow \hat{\theta}_{(\texttt{FP.RDSA})}$		
$(\texttt{RDSA.XOM}) \rightsquigarrow \hat{\theta}_{(\texttt{RDSA.XOM})}$				

Table 1: The estimation procedure in practice.

```
> result

Class: hac
Generator: Gumbel
((FP.RDSA)_{2.1}.(XOM.CVX)_{2.83})_{1.83}
```

The returned object `result` is of class `hac`, whose properties are explored below.

The multi-step estimation procedure is illustrated in table 3 for the four-dimensional example from above. At the lowest hierarchical level, the parameter of all bivariate copulae are estimated. The couple $(X_{\texttt{CVX}}, X_{\texttt{XOM}})$ produces the strongest dependency, hence the best fit. Then, the pseudo variable $Z_{(\texttt{CVX.XOM})} \overset{\text{def}}{=} \phi_{\hat{\theta}_{(\texttt{CVX.XOM})}}\left[\phi^{-1}_{\hat{\theta}_{(\texttt{CVX.XOM})}}\left\{\widehat{F}_{\texttt{XOM}}\left(X_{\texttt{XOM}}\right)\right\} + \phi^{-1}_{(\hat{\theta}_{\texttt{CVX.XOM}})}\left\{\widehat{F}_{\texttt{CVX}}\left(X_{\texttt{CVX}}\right)\right\}\right]$ is defined and the corresponding realizations are computed. The involved variables $X_{\texttt{XOM}}$ and $X_{\texttt{CVX}}$ are substituted by this pseudo variable in the dataset. At the next nesting level the parameters of all bivariate subsets are estimated and the variables $X_{\texttt{FP}}$ and $X_{\texttt{RDSA}}$ exhibit the best fit. Finally, the realizations of the remaining random variables $Z_{(\texttt{CVX.XOM})}$ and $Z_{(\texttt{FP.RDSA})}$ are grouped at the highest level of the hierarchy, where $Z_{(\texttt{FP.RDSA})}$ is defined analogously to $Z_{(\texttt{CVX.XOM})}$.

In general, `estimate.copula` includes the following arguments:

```
> names(formals(estimate.copula))

 [1] "X"          "type"       "method"     "hac"        "epsilon"
 [6] "agg.method" "margins"    "theta.eps"  "na.rm"      "max.min"
[11] "..."
```

The whole procedure is divided in three (optional) computational blocks. First, the margins are specified. Secondly, the copula parameter, $\boldsymbol{\theta}$, is estimated through the multi-stage procedure as explained above and finally the HAC is checked for aggregation possibilities. The `margins` of the $(n \times d)$ data matrix, `X`, are assumed to follow the standard Uniform distribution by default, i.e., `margins = NULL`, but the function permits non-uniformly distributed data as input, if the argument `margins` is specified. The marginal distributions can be determined non-parametrically, `margins = "edf"`, or in a parametric way, e.g., `margins = "norm"`. Following the latter approach, the log-likelihood of the marginal `Distributions` is optimized with respect to the first (and second) parameter(s) of the density `dxxx`. Basing on these estimates, the values of the univariate margins are computed. If the argument is defined as scalar, all margins are computed according to this specification. Otherwise, different margins can be defined, e.g., `margins = c("norm", "t", "edf")` for a three-dimensional sample. Except the Uniform distribution, all continuous `Distributions` of the **stats** package are available: `"beta"`, `"cauchy"`, `"chisq"`, `"exp"`, `"f"`, `"gamma"`, `"lnorm"`, `"norm"`, `"t"` and `"weibull"`. The values of non-parametrically estimated distributions are computed accordingly to

$$\widehat{F}(x) = (n+1)^{-1}\sum_{i=1}^{n}\mathbf{I}(X_i \leq x)\,. \tag{4}$$

Inappropriate usage of this argument might lead to misspecified margins, e.g., `margins = "exp"` although the sample contains negative values. Even though the margins might be assumed to follow parametric distributions if `margins != NULL`, no joint log-likelihood is maximized, but the margins are estimated in advance. As the asymptotic theory works well for parametric and nonparametric estimation of margins, for the univariate analysis we refer to other built-in packages. In practice, the column names of `X` should be specified, as the default names `X1, X2, ...` are given otherwise.

A further optional argument of `estimate.copula` determines the estimation `method`. We present three procedures: based on quasi `ML`, on Kendall's `TAU` and full ML `FML` respectively. Generally, the implemented HAC `type`s are not able to describe negative dependence, for which reason any identified negative dependence is set to the predefined minimal correlation `theta.eps` equal to 0.001 by default, if `method = TAU`. If a simple Archimedean copula is fitted to the data, the routines of the **copula** package are imported, see Yan (2007); Kojadinovic and Yan (2010). The supplementary function `theta2tau` computes Kendall's rank correlation coefficient basing on the value(s) of the dependency parameter(s), whereas `tau2theta` corresponds to the inverse function, see table 2.

At the final computational step of the procedure the binary HAC is checked for aggregation possibilities, if `epsilon > 0`. Then, the new dependency parameter is computed according to the specification `agg.method`, i.e., the `"min"`, `"max"` or `"mean"` of the original parameters. To emphasize this point, recall the four-dimensional binary HAC

$$C(u_1,\ldots,u_4;(((12)3)4),\boldsymbol{\theta}) = \phi_{\theta_3}\left\{\phi_{\theta_3}^{-1}\circ C\{u_1,\ldots,u_3;((12)3),(\theta_1,\theta_2)^\top\} + \phi_{\theta_3}^{-1}(u_4)\right\},$$

from section 2. If we assume additionally $\theta_1 \approx \theta_2$, such that $\theta_1 - \theta_2 < \varepsilon$, the copula C can be approximated by

$$C^*(u_1,\ldots,u_4;((123)4),\boldsymbol{\theta}) = \phi_{\theta_3}\left\{\phi_{\theta_3}^{-1}\circ C\{u_1,\ldots,u_3;(123),\theta^*\} + \phi_{\theta_3}^{-1}(u_4)\right\},$$

where $\theta^* = (\theta_1+\theta_2)/2$. This is referred to as the associativity property of Archimedean copulae, see Theorem 4.1.5 of Nelsen (2006). If the variables of two nodes are aggregated, the new copula is checked for aggregation possibilities as well. Beside the threshold approach, the realized estimates $\hat{\theta}_1$ and $\hat{\theta}_2$ can obviously be used to test $H_0 : \theta_1 - \theta_2 = 0$, since the asymptotic distribution is known. On the other hand, this approach is extremely computationally expensive. The estimation results for the non-aggregated and the aggregated cases are presented in the following:

Family	$\phi(u;\theta)$	Parameter range	$\tau(\theta)$
Gumbel	$\exp\left(-u^{1/\theta}\right)$	$1 \leq \theta < \infty$	$1 - 1/\theta$
Clayton	$(u+1)^{-1/\theta}$	$0 < \theta < \infty$	$\theta/(\theta+2)$

Table 2: Generator functions and the relations between the copula parameter and Kendall's τ.

```
> result.agg = estimate.copula(sample, margins = "edf", epsilon = 0.3)
> plot(result, circles = 0.3, index = TRUE, l = 1.7)
> plot(result.agg, circles = 0.3, index = TRUE, l = 1.7)
```

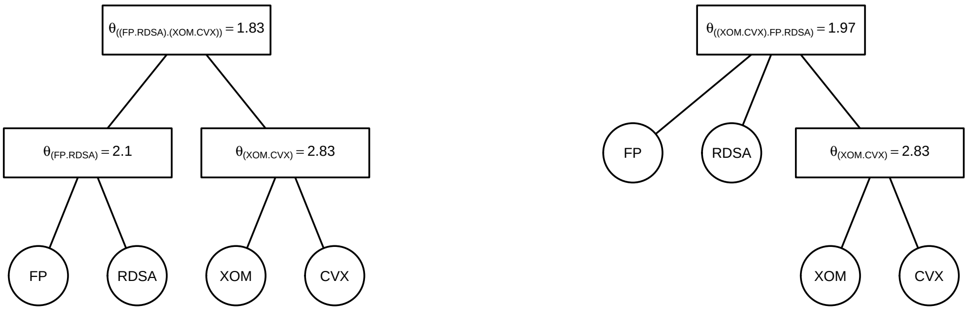

Figure 2: Plot of `result` on the left and `result.agg` on the right hand side.

3.1. The `hac` object

`hac` objects can be constructed by the general function `hac`, with the same name as the object it creates, and its simplified version `hac.full` for building fully nested HAC. For instance, consider the construction of a four-dimensional fully nested HAC with Gumbel generator, i.e.,

```
> G.cop = hac.full(type  = HAC_GUMBEL,
+                  y     = c("X4", "X3", "X2", "X1"),
+                  theta = c(1.1, 1.8, 2.5))
> G.cop

Class: hac
Generator: Gumbel
(((X1.X2)_{2.5}.X3)_{1.8}.X4)_{1.1}
```

where `y` denotes the vector of variables of class `character` and `theta` denotes the vector of dependency parameters. The parameters should be ascending ordered, so that the first parameter, `1.1`, is referred to the initial node of the HAC and the last parameter, `2.5`, corresponds to the first hierarchical level with variables `"X1"` and `"X2"`. Guarantee that the vector `y` contains one element more than the vector `theta`.

The returned output of `hac` objects is structured by three lines: (i) the object's `Class`, (ii) the `Generator` function and (iii) the HAC structure s. The structure can also be produced by the supplementary function `tree2str`. Variables, grouped at the same node are separated by a dot "`.`" and the dependency parameters are printed within the curly parentheses.

Partially nested Archimedean copulae are constructed by `hac` with the main argument `tree`. For a better understanding let us first consider a four-dimensional simple Archimedean copula with dependency parameter $\theta = 2$:

```
> hac(tree = list("X1", "X2", "X3", "X4", 2))

Class: hac
Generator: Gumbel
(X1.X2.X3.X4)_{2}
```

Obviously, the copula `tree` is constructed by a `list` consisting of four `character` objects, i.e., `"X1", "X2", "X3", "X4"`, and a number, which denotes the dependency parameter of the Archimedean copula. According to the theoretical construction of HAC in section 2, we can induce structure by substituting margins through a subcopula. The four variables `"X1", "X2", "X3", "X4"` can for example be structured by

```
> hac(tree = list(list("X1", "X2", 2.5), "X3", "X4", 1.5))

Class: hac
Generator: Gumbel
((X1.X2)_{2.5}.X3.X4)_{1.5}
```

where the nested component, `list("X1", "X2", 2.5)`, is referred to the subcopula of the lower hierarchical level. Note, that the nested component is of the same general form `list(..., numeric(1))` as the simple Archimedean copula, where `numeric(1)` denotes the dependency parameter and "`...`" refers to arbitrary variables and subcopulae, which may contain subcopulae as well, like presented in the following

```
> HAC = hac(tree = list(list("Y1", list("Z3", "Z4", 3), "Y2", 2.5),
+                       list("Z1", "Z2", 2), list("X1", "X2", 2.4),
+                       "X3", "X4", 1.5))
> HAC

Class: hac
Generator: Gumbel
((Y1.(Z3.Z4)_{3}.Y2)_{2.5}.(Z1.Z2)_{2}.(X1.X2)_{2.4}.X3.X4)_{1.5}
```

We cannot avoid the notation becoming more cumbersome for higher dimension, but the principle stays the same for arbitrary dimensions, i.e., variables are substituted by `lists` of the general form `list(..., numeric(1))`. The function `hac` provides a further argument for specifying the `type` of the HAC.

3.2. Graphics

As the string representation of the structure becomes more unclear as dimension increases, the package allows to produce graphics of `hac` objects by the standard generic `plot` function. Figure 3 illustrates for example the dependence structure of the lastly defined object `HAC`.

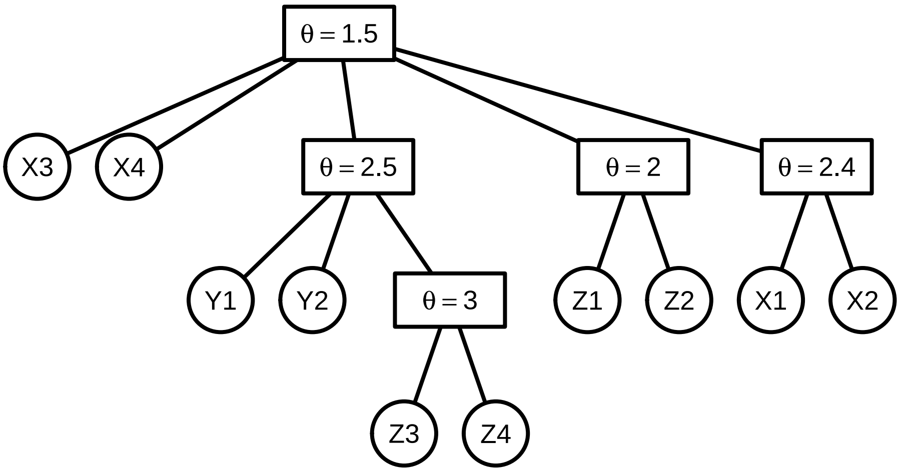

Figure 3: Plot of the final object `HAC`.

```
> plot(HAC, cex = 0.8, circles = 0.35)
```

The explanatory power of these plots can be enhanced by several of the usual `plot` parameters, e.g.,

```
> names(formals(plot.hac))
```

```
 [1] "x"         "xlim"      "ylim"      "xlab"      "ylab"      "col"
 [7] "fg"        "bg"        "col.t"     "lwd"       "index"     "numbering"
[13] "theta"     "h"         "l"         "circles"   "digits"    "..."
```

where the optional, boolean argument `theta` determines, whether the dependency parameter of the copula θ or Kendall's τ is printed, whereby Kendall's τ cannot be easily interpreted in the usual way for more than two dimensions. If `index = TRUE`, strings illustrating the subcopulae of the nodes, are used as subsrcipts of the dependency parameters. If additionally `numbering = TRUE`, the parameters are numbered, such that the subscripts correspond to the estimation stages, if the non-aggregated output of `estimate.copula` is plotted. The radius of the `circles`, the width `l` and the height `h` of the rectangles and the specific colors of the lines and the text can be adjusted. Further arguments "`...`" can for example be used to modify the font size `cex` or to include a subtitle `sub`.

3.3. Random sampling

To be in line with other R-packages providing tools for different univariate and multivariate distributions we provide: (i) `dHAC` for computing the values of the copula density, (ii) `pHAC` for the cumulative distribution function and (iii) `rAC` and `rHAC` for simulations. `rAC` is based on the algorithm of Marshall and Olkin (1988) for sampling from simple Archimedean copulae and `rHAC` simulates from arbitrary HAC as suggested in Hofert and Mächler (2011), who summarize the procedure for the former **nacopula** package as follows:

Algorithm 1 (Hofert and Mächler (2011)). *Let C be a nested Archimedean copula with root copula C_0 generated by ϕ_0. Let U be a vector of the same dimension as C_0.*

```
1. sample from inverse Laplace transform $\mathcal{LS}^{-1}$ of $\phi_0$, i.e., $V_0 \sim F_0 \overset{\text{def}}{=} \mathcal{LS}^{-1}(\phi_0)$
2. for all components $u$ of $C_0$ that are nested Archimedean copulae do:
    (a) set $C_1$ with generator $\phi_1$ to the nested Archimedean copula $u$
    (b) sample $V_{01} \sim F_{01} \overset{\text{def}}{=} \mathcal{LS}^{-1}\{\phi_{01}(\cdot; V_0)\}$
    (c) set $C_0 \overset{\text{def}}{=} C_1, \phi_0 \overset{\text{def}}{=} \phi_1$, and $V_0 \overset{\text{def}}{=} V_{01}$ and continue with 2.
3. for all other components $u$ of $C_0$ do
    (a) sample $R \sim Exp(1)$
    (b) set the component of $U$ corresponding to $u$ to $\phi_0(R/V_0)$
4. return $U$
```

The function requires only two arguments: (i) the sample size `n` and (ii) an object of the class `hac` specifying the characteristics of the underlying HAC, e.g.,

```
> sim.data = rHAC(500, G.cop)
> pairs(sim.data, pch = 20)
```

In particular the contributions of McNeil (2008), Hofert (2008) and Hofert (2011) provide the theoretical foundations to sample computationally efficient random vectors from HACs. Since the functions of the **HAC** package are not directly compatible with R-objects for nested Archimedean copula of the **copula** package and vice versa, we implemented algorithm 1 to avoid transformations of elaborate structures from one object to another. The algorithm exploits the recursively determined structure of HACs and samples from the major random components F_0 and F_{01}, which are presented in table 3, where S denotes the stable distribution with $S1$ parametrization, Γ denotes the Gamma distribution and $\widetilde{S}$ refers to the exponentially titled stable distribution. Consider Nolan (1997); Samorodnitsky and Taqqu (1994) for the first, Ahrens and Dieter (1974, 1982) for the second and Hofert (2011); Hofert and Mächler (2011) for the third as a reference.

3.4. The cdf and density

The arguments for `pHAC` are a `hac` object and a sample `X`, whose column names should be identical to the variables' names of the `hac` object, e.g.,

```
> probs = pHAC(X = sim.data, hac = G.cop)
```

As the copula density is defined as d-th derivative of the copula C with respect to the arguments u_j, $j = 1, \dots, d$, c.f. Savu and Trede (2010), the explicit form of the density varies with the structure of the underlying HAC. Hence, including the explicit form of all possible

Family	F_0	F_{01}, $\alpha = \theta_0/\theta_1$
Gumbel	$S(1/\theta, 1, \cos^{\theta}\{\pi/(2\theta)\}, \mathbf{I}\{\theta = 1\}; 1)$	$S(\alpha, 1, \{\cos(\alpha\pi/2)V_0\}^{1/\alpha}, V_0\mathbf{I}\{\alpha = 1\}; 1)$
Clayton	$\Gamma(1/\theta, 1)$	$\widetilde{S}(\alpha, 1, \{\cos(\alpha\pi/2)V_0\}^{1/\alpha}, V_0\mathbf{I}\{\alpha = 1\}, \mathbf{I}\{\alpha \neq 1\}; 1)$

Table 3: Functions of algorithm 1. The parameters of $S\,(\alpha,\beta,\gamma,\delta;1)$ and $\widetilde{S}\,(\alpha,\beta,\gamma,\delta;1)$ denote the index parameter $\alpha \in (0,2)$, skewness parameter $\beta \in [-1,1]$, scale parameter $\gamma \in [0,\infty)$ and shift parameter $\delta \in (-\infty,\infty)$. The first parameter of $\Gamma(\cdot,\cdot)$ refers to the shape and the second parameter to the intensity parameter.

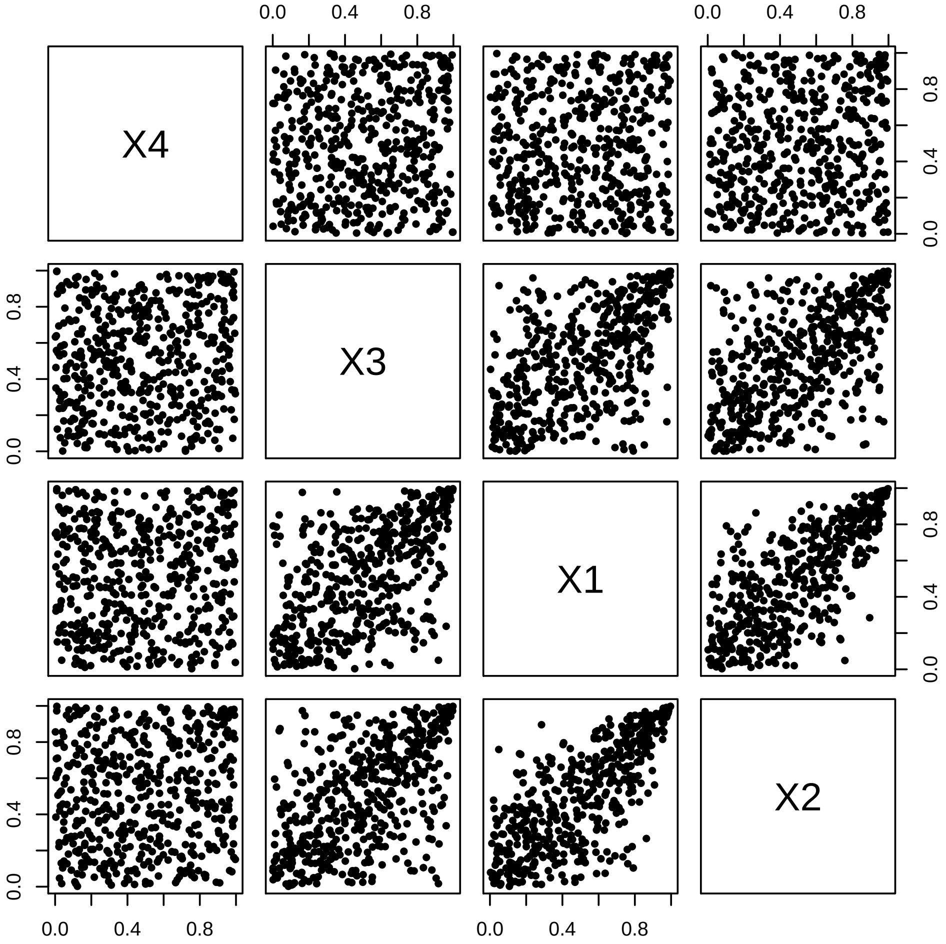

Figure 4: Scatterplot of the sample `sim.data`.

d-dimensional copula densities is absolutely unrealistic in practice. Our function `dHAC` derives an analytical expression of the density for a given `hac` object, which can be instantaneously evaluated, if `eval = TRUE`. The analytical expression of the density is found by subsequently using the `D` function to differentiate the algebraic form of the copula "symbolically" with respect to the variables of the inserted `hac` object. Although the derivation and evaluation of the density is either computationally or numerically demanding, `dHAC` provides a flexible way to work with the HAC densities in practice, because they have not to be derived manually or numerically. Since the densities of the two-dimensional Archimedean copulae are frequently called during the multi-stage estimation procedure, their closed form expressions are given explicitly.

3.5. Empirical copula

As long as our package does not cover goodness-of-fit tests, which are difficult to implement in general and involve intensive computational techniques via bootstrapping, see Genest, Rémillard, and Beaudoin (2009), it might be difficult to justify the choice of a parametric assumption. However, the values of `probs` can be compared to the values the corresponding empirical copula, i.e.,

$$\widehat{C}\left(u_1, \ldots, u_d\right) = n^{-1} \sum_{i=1}^{n} \prod_{j=1}^{d} \mathbf{I}\left\{\widehat{F}_j\left(X_{ij}\right) \leq u_j\right\},$$

where $\widehat{F}_j$ denotes the estimated marginal distribution function of variable X_j. Figure 5 suggests a proper fit of the empirical copula computed by

```
> probs.emp = emp.copula.self(sim.data, proc = "M")
```

There are two functions, which can be used for computing the empirical copula:

```
> emp.copula(u, x, proc = "M", sort = "none", margins = NULL,
+            na.rm = FALSE, ...)
> emp.copula.self(x, proc = "M", sort = "none", margins = NULL,
+                 na.rm = FALSE, ...)
```

The difference in the arguments of these functions is, that `emp.copula` requires a matrix `u`, at which the function is evaluated. In contrast, `emp.copula.self` evaluates the sample `x` itself. The argument `proc` enables the user to choose between two computational methods. We recommend to use the default method, `proc = "M"`, which is based on `matrix` manipulations, because its computational time is just a small fraction of the taken time of method `"A"`, see figure 6. However, method `"M"` is sensitive with respect to the size of the working memory and therefore inapplicable for very large datasets. Figure 6 illustrates rapidly increasing computational times of the `matrix` based method for more than 5000 observations until the method collapses. In contrast, the runtimes of the alternative method `proc = "A"` are more robust against increasing the sample size. An other option to deal with large datasets is specifying the matrix `u` manually in order to reduce the number of vectors to evaluate.

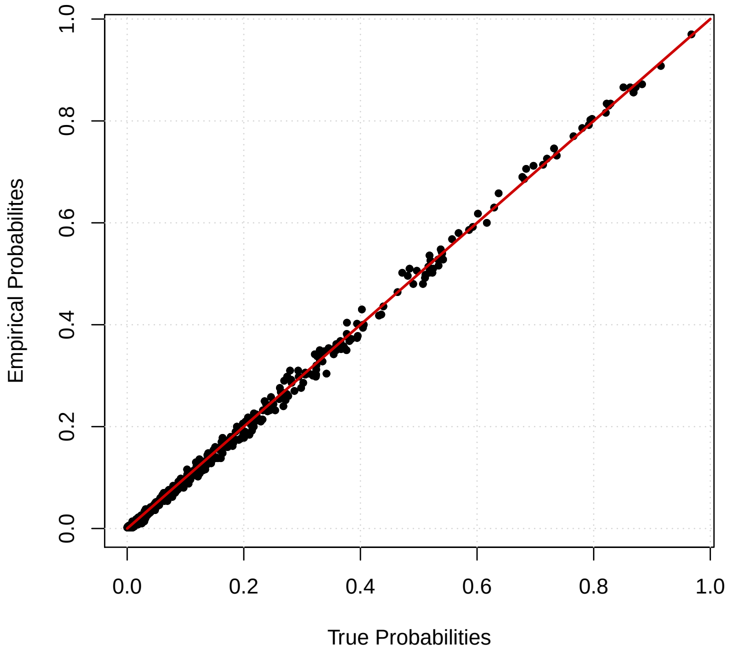

Figure 5: The values of `probs` on the x-axis against the values of `probs.emp`.

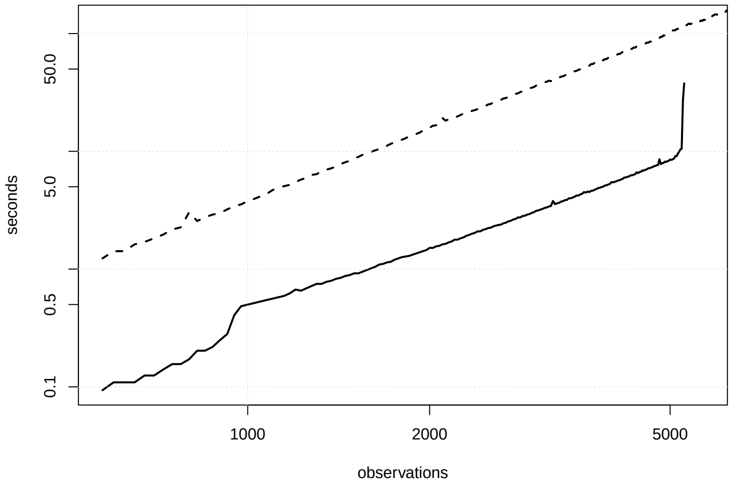

Figure 6: The figure shows the computational times for an increasing sample-size but a fixed dimension $d = 5$ on a log-log scale. The solid line is referred to `proc = "M"` and the dashed line to `proc = "A"`.

4. Simulation study

To ensure the accuracy of the proposed methods, we generate random data from six copula models of different dimension $C_i^j = C\left(\cdot; s_i, \boldsymbol{\theta}_i\right)^j$, for $i = 1, 2, 3$ and $j = C, G$, and show, that the parameter estimates almost coincide with the theoretical models. Here, j denotes the copula family (Clayton or Gumbel) and the structures are given by $s_1 = ((12)3)$, $s_2 = ((((12)3)4)5)$ and $s_3 = ((12)(34)5)$. The values of $\boldsymbol{\theta}_i$ are presented in the respective tables 4, 5 and chosen, such that the similar strength of dependence is produced by the Clayton and Gumbel based models.

The summary statistics of tables 4 and 5 rely on $n = 1000$ estimates, whereby only estimates with the same structure are to be compared. For this reason the procedure was m times replicated till $n = 1000$ estimates were available. As `estimate.copula` approximates the true structure, we set `epsilon = 0.15` for C_3, which is not based on a binary structure. The simulated samples for the copula estimation consist of 500 observations for the three-dimensional case and 1000 for the five-dimensional cases. Tables 4 and 5 indicate, that the estimation procedure works properly for the suggested models, as the estimates are on average consistent with the true parameters. Nevertheless, a few points remain to mention: (i) the statistic $\bar{s}$ denotes the percentage of correctly classified structures, i.e., $\bar{s} = n/m$, taking the permutation symmetry of the variables at the same node into consideration. The procedure detects the true structure as long as it is binary and the parameters exhibit the imposed distance. (ii) The estimates at lower hierarchical levels show a higher volatility than the estimates close to the initial node and the estimates for the Clayton models are more volatile than the estimates of the Gumbel based HACs. (iii) All estimated models indicate more imprecise estimates for higher nesting levels implying the high number of trials m to yield $n = 1000$ usable sample realizations of $\boldsymbol{\theta}$ for computing the summary statistics in the case of model C_3, see table 5, which contains the frequencies for adopted misspecified structures and the averaged full ML estimates as well.

To shed some light on this issue, let us recover the estimation procedure step by step. After estimating θ_{12} and θ_{34} at the first and second interation step respectively, the realizations of the variables

$$Z_{12} \overset{\text{def}}{=} \phi_{\hat{\theta}_{12}}\left[\phi^{-1}_{\hat{\theta}_{12}}\left\{F_1\left(X_1\right)\right\} + \phi^{-1}_{\hat{\theta}_{12}}\left\{F_2\left(X_2\right)\right\}\right]$$
$$Z_{34} \overset{\text{def}}{=} \phi_{\hat{\theta}_{34}}\left[\phi^{-1}_{\hat{\theta}_{34}}\left\{F_3\left(X_3\right)\right\} + \phi^{-1}_{\hat{\theta}_{34}}\left\{F_4\left(X_4\right)\right\}\right]$$

are computed. At the next step, the pseudo variables are grouped and the corresponding parameter is apparently overestimated. If additionally the parameter at the fourth estimation stage is underestimated, like the parameter of the initial node for the C_2^C model, the variables of the first and second node of the binary structure cannot be aggregated, i.e., $\hat{\theta}_{((12)(34))} - \hat{\theta}_{(((12)(34))5)} \not< 0.15$. Hence, the procedure leads to the binary approximation $s = (((12)(34))5)$ for 62.71% of the Gumbel and 93.95% of the Clayton based HACs respectively. The distributions based on this binary structure are however very good approximations of the true distributions and provide more flexibility. The gains from the full ML approach regarding the precision are only observable at the root node of the copula. However, this minor improvement is costly since the results are based on a preestimated structure. Note, any `Warning messages` related to the full ML estimation stating that `NaNs` were produced are reasoned by the numerically challenging evaluation of the density as mentioned above.

			Statistics				
Model	$\boldsymbol{\theta}$	$\bar{s}$	min	median	mean	max	sd
C_1^G	$\theta_2 = 1.500$	100%	1.32	1.50	1.50	1.71	0.05
	$\theta_1 = 3.000$		2.68	3.00	3.00	3.35	0.11
C_1^C	$\theta_2 = 1.000$	100%	0.77	0.99	0.99	1.44	0.09
	$\theta_1 = 4.000$		3.42	4.01	4.01	4.62	0.20
C_2^G	$\theta_4 = 1.125$	100%	1.04	1.10	1.10	1.17	0.02
	$\theta_3 = 1.500$		1.36	1.45	1.45	1.58	0.03
	$\theta_2 = 2.250$		2.06	2.24	2.24	2.44	0.06
	$\theta_1 = 4.500$		4.08	4.50	4.50	4.94	0.12
C_2^C	$\theta_4 = 0.250$	100%	0.06	0.17	0.17	0.28	0.03
	$\theta_3 = 1.000$		0.73	0.92	0.92	1.09	0.06
	$\theta_2 = 2.500$		2.20	2.49	2.49	2.83	0.10
	$\theta_1 = 7.000$		6.26	7.00	7.00	7.75	0.22

Table 4: The models for the Gumbel family C_1^G, C_2^G and for the Clayton family C_1^C, C_2^C, where $\boldsymbol{\theta}$ denotes the true copula parameters and $\bar{s} = n/m$ the percentage of correctly classified structures with $n = 1000$.

Irrespective of this inaccuracy, the distributions of samples from the **nacopula** package can also be reconstructed by the **HAC** package with identical results.

5. Conclusion

The package **HAC** focuses on the computationally efficient estimation of hierarchical Archimedean copula, which is based on grouping binary structures within a recursive multi-stage ML procedure. Its theoretical and practical advantages are (i) the avoiding of the demanding asymptotic theory, which arises due to one-step ML estimation and (ii) the consecutive optimization of the two-dimensional log-likelihood instead of the singular optimization of the d-dimensional one. Since HACs permit to model high-dimensional random variables, the package allows to `plot` the related `hac` objects. According to the usual naming of distributions in R, we provide `dHAC`, `pHAC` and `rHAC` to compute the values of density- and distribution functions or to sample from arbitrary HACs. The constructed framework can be easily extended to generator families for which the required nesting condition is fulfilled, e.g., Frank and Joe. Finally, the accuracy of the methods is shown in a small simulation study.

		Statistics for multi-stage ML					
Model	$\boldsymbol{\theta}$	((12)(34)5)	min	median	mean	max	sd
C_3^G	$\theta_3 = 1.125$	35.41%	1.10	1.17	1.17	1.23	0.02
	$\theta_2 = 1.500$		1.37	1.51	1.51	1.61	0.04
	$\theta_1 = 3.000$		2.80	3.01	3.01	3.30	0.08
C_3^C	$\theta_3 = 0.250$	06.05%	0.15	0.29	0.28	0.41	0.06
	$\theta_2 = 1.000$		0.79	1.00	1.00	1.22	0.06
	$\theta_1 = 4.000$		3.65	4.01	4.01	4.50	0.14
		Misspecified structures					
		(((12)(34))5)		(((12)34)5)			
C_3^G	-	62.71%		01.88%			
C_3^C	-	93.95%		00.00%			
		Statistics for full ML					
		-	min	median	mean	max	sd
C_3^G	$\theta_3 = 1.125$	-	1.08	1.12	1.12	1.17	0.01
	$\theta_2 = 1.500$		1.36	1.51	1.51	1.62	0.04
	$\theta_1 = 3.000$		2.81	3.01	3.01	3.30	0.08
C_3^C	$\theta_3 = 0.250$	-	0.19	0.24	0.24	0.32	0.03
	$\theta_2 = 1.000$		0.79	1.00	1.00	1.22	0.06
	$\theta_1 = 4.000$		3.64	4.01	4.02	4.51	0.14

Table 5: The model for the Gumbel family C_3^G and for the Clayton family C_3^C, where $\boldsymbol{\theta}$ denotes the true copula parameters and the column ((12)(34)5) refers to the percentage of correctly classified structures, i.e. n/m with $n = 1000$. The misspecified structures and the results of the full ML estimation are also presented.

References

Acar EF, Craiu RV, Yao F (2011). "Dependence Calibration in Conditional Copulas: A Nonparametric Approach." *Biometrics*, **67**(2), 445–453.

Ahrens JH, Dieter U (1974). "Computer Methods for Sampling from Gamma, Beta, Poisson and Bionomial Distributions." *Computing*, **12**, 223–246.

Ahrens JH, Dieter U (1982). "Generating Gamma Variates by a Modified Rejection Technique." *Communications of the ACM*, **25**, 47–54.

Bárdossy A (2006). "Copula-Based Geostatistical Models for Groundwater Quality Parameters." *Water Resources Research*, **42**(11), 1–12.

Bárdossy A, Li J (2008). "Geostatistical Interpolation Using Copulas." *Water Resources Research*, **44**(7), 1–15.

Chen X, Fan Y (2006). "Estimation and Model Selection of Semiparametric Copula-Based Multivariate Dynamic Models under Copula Misspecification." *Journal of Econometrics*, **135**, 125–154.

Cherubini U, Luciano E, Vecchiato W (2004). *Copula Methods in Finance*. John Wiley & Sons, New York.

Embrechts P, Lindskog F, McNeil AJ (2003). "Modeling Dependence with Copulas and Applications to Risk Management." In ST Rachev (ed.), *Handbook of Heavy Tailed Distributions in Finance*. Elsevier, North-Holland.

Embrechts P, McNeil AJ, Straumann D (1999). "Correlation And Dependence in Risk Management: Properties and Pitfalls." In *Risk Management: Value at Risk and Beyond*, pp. 176–223. Cambridge University Press.

Franke J, Härdle WK, Hafner C (2011). *Statistics of Financial Markets: An Introduction*. Universitext, 3 edition. Springer-Verlag.

Genest C, Favre AC (2007). "Everything You Always Wanted to Know about Copula Modeling but Were Afraid to Ask." *Journal of Hydrologic Engineering*, **12**(4), 347–368.

Genest C, Rémillard B, Beaudoin D (2009). "Goodness-of-Fit Tests for Copulas: A Review and a Power Study." *Insurance: Mathematics and Economics*, **44**(2), 199–213.

Genest C, Rivest LP (1993). "Statistical Inference Procedures for Bivariate Archimedean Copulas." *Journal of the American Statistical Association*, **88**, 1034–1043.

Hofert M (2008). "Sampling Archimedean Copulas." *Computational Statistics & Data Analysis*, **52**, 5163–5174.

Hofert M (2011). "Efficiently Sampling Nested Archimedean Copulas." *Computational Statistics & Data Analysis*, **55**, 57–70.

Hofert M, Mächler M (2011). "Nested Archimedean Copulas Meet R: The **nacopula** Package." *Journal of Statistical Software*, **39**(9), 1–20. URL http://www.jstatsoft.org/v39/i09/.

Jaworski P, Durante F, Härdle WK, Rychlik T (2010). *Copula Theory and Its Applications*. Lecture notes in statistics, 1 edition. Springer-Verlag.

Joe H (1996). "Families of m-Variate Distributions with Given Margins and $m(m-1)/2$ Bivariate Dependence Parameters." In L Rüschendorf, B Schweizer, M Taylor (eds.), *Distribution with fixed marginals and related topics*, IMS Lecture Notes – Monograph Series. Institute of Mathematical Statistics.

Joe H (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall, London.

Junker M, May A (2005). "Measurement of Aggregate Risk with Copulas." *Econometrics Journal*, **8**(3), 428–454.

Kojadinovic I, Yan J (2010). "Modeling Multivariate Distributions with Continuous Margins Using the **copula** R Package." *Journal of Statistical Software*, **34**(9), 1–20. URL http://www.jstatsoft.org/v34/i09/.

Lakhal-Chaieb ML (2010). "Copula Inference under Censoring." *Biometrika*, **97**(2), 505–512.

Li DX (2000). "On Default Correlation: A Copula Function Approach." *Journal of Fixed Income*, **9**(4), 43–54.

Marshall AW, Olkin J (1988). "Families of Multivariate Distributions." *Journal of the American Statistical Association*, **83**, 834–841.

McNeil AJ (2008). "Sampling Nested Archimedean Copulas." *Journal of Statistical Computation and Simulation*, **78**, 567–581.

McNeil AJ, Nešlehová J (2009). "Multivariate Archimedean Copulas, d-Monotone Functions and l_1 Norm Symmetric Distributions." *The Annals of Statistics*, **37**(5b), 3059–3097.

McNeil AJ, Ulman S (2011). *QRMlib: Provides R-language code to examine Quantitative Risk Management concepts*. R package version 1.4.5.1, URL http://cran.r-project.org/src/contrib/Archive/QRMlib/.

Mendes BVM, Souza RM (2004). "Measuring Financial Risks with Copulas." *International Review Of Financial Analysis*, **13**, 53–77.

Nelsen RB (2006). *An Introduction to Copulas*. Springer Verlag, New York.

Nolan JP (1997). "Numerical Calculation of Stable Densities and Distribution Functions." *Communications in Statistics-Stochastic Models*, **13**(4), 759–774.

Okhrin O, Okhrin Y, Schmid W (2011a). "On the Structure and Estimation of Hierarchical Archimedean Copulas." *under second revision in Journal of Econometrics*.

Okhrin O, Okhrin Y, Schmid W (2011b). "Properties of Hierarchical Archimedean Copulae." *under third revision in Statistics and Risk Modeling (former Statistics and Decisions)*.

Samorodnitsky G, Taqqu M (1994). *Stable non-Gaussian Random Processes: Stochastic Models with Infinite Variance*. Stochastic modeling. Chapman & Hall.

Savu C, Trede M (2010). "Hierarchies of Archimedean Copulas." *Quantitative Finance*, **10**(3), 295–304.

Sklar A (1959). "Fonctions de Répartition à n Dimension et Leurs Marges." *Publications de l'Institut de Statistique de l'Université de Paris*, **8**, 299–231.

Whelan N (2004). "Sampling from Archimedean Copulas." *Quantitative Finance*, **4**, 339–352.

Wuertz D, *et al.* (2009). *fCopulae: Rmetrics - Dependence Structures with Copulas*. R package version 2110.78, URL http://CRAN.R-project.org/package=fCopulae.

Yan J (2007). "Enjoy the Joy of Copulas: With a Package **copula**." *Journal of Statistical Software*, **21**(4), 1–21. URL http://www.jstatsoft.org/v21/i04/.

Affiliation:

Ostap Okhrin
C.A.S.E - Center for Applied Statistics and Economics
Ladislaus von Bortkiewicz Chair of Statistics
Humboldt-Universität zu Berlin
10099 Berlin, Germany
E-mail: ostap.okhrin@wiwi.hu-berlin.de
URL: http://lehre.wiwi.hu-berlin.de/Professuren/quantitativ/statistik/members/personalpages/o2

Alexander Ristig
Ladislaus von Bortkiewicz Chair of Statistics
Humboldt-Universität zu Berlin
10099 Berlin, Germany
E-mail: ristigal@hu-berlin.de
URL: http://lehre.wiwi.hu-berlin.de/Professuren/quantitativ/statistik/members/personalpages/ar

SFB 649 Discussion Paper Series 2012

For a complete list of Discussion Papers published by the SFB 649, please visit http://sfb649.wiwi.hu-berlin.de.

001 "HMM in dynamic HAC models" by Wolfgang Karl Härdle, Ostap Okhrin and Weining Wang, January 2012.

002 "Dynamic Activity Analysis Model Based Win-Win Development Forecasting Under the Environmental Regulation in China" by Shiyi Chen and Wolfgang Karl Härdle, January 2012.

003 "A Donsker Theorem for Lévy Measures" by Richard Nickl and Markus Reiß, January 2012.

004 "Computational Statistics (Journal)" by Wolfgang Karl Härdle, Yuichi Mori and Jürgen Symanzik, January 2012.

005 "Implementing quotas in university admissions: An experimental analysis" by Sebastian Braun, Nadja Dwenger, Dorothea Kübler and Alexander Westkamp, January 2012.

006 "Quantile Regression in Risk Calibration" by Shih-Kang Chao, Wolfgang Karl Härdle and Weining Wang, January 2012.

007 "Total Work and Gender: Facts and Possible Explanations" by Michael Burda, Daniel S. Hamermesh and Philippe Weil, February 2012.

008 "Does Basel II Pillar 3 Risk Exposure Data help to Identify Risky Banks?" by Ralf Sabiwalsky, February 2012.

009 "Comparability Effects of Mandatory IFRS Adoption" by Stefano Cascino and Joachim Gassen, February 2012.

010 "Fair Value Reclassifications of Financial Assets during the Financial Crisis" by Jannis Bischof, Ulf Brüggemann and Holger Daske, February 2012.

011 "Intended and unintended consequences of mandatory IFRS adoption: A review of extant evidence and suggestions for future research" by Ulf Brüggemann, Jörg-Markus Hitz and Thorsten Sellhorn, February 2012.

012 "Confidence sets in nonparametric calibration of exponential Lévy models" by Jakob Söhl, February 2012.

013 "The Polarization of Employment in German Local Labor Markets" by Charlotte Senftleben and Hanna Wielandt, February 2012.

014 "On the Dark Side of the Market: Identifying and Analyzing Hidden Order Placements" by Nikolaus Hautsch and Ruihong Huang, February 2012.

015 "Existence and Uniqueness of Perturbation Solutions to DSGE Models" by Hong Lan and Alexander Meyer-Gohde, February 2012.

016 "Nonparametric adaptive estimation of linear functionals for low frequency observed Lévy processes" by Johanna Kappus, February 2012.

017 "Option calibration of exponential Lévy models: Implementation and empirical results" by Jakob Söhl und Mathias Trabs, February 2012.

018 "Managerial Overconfidence and Corporate Risk Management" by Tim R. Adam, Chitru S. Fernando and Evgenia Golubeva, February 2012.

019 "Why Do Firms Engage in Selective Hedging?" by Tim R. Adam, Chitru S. Fernando and Jesus M. Salas, February 2012.

020 "A Slab in the Face: Building Quality and Neighborhood Effects" by Rainer Schulz and Martin Wersing, February 2012.

021 "A Strategy Perspective on the Performance Relevance of the CFO" by Andreas Venus and Andreas Engelen, February 2012.

022 "Assessing the Anchoring of Inflation Expectations" by Till Strohsal and Lars Winkelmann, February 2012.

SFB 649, Spandauer Straße 1, D-10178 Berlin
http://sfb649.wiwi.hu-berlin.de

This research was supported by the Deutsche Forschungsgemeinschaft through the SFB 649 "Economic Risk".

SFB 649 Discussion Paper Series 2012

For a complete list of Discussion Papers published by the SFB 649, please visit http://sfb649.wiwi.hu-berlin.de.

023 "Hidden Liquidity: Determinants and Impact" by Gökhan Cebiroglu and Ulrich Horst, March 2012.

024 "Bye Bye, G.I. - The Impact of the U.S. Military Drawdown on Local German Labor Markets" by Jan Peter aus dem Moore and Alexandra Spitz-Oener, March 2012.

025 "Is socially responsible investing just screening? Evidence from mutual funds" by Markus Hirschberger, Ralph E. Steuer, Sebastian Utz and Maximilian Wimmer, March 2012.

026 "Explaining regional unemployment differences in Germany: a spatial panel data analysis" by Franziska Lottmann, March 2012.

027 "Forecast based Pricing of Weather Derivatives" by Wolfgang Karl Härdle, Brenda López-Cabrera and Matthias Ritter, March 2012.

028 “Does umbrella branding really work? Investigating cross-category brand loyalty” by Nadja Silberhorn and Lutz Hildebrandt, April 2012.

029 “Statistical Modelling of Temperature Risk” by Zografia Anastasiadou, and Brenda López-Cabrera, April 2012.

030 “Support Vector Machines with Evolutionary Feature Selection for Default Prediction” by Wolfgang Karl Härdle, Dedy Dwi Prastyo and Christian Hafner, April 2012.

031 “Local Adaptive Multiplicative Error Models for High-Frequency Forecasts” by Wolfgang Karl Härdle, Nikolaus Hautsch and Andrija Mihoci, April 2012.

032 “Copula Dynamics in CDOs.” by Barbara Choroś-Tomczyk, Wolfgang Karl Härdle and Ludger Overbeck, Mai 2012.

033 “Simultaneous Statistical Inference in Dynamic Factor Models” by Thorsten Dickhaus, Mai 2012.

034 “Realized Copula” by Matthias R. Fengler and Ostap Okhrin, Mai 2012.

035 “Correlated Trades and Herd Behavior in the Stock Market” by Simon Jurkatis, Stephanie Kremer and Dieter Nautz, Mai 2012

036 “Hierarchical Archimedean Copulae: The HAC Package” by Ostap Okhrin and Alexander Ristig, Mai 2012.

SFB 649, Spandauer Straße 1, D-10178 Berlin
http://sfb649.wiwi.hu-berlin.de

This research was supported by the Deutsche Forschungsgemeinschaft through the SFB 649 "Economic Risk".