University of Cologne

Malte Bonart

Testing for Structural Breaks in Factor Copula Models - Implementation and Application in Social Media Topic Analysis

Master thesis

Supervisor: Prof.Dr. Dominik Wied

Submitted for the Master Examination in Economics at the Faculty of Management, Economics and Social Sciences of the University of Cologne in June 2018.

This is my abstract.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

cdf	Cumulative distribution function
iid	Identically and independent distributed
SMM	Simulated methods of moments
EDF	Empirical distribution function
DGP	Data generating process
btw17	Bundestag election 2017
API	Application programming interface
CV	Critical value

1. Introduction

Krupskii and Joe (2013) and Joe (2015) also have model named “factor copula” but they use a different appraoch.

Statistical models involving copula functions became increasingly popular since the 1990th (Nelsen 1999, p. 1). The concept was first introduced in the work by Sklar (1959). Copulas are mainly used in two ways: First, to model the dependence structure of multivariate distributions independent of their underlying marginal distributions and second, to construct bivariate or multivariate distributions (Sempi 2011, p. 302). This paper focuses on the first application. Up to the year 1999 the there are 38 publication listed for the topics *copula* and *dependence*. From 2000 - 2008 there were 334 publication and from 2009 - 2017 this number increased to 2048 publication.[1]

Skalr’s theorem can be used to construct multivariate models first, by specifying the marginal distributions of the random variables involved and second, by specifying the dependence structure among the variables via a copula function Sempi 2011. By doing so, one allows for non-parametrized or semi-parametrized estimation of the marginal distributions together with a parametrized copula. High dimensional problems become traceable since the number of parameters can be drastically reduced (Patton 2009, p. 777).

For time series data copula theory can be used in two ways: First, to describe the cross sectional dependence structure by estimating the conditional copula function of the conditional joint distribution $F(\mathbf{y}|\mathcal{F}_{t-1})$ with $\mathbf{Y}_t = [Y_{1t}, \dots, Y_{nt}]'$ and past information $\mathcal{F}_{t-1}$. To obtain a valid distribution, the information set must be the same for both the copula and the marginal distributions. Second, copulas can be used to describe the dependence between observations of a univariate time series $[Y_t, Y_{t+1}, \dots, Y_{t+n}]'$. This is related to the study of Markov processes. (Patton 2009, p. 771 ff). This paper focuses on the first application.

Applications for copula modeling can be found in various disciplines but they became increasingly popular in the field of finance, actuarial science and hydrology Sempi 2011.

Correlation or covariance matrices can be used to model linear dependence especially for multivariate normal or t-distributions. But they lack the abil-

[1] Numbers are taken from the *Web of Science Core Collection* for the search term `TOPIC: (copula) AND TOPIC: (dependence)`.

ity to model the dependence e.g. in the presence of heavy tails or outliers (**Kumar2011**).

Rank correlation matrices such as *Spearman's rho* are invariant under monotonic transformations but they are not moment-based.

Copulas are usefull whenever one wants to draw conclusion about the full joint distribution but other models are more suited if one focuses on predicting the conditional mean (Patton 2009, p. 769)

This thesis summarizes and strucutures the current developement in the field of factor copula models. It contributes to the sicentific disucssion in two ways: First, a software library for a consistent specification, simulation and estimation of factor copula models is presented. With the library, the methods and the strucutral break test are made avalaible to a broader scientific audience and can be used by applied reasearchers. Second, we study novel ways of applying the discussed methods to areas outside of the finance community. The strucutral break test is applied to an aggregated dataset of social media posts from german politicians and political parties.

The thesis is structured in four main chapters: The first chapter lays the theoretical foundation by summarizing important aspects of copula theory and by presenting the factor copula approach, its estimation strategy via simulated methods of moments and a suitable test for time varying dependence structures. The second chapter presents implementational details of the software library *factorcopula*, written in the statistical programming language R (Bonart 2018). To ensure the validity of the package and its performance two simulation studies are performed. In the last chapter, the methods are applied to a large dataset of textual social media posts from german politicians and political parties. The goal is to illustrate and assess novel ways of applying the factor copula approach and the structural break test.

2. Theoretical foundation

In this chapter we present and summarize the theoretical foundation of this thesis. First, the general idea behind copula functions is introduced. Second, a special class of copula models, the so called *factor copula* model is presented. Third, we discuss copula models in the context of time series data and present a specific framework for modelling dependencies of multivariate time series. Fourth, the *simulated methods of moments* is explained. It is the estimation method which is used throughout this work. Finally we summarize the ideas of a structural break test for the possibly time varying parameters of a factor copula model.

2.1. Copula theory

The joint cumulative distribution function $F_Y(y_1, \dots, y_N) = P(Y \leq y_1, \dots, Y \leq y_N)$ for some multivariate random variable Y of dimension N and its corresponding marginal distributions $F_{Y_i}(y_i) = P(Y_i \leq y_i) \; \forall \; i = 1, \dots, N$ can be used to describe a joint probability distribution. Estimating F_Y is computationally demanding espacialy if N becomes large. Therefore, a copula function is introduced which can be used to link the marginal and the joint distribution function.

A function of the type $C : [0,1]^N \to [0,1]$, with $N \geq 2$ is called a *copula* if first, it shares the common properties of a multivariate distribution function such that $C_U(u_1, \dots, u_N) = P(U_1 \leq u_1, \dots, U_N \leq u_N)$ and if second, its marginal distributions are $U_i \sim U(0,1)$, e.g. uniformly distributed (Joe 2015, p. 7). A thereom by Sklar (1959) states, that every d-variate distribution function $F_Y(y_1, \dots, y_n)$ can be eypressed in terms of its marginal distributions and a copula function such that

$$F(y_1, \dots, y_N) = C(F_1(y_1), \dots, F_N(y_N)). \tag{2.1}$$

If F_Y is *continuous* with marginal quantile functions $F_{Y_1}^{-1}, \dots, F_{Y_N}^{-1}$ then $C(\mathbf{u})$ is uniquely determined by $C(\mathbf{u}) = F(F_{Y_1}^{-1}(u_1), \dots, F_{Y_N}^{-1}(u_N))$.

Some bivariate measures of dependency share the property of *scale invariance*. Thus, the measures are invariant with respect to the marginal distributions and can therefore expressed as a function of their copula (**Schmid2010**). The widely used linear correlation coefficient is dependend on both the copula and the marginal distributions (Patton 2009, p. 768). Two scale invariant measures are Spearman's rank correlation

$$\rho_{Y_i,Y_j} = 12 \int\int_{[0,1]^2} u_1 u_2 dC(u_i, u_j) - 3, \tag{2.2}$$

and Kendal's rank correlation

$$\tau_{Y_i,Y_j} = 4 \int\int_{[0,1]^2} C(u_i,u_j) dC(u_i,u_j) - 1 = 4E(C(U_i,U_j)) - 1. \tag{2.3}$$

Lower and upper tail dependency is defined as

$$\begin{aligned} \tau^L_{Y_i,Y_j} &= \lim_{q \to 0} \frac{P(Y_i \leq F^{-1}_{Y_i}(q), Y_j \leq F^{-1}_{Y_j}(q))}{q} \\ \tau^U_{Y_i,Y_j} &= \lim_{q \to 1} \frac{P(Y_i > F^{-1}_{Y_i}(q), Y_j > F^{-1}_{Y_j}(q))}{q} \end{aligned} \tag{2.4}$$

2.2. Factor copulas

Factor copulas are a family of copula models for which the copula function $C(u_1, \ldots, u_N)$ is based on a latent factor structure as defined in Oh and Patton (2017, p. 140 ff).

Consider a set of artificial variables $X_i, i = 1, \ldots, N$ which linearly depend on some latent factors $Z_k, k = 1, \ldots, K$ and some error e_i such that $X_i = \sum_{k=1}^K \beta_{ik} Z_k + \epsilon_i$, with the linear coefficients β_{ik}. The latent variables Z_k and the error term ϵ_i follow some parametrized distributions, e.g. $\epsilon_i \overset{iid}{\sim} F_\epsilon(\boldsymbol{\gamma_\epsilon})$ and $Z_k \sim F_{Z_k}(\boldsymbol{\gamma_{Z_k}})$ with $Z_i \perp Z_j \forall i \neq j$, $Z_k \perp \epsilon_i \forall i, k$ and $\boldsymbol{\gamma_\epsilon}$, $\boldsymbol{\gamma_{Z_k}}$ some distribution specific parameter vectors.

The joint probability function $F_X(x_1, \ldots, x_N)$ of the artificial variables can then be expressed in terms of its marginal distributions $F_{X_i}(x)$ and a factor copula function $C_U(u_1, \ldots, u_N)$ such that $F_X(x_1, \ldots, x_N) = C_U(F_{X_1}(x_1), \ldots, F_{X_N}(x_N); \boldsymbol{\theta})$.

The parameters of the factor model are collected in a parameter vector $\boldsymbol{\theta} = (\beta_{11}, \ldots, \beta_{i1}, \ldots \beta_{ik}, \boldsymbol{\gamma}'_{\boldsymbol{Z_1}}, \ldots, \boldsymbol{\gamma}'_{\boldsymbol{Z_K}}, \boldsymbol{\gamma}'_{\boldsymbol{\epsilon}})'$. It consists of all linear coefficients and the distributional parmaters of the error term and the laten variables. The number of latent variables K and the distribution functions $F_{Z_1}, \ldots, F_{Z_k}, F_\epsilon$ are hyper-parameters of the model which have to be chosen prior to the estimation.[2]

[2] Oh and Patton (2017, p. 143ff) provide a heuristic of finding the number of latent variables by analyzing so called *scree-plots*: Ordered eigenvalues from the sample rank-correlation matrix of the data.

The artifical set of variables $\boldsymbol{X}$ is only used to construct the factor copula function $C_U(u_1, \ldots, u_N)$. The parameters of the factor strucutre are choosen in such a way that the resulting copula function fits the copula of the observable data $\boldsymbol{Y}$. The copula function can then be used to model the joint distribution of the observable variables such that $F_Y(y_1, \ldots, y_n) = C_U(F_{Y_1}(y_1), \ldots, F_{Y_N}(y_N))$. Once the factor copula function is determined, the artificial variables and its marginal distributions $F_{X_i}(x)$ are of no interest. Using matrix notation, the model can be summarized in the following set of equations:

$$\begin{aligned} \boldsymbol{Y} &= (Y_1, \ldots, Y_N)' \\ \boldsymbol{X} &= (X_1, \ldots, X_N)' = \boldsymbol{\beta Z} + \boldsymbol{\epsilon} \\ F_Y &= C_U(F_{Y_1}(y_1), \ldots, F_{Y_N}(y_N); \boldsymbol{\theta}) \\ F_X &= C_U(F_{X_1}(x_1), \ldots, F_{X_N}(x_N); \boldsymbol{\theta}) \end{aligned} \tag{2.5}$$

To model the joint probability $F_Y(\boldsymbol{y})$, a two-stage estimation can be applied: First, the marginal distributions $\hat{F}_{Y_i}$ are estimated parametrically or non-parametrically, e.g. by using some parametric model or the empirical distribution function. Second, the factor structure for the copula function is fitted to the data by finding the optimal $\hat{\boldsymbol{\theta}}$. Usually a closed form of the factor copula does not exist. Therefore, one has to rely on simulation based estimation methods as described in section 2.4.

This aproach allows for a variety of different dependence structures and can be applied to high dimensional data. An upper bound for the number of parameters $P = |\boldsymbol{\theta}|$ to be estimated is given by the size of the factor matrix and the number of additional free distributional paramters and it holds that $P \leq (NK + |\boldsymbol{\gamma_{Z_1}}| + \cdots + |\boldsymbol{\gamma_{Z_K}}| + |\boldsymbol{\gamma_\epsilon}|)$. To reduce the number of parameters, Oh and Patton (2017) present two restrictions on $\boldsymbol{\beta}$: the restrictive *equidependence* and the less restrictive *block-equidependence* model.

For the first model it is assumed that $K = 1$ and $\boldsymbol{\beta} = (\beta, \ldots, \beta)'$. Thus, the model consists of a single latent factor and a single factor loading β which is the same for all variables. This implies equal pairwise dependencies for all observable variables.

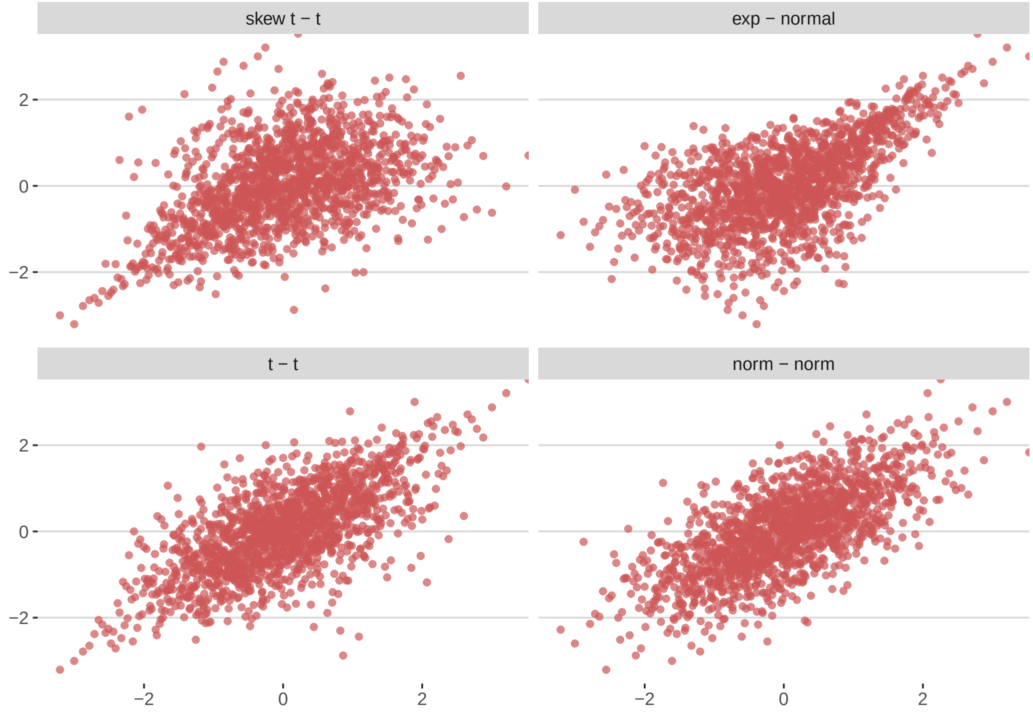


Figure 2.1: Illustration of different equidependence factor copula models with $N = 2$, $\beta = 1.5$, $Y_i \sim N(0, 1)$ and different distributions for the latent variable and the error term.

Figure 2.1 shows four different simulations from a two dimensional one factor equidependence factor copula model. The marginal distributions are standard normal, the linear coefficient is fixed at $\beta = 1.5$ but the distributions of the latent variable and the error term differ. All models produce positive dependence but the symmetry and tail dependency differs through the choice of the distributions. The upper left panel are realizations from a factor copula model with a skew-t distributed latent factor and a t-distributed error term. The degrees of freedom are set to $df = 4$ and the skewness parameter to $\lambda = -0.8$. This produces strong assymetric tail dependencies. The bottom left panel produces symmetric tail dependencies since $\lambda = 0$. This results in an ordinary t-distribution for the latent variable. The bottom right panel shows realizations from a gaussian copula which results in a multivariate gaussian distribution. The last panel shows the combination of an exponential latent variable with a normal error term.

The second model, the block-equidependence model, is less restrictive and is suitable for variables which can be naturally partitioned into different groups.[3] The model assumes a common factor for all groups and a group specific factor for each group. Thus, each variable is only affected by two factors. For the matrix of factor loadings, it is further assumed that all variables in the same group have the same factor loading while variables in different groups can have different loadings. This implies equal pairwise intra-group dependencies while the pairwise inter-group dependencies can vary between the groups.

Formally, consider a partition of $\boldsymbol{Y} = (Y_1, \ldots, Y_N)'$ into M groups. A single variable can then be written as Y_i^j, where $i = 1, \ldots, N$, $j = 1, \ldots, M$. The value k_j is the number of variables in group j and it holds $\sum_{j=1}^M k_j = N$. Then the factor copula model can be summarized as:

$$\begin{aligned}
\boldsymbol{X} &= (X_1^1, \ldots, X_{k_1}^1, X_{k_1+1}^2, \ldots, X_{k_1+k_2}^2, \ldots, \ldots, X_N^M)' = \boldsymbol{\beta}\boldsymbol{Z} + \boldsymbol{\epsilon} \\
\boldsymbol{Z} &= (Z_0, Z_1, \ldots, Z_M)' \\
X_i^j &= \beta_j Z_0 + \beta_{M+j} Z_j + \epsilon_i \\
\boldsymbol{\beta} &= \begin{pmatrix}
\beta^1 & \beta^{M+1} & 0 & \cdots & 0 \\
\beta^1 & \beta^{M+1} & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\beta^1 & \beta^{M+1} & 0 & \cdots & 0 \\
\beta^2 & 0 & \beta^{M+2} & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\beta^M & 0 & 0 & \cdots & \beta^{M+M} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\beta^M & 0 & 0 & \cdots & \beta^{M+M}
\end{pmatrix},
\end{aligned} \tag{2.6}$$

where the matrix $\boldsymbol{\beta}$ is of size $N \times (M+1)$ but with only $2M$ actual factor loadings.

2.3. Copula models for multivariate time series

Sklar's theorem in (2.1) which shows the link between a copula function and the multivariate distribution can be easily extended to the multivariate time series case: Consider the random vector $Y_t = (Y_{1t}, \ldots, Y_{Nt})$. The goal is to

[3] E.g. this could be stock market prices grouped into different industry sectors.

model the conditional multivariate distribution of $\boldsymbol{Y}_t|\mathcal{F}_{t-1}$, where the σ-algebra $\mathcal{F}_{t-1}$ possibly contains past information and information from other exogenous variables $\{\boldsymbol{Y}'_{t-1}, \boldsymbol{Y}'_{t-2}, \dots, \boldsymbol{X}'_t, \boldsymbol{X}'_{t-1}, \dots\}$.

Using the definition of a conditional copula as in (**Patton2006**) one can write

$$F_{Y_t|\mathcal{F}_{t-1}}(y_1, \dots, y_N) = C_{Y_t|\mathcal{F}_{t-1}}(F_{Y_1|\mathcal{F}_{t-1}}(y_1), \dots, F_{Y_N|\mathcal{F}_{t-1}}(y_N)).$$

To have a valid conditional multivariate distribution, the conditioning set must be the same for the marginal distributions and the copula (Patton 2009, p. 772).

For this work we use a semiparametric copula-based multivariate dynamic model as described in Chen and Fan (2006, p. 129 ff). The conditional means and variances of $\boldsymbol{Y}_t|\mathcal{F}_{t-1}$ are estimated parametrically. If we denote the parametrized conditional mean of a single variable as $\mu_{it} = E(Y_{it}|\mathcal{F}_{t-1}; \boldsymbol{\phi})$ and the parametrized conditional standard deviation as $\sigma_{it} = \sqrt{V(Y_{it}|\mathcal{F}_{t-1}; \boldsymbol{\phi})}$ we can write the multivariate time series as:

$$\boldsymbol{Y}_t = \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \boldsymbol{\eta}_t, \tag{2.7}$$

with $\boldsymbol{\sigma}_t = \text{diag}(\sigma_{1t}, \dots, \sigma_{Nt})$. The innovations $\boldsymbol{\eta}_t = (\eta_{1t}, \dots, \eta_{Nt})'$ have a standardized conditional distribution which can be modelled with a conditional copula.

For this model, the marginal distributions of the innovations are modelled with the empirical distribution function (EDF) and it is assumend that they are independent of past informtion and iid distributed such that $F_{\boldsymbol{\eta}_t} = F_{\boldsymbol{\eta}} \forall t = 1, \dots, T$.

In other words, the observations are first filtered by removing serial dependence or volatility clustering such that the leftover standardized innovations are independent of past information. Finally, the innovations are modeled using a parametric copula and nonparametric rank based estimates of the marginal distributions. The cdf of the innovations can be expressed in terms of a copula and the marginal distributions such that $F_\eta(x_1, \dots, x_N) = C(F_{\eta_1}(x_1), \dots, F_{\eta_N}(x_N); \theta_t)$.

2.4. Simulated methods of moments estimation for factor copulas

Estimation methods used for copula models depends on the degree of parametrization: For fully parametrized models for the copula and the marginal distributions maximum likelihood or multi-stage maximum likelihood is used. But one can also non parametrically estimate the marginal distributions and combine them with a parametric copula as shown in the previous section.

In this case, pseudo-maximum likelihood estimation can be used. If a closed form functional relation of spearman's rho or kendall's thau to the copula parameters is available, one can also solve the system directly by using a method of moments approach. For the SMM approach, the population based statistics are replaced by their sample counterparts (inversion method).

For the factor copula model a closed form one to one mapping of the copula's parameters θ to measures of dependency as defined in (2.3) - (2.4) is not available in general. If it were available, methods of moments or generalized methods of moments (if the number of moment conditions is larger than the number of parameters) could be applied (Oh and Patton 2013, p. 689f).

Instead one can use a set of scale-invariant empirical dependence measures calculated with simulations from the artificial variables $\boldsymbol{X}$ and compare them to the dependence measures obtained from the observable data $\boldsymbol{Y}$. Minimizing the weighted squared difference of the two dependency vectors yields an estimator for θ.

Formally, the estimator is given by

$$\hat{\boldsymbol{\theta}} = \arg\min Q(\boldsymbol{\theta}) = \arg\min \boldsymbol{g(\theta)' \hat{W} g(\theta)} \tag{2.8}$$

with

$$\boldsymbol{g(\theta)} = \hat{\boldsymbol{m}} - \tilde{\boldsymbol{m}}(\boldsymbol{\theta}), \tag{2.9}$$

where $\hat{\boldsymbol{m}}$ and $\tilde{\boldsymbol{m}}$ are vectors of empirical dependency measures computed from the observable and the simulated data respectively.

For the dependency measures one uses the empircal counterparts of Spearmans Rho as in 2.2 and lower and upper tail dependency as stated in 2.4. Other invariant measures which only dependend on the copula, e.g. Kendall's Tau, can also be used.

The vector $\boldsymbol{m}$ consists of a vectorized set of pairwise dependence vectors $\boldsymbol{\delta}_{i,j}$, where $i, j \in \{1, \ldots, N\}, j < i$.

For an unrestrictive model there exist $0.5 * N * (N-1)$ vectors of dependencies, one for each unique pair of variables. For the simpler equidependence model, which assumes the same factor loading for all variables, one can average over all vectors such that we can write $\bar{\boldsymbol{\delta}} = \frac{2}{N*(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \boldsymbol{\delta}_{i,j}$. This results in a single vector of dependencies for $\boldsymbol{m}$.

For the bloc-equidependence model the final number of dependency vectors is M since one can average over all intra- and intergroup dependencies. For each group $s = 1, \ldots, M$, we can write

$$\bar{\boldsymbol{\delta}}_s = \frac{1}{M}(\underbrace{\sum_{r=1,r\neq s}^{M} \frac{1}{k_s k_r} \sum_{i=1}^{k_s} \sum_{j=1}^{k_r} \boldsymbol{\delta}_{is,jr}}_{\text{intergroup dependencies}} + \underbrace{\frac{2}{k_s(k_s-1)} \sum_{i=k_s}^{k_s-1} \sum_{j=i+1}^{k_s} \boldsymbol{\delta}_{is,js}}_{\text{intragroup dependencies}}), s, r = 1, \ldots, M. \tag{2.10}$$

Oh and Patton (2013, p. 691ff) showed that given some assumptions the SMM estimator is weakly consistent and asymptotically normal distributed. The assumptions ensure that for both the iid and the time series case the sample dependency measures converge in probability to their theoretical population values. Further it is important to note, that it is assumed that the population version of the moment conditions (2.9) is differentiable at the true parameter $\boldsymbol{\theta}_0$ while this is not true for the finite sample version.

The convergence can be summarized as:

$$\hat{\boldsymbol{\theta}} \sim N(\boldsymbol{\theta_0}, (\frac{1}{T} + \frac{1}{S})\boldsymbol{\Omega}) \quad \text{for} \quad T, S \to \infty, \tag{2.11}$$

with covariance matrix $\boldsymbol{\Omega} = \boldsymbol{(G'WG)^{-1}G'W'\Sigma WG(G'WG)^{-1}}$, where $\boldsymbol{G} = \frac{\partial \boldsymbol{g(\theta)}}{\partial \boldsymbol{\theta}_0}$ is the Jacobian matrix of the first order derivitives of the moments function $\boldsymbol{g(\theta)}$ and $\boldsymbol{\Sigma}$ is the asymptotic variance of the moments estimator $\hat{\boldsymbol{m}}$.

The matrix $\boldsymbol{W}$ is a positive definite weight matrix and can be choosen to be the identity matrix. If the efficient weight matrix $\boldsymbol{W} = \boldsymbol{\Sigma}^{-1}$ is used the aymptotic variance simplifies to $\boldsymbol{\Omega} = \boldsymbol{(G'WG)^{-1}}$.

It is crucial to note, that the standard error of the copula parameters $\hat{\boldsymbol{\theta}}$ is not affected by the error arriving from estimating $\hat{\boldsymbol{\phi}}$, which is the parameter vector for the marginal models for the conditional mean and variance of $\boldsymbol{Y}_t$ (see (2.7)). The asymptotic results as they are state here are ony valid if the marginal distribution of the residuals is estimated non-parametically by using the empirical distribution function.

The asymptotic variance $\boldsymbol{\Sigma}$ can be estimated using an iid bootstrap procedure:

```
Input: ${\hat{\boldsymbol{\epsilon}}_t}_{t=1}^T$
Output: $\hat{\boldsymbol{\Sigma}}_{T,B}$
$\hat{\boldsymbol{m}}_T \leftarrow$ compute sample moments from residuals $\{\hat{\boldsymbol{\epsilon}}_t\}_{t=1}^T$ ;
for $b \leftarrow 1$ to $B$ do
    $\{\hat{\boldsymbol{\epsilon}}_t^{(b)}\}_{t=1}^T \leftarrow$ sample $T$ values with replacement from $\{\hat{\boldsymbol{\epsilon}}_t\}_{t=1}^T$;
    $\hat{\boldsymbol{m}}_T^{(b)} \leftarrow$ compute sample moments from bootstrap sample $\{\hat{\boldsymbol{\epsilon}}_t^{(b)}\}_{t=1}^T$;
    $\boldsymbol{g}_T^{(b)} \leftarrow \hat{\boldsymbol{m}}_T^{(b)} - \hat{\boldsymbol{m}}_T$;
end
$\hat{\boldsymbol{\Sigma}}_{T,B} \leftarrow \frac{T}{B}\sum_{b=1}^{B}(\boldsymbol{g}_{T,B}\boldsymbol{g}'_{T,B})$;
return $\hat{\boldsymbol{\Sigma}}_{T,B}$;
```

Algorithm 1: Bootstrap procedure for the estimation of $\hat{\boldsymbol{\Sigma}}_{T,B}$: The asymptotic covariance matrix of the moment condition for the residual data.

The derivative $\boldsymbol{G}$ can be estimated using a numerical approximation arround the parameter estimator $\hat{\boldsymbol{\theta}}$. For the kth column we can write:

$$\hat{\boldsymbol{G}}_{T,S,k} = \frac{\boldsymbol{g}_{T,S}(\hat{\boldsymbol{\theta}} + \boldsymbol{e}_k \epsilon_{T,S}) - \boldsymbol{g}_{T,S}(\hat{\boldsymbol{\theta}} - \boldsymbol{e}_k \epsilon_{T,S})}{2\epsilon_{T,S}},$$

where $\boldsymbol{e}$ is the kth unit vector and $\epsilon_{T,S}$ the step size which is usually set to $\epsilon_{T,S} = 0.1$.

2.5. Structural break test for factor copulas

In the following section a structural break test based on Manner, Stark, and Wied (2017) for a change of the copula parameters in (2.7) is presented. Note that it is assumed that the functional form of the copula is time invariant while the copula's parameters θ_t for $t = 1, \dots, T$ can vary over time. The model presented in 2.3 allows for a wide variety of parametrization and copula functions. Here, we focus on the factor copula model and the SMM estimation procedure as presented in the previous sections.

The test is based on recursive estimations of the copula model at each timepoint $t \in \epsilon 1, \ldots, T$. θ_t is the estimated parameter using the data up to point t. The strictly positive trimming parameter ϵ has to be choosen such that the model parameters don't fluctutate too much just because of small sample sizes. The null hyopthesis states that the estimator is time invariant while the alternative indicates at least one significant break point in time.

$$H_0 : \theta_1 = \theta_2 = \cdots = \theta_T \qquad H_1 : \theta_t \neq \theta_{t+1} \text{ for some } t = \{1, \ldots, T\}$$

The the test statistics is the maximum of the scaled squared distance between the full model and the recursive estimator:

$$P = \max_{\epsilon T \leq t \leq T} \frac{t}{T}^2 T(\boldsymbol{\theta}_{t,S} - \boldsymbol{\theta}_{T,S})'(\boldsymbol{\theta}_{t,S} - \boldsymbol{\theta}_{T,S}). \tag{2.12}$$

An alternative test statistic is solely based on the moment generating functions and does not involve the estimation of a copula model:

$$M = \max_{\epsilon T \leq t \leq T} (\frac{t}{T})^2 T(\hat{\boldsymbol{m}}_t - \hat{\boldsymbol{m}}_T)'(\hat{\boldsymbol{m}}_t - \hat{\boldsymbol{m}}_T). \tag{2.13}$$

If the test statistic is larger than some critical value, the time t where the test statistic occurs can be interpreted as the detected breakpoint.

Under the null hypothesis and given some assumptions which are similar in spirit to the assumptions for the asymptotic distribtion as in (2.11) the test statistics converges in distribution to

$$P \xrightarrow{d} \max_{\epsilon T \leq t \leq T} (\boldsymbol{A}^*(t) - s\boldsymbol{A}^*(T))'(\boldsymbol{A}^*(t) - s\boldsymbol{A}^*(T)), \tag{2.14}$$

with $\boldsymbol{A}^*(t) = (G'WG)^{-1}G'W(A(t) - \frac{s}{\sqrt{k}}A(1))$ and $T, S \to \infty$, $\frac{S}{T} \to k$ or $\frac{S}{T} \to \infty$.

The distribution under the null hypothesis can be estimated using the following bootstrap procedure:

```
Input: $\hat{\boldsymbol{\epsilon}}_t; \alpha$
Output: $K$
compute full sample moments $\hat{\boldsymbol{m}}_T$;
if copula based test then
    $L \leftarrow (\hat{G}'\hat{W}\hat{G})^{-1}\hat{G}'$;
else
    $L \leftarrow 1$;
end
for $b \leftarrow 1$ to $B$ do
    generate bootstrap sample $\{\hat{\boldsymbol{\epsilon}}_t^{(b)}\}_{t=1}^{T}$;
    for $t \leftarrow T$ to $\epsilon T$ do
        compute recursive bootstrap sample moments $\hat{\boldsymbol{m}}_t^{(b)}$;
        $A_t^{(b)*} \leftarrow L\frac{t}{T}\sqrt{T}(\hat{\boldsymbol{m}}_t^{(b)} - \hat{\boldsymbol{m}}_T)$;
        $K_t^{(b)} \leftarrow (A_t^{(b)*} - \frac{t}{T}A_T^{(b)*})'(A_t^{(b)*} - \frac{t}{T}A_T^{(b)*})$;
    end
    calculate the maximum $K^{(b)}$ over all $K_t^{(b)}$;
end
calculate the $1 - \alpha$ sample quantile $K$ from all $K^{(b)}$;
return K;
```

Algorithm 2: Bootstrap procedure for the estimation of the critical value for the structural break test.

3. *factorcopula* - an R package for simulation and estimation of factor copulas

In this chapter we present the implementation of the previous methods. Using the programming language *R*, the functions are bundled in an R-package such that the methods can be easily installed and distributed. The validity of the package is tested in two simulation studies.

The package consists of a set of high level functions which can be used to construct, simulate and fit various factor copula models. The specification of the factor copula model is handled by the functions `config_factor`, `config_error` and `config_beta`. The two functions `fc_create` and `fc_fit` can be used to either simulate values from a factor copula or to fit a model to a dataset. For conducting the break test as described in section 2.5, the library offers the functions `fc_critval`, `fc_mstat` and `fc_pstat`. The former simulates critical values for either the moments or copula based test. The latter two calculate the recursive test statistics. A matrix of recursive θ estimates is needed for the calculation of the copula based test statistic. It can be obtained by recursively applying `fc_fit` to the data.

3.1. Copula specification and simulation

With the functions `config_factor`, `config_error` and `config_beta`, the user can define the distribution of the latent variables $\boldsymbol{Z}$, the error term $\boldsymbol{\epsilon}$ and the matrix of factor loadings $\boldsymbol{\beta}$. For the specification of the distributions, the function name of any available random number generator can be used. Additional arguments such as distributional parameters can be declared in a named list.

In the following example, a factor copula with three latent variables is defined. The first factor is skewed-t distributed, the second and third are standard normal distributed. The error term is t distributed with 4 degrees of freedom. The distributional parameters `df` and `lambda` of the skewed-t distribution are free parameters of the model.

```r
library(factorcopula)
Z <- config_factor(rst = list(nu = df, lambda = lambda),
                   rnorm = list(),
                   rnorm = list(),
                   par = c("df", "lambda"))
eps <- config_error(rt = list(df = 4))
```

Distributional parameters can either be fixed or passed as non-evaluated expressions. To distinguish free model parameters from fixed distributional parameters, an additional character vector with the name of the model parameters has to be passed to `config_factor` and `config_error`.

The random number generating functions must have an additional argument `n` which defines the number of observations to be simulated. This argument should not be set explicetly since the number of simulations is controlled by the value S later.

For the factor loadings the user can either manually construct a character matrix of parameters or one can use the function `config_beta`. Given a vector `k` and the number of latent variables K this functions constructs a suitable character matrix of zeros and some parameter names.

The vector k is of length N and defines the group for each observable variable. Therefore, an equidependence model can be specified with $k_N = (1, 1, \ldots, 1)$, an unrestrictive model with $k_N = (1, \dots, N)$ and a bloc-equidependence model with $k_N = (1, 1, \dots, 2, 2, \dots, M, M, \dots)$, where M is the number of groups.

Continuing the example, a bloc-equidependence model with $K = 3$, $N = 6$, $k_1 = k_2 = 3$ and $M = 2$ is constructed. Together with the specification of the latent variables this results in a bloc-equidependence model with a skewed-t distribution for the common factor and a standard normal distribution for each of the two group specific factors.

```
k <- c(1, 1, 1, 2, 2, 2)
beta <- config_beta(k, 3)
```

The function `fc_create` returns itself a random number generating function. To simulate values from it, the user has to specify a *named* vector θ of parameters, the number of simulations S and an optional random seed. Fixing the seed at some value always gives the same simulated numbers. The vector `theta` must be a named vector for which the names correspond to the model parameters specified during the configuration of the model.

In the example, $S = 10$ random values from the copula model are simulated. The loadings for the common factor are $\beta_1 = \beta_2 = 1$ and the group specific loadings β_3 and β_4 are set to zero.

```
copfun <- fc_create(Z, eps, beta)
theta <- c(beta1 = 1, beta2 = 1, beta3 = 0, beta4 = 0,
           df = 10, lambda = -0.8)
U <- copfun(theta, S = 10)
```

The simulation of new values is a time consuming part. To avoid unnescessary calls to the underlying random number generators, the function remembers the state of the Z and ϵ matrix over repeated function calls and only updates the values if nescessary. Thus if neither the seed nor the distributional parameters in `theta` change, the function uses the same random values from a previous call. This reduces the computation time at the cost of a higher memory consumption.[4]

[4] A simple local comparisson of an optimized and unoptimized function showed that given a fixed seed and no change in distributional parameters, the opimized version runs five times faster for $S = 10000$ and three times faster for $S = 100000$.

3.2. Optimization strategy

While the function `fc_create` is used to simulate values from a factor copula model given a parameter vector θ, the function `fc_fit` estimates θ via SMM as described in section 2.4. Given a copula specification and some observable data the function minimizes the weighted squared distance between the moments based on simulated data and the moments based on observable data as presented in equation (2.8).

Optionally, the function also estimates the standard errors by impementing the bootstrap algorithm [1]. The weight matrix W is always set to the identity matrix, since previous studies showed no significant improvement when the efficient weight matrix $W = \hat{\mathbf{\Sigma}}_{T,B}^{-1}$ is used (Oh and Patton 2013, p. 694).

As a backend, the method builts on top the `NLopt` optimization library which implements various algorithms for global, local or derivative-free optimization (see Johnson (2018) and Ypma (2014)). The choice of the optimization algorithms and the stopping criteria can be altered by the user via the arguments `control.first.stage` and `control.second.stage`. The authors of the library recommend a two step optimization procedure for global optimization: First, a global optimizer should be used to approximates the optimal region in which the global optimum lies. Second, a local derivative free optimizer can then be applied using the approximated solution from the first stage.

By default, `fc_fit` uses the *Multi-Level Single-Linkage* algorithm in the first step. The algorithm creates a sequence of optimal distributed starting values which are then passed to a local optimizer (Kucherenko and Sytsko 2005). For the local optimizer and the second stage, the *Subplex* algorithm is used, which is based on the popula Nelder-Mead Simplex procedure (Rowan 1990).

During the optimization process, many values from the factor copula model are simulated. To avoid numerical instablities, the random seed is kept fixed (app. to Oh and Patton 2013, p. 12f). As described in the previous section, the memory functionability of `fc_create` can save some computational costs, since redraws from the distributions are avoided if the distributional parameter don't change. This can improve the overall performance of the optimizataion process. Especially, if only the factor loadings $\boldsymbol{\beta}$ are optimized. In this case, the random number generators are only called once at the beginning of the optimization.

3.3. Simulation study

To illustrate the discussed methods and the validity and performance of the package, two simulation studies are performed: First, an equidependence factor copula model with varying dimensions and sample sizes is estimated repeatedly to show the consistency of the SMM procedure. Second, both the moments and copula based structural break test are applied once to simulated data from a bloc-equidependence model.

The DGP of the first study is based on a simple equidependence model with standard-normal marginal distributions, one skew-t distributed latent variable and a t-distributed error term. The copula model produces strong assymetric tail-dependencies. This is achieved by fixing the degrees of freedom at $df = 4$ and the skeweness at $\lambda = -0.8$. The single factor loading is set to $\beta = 1.5$. The only free parameter to be estimated is the factor loading. Thus we can write $\theta = \beta$. The model equations can be written as:

$$\begin{aligned} Y_i &\sim N(0,1) \; \forall \; i = 1, \dots, N \\ Z &\sim \text{skew-t}(df = 4, \lambda = -0.8) \\ \epsilon &\sim \text{t}(df = 4) \\ X_i &= \beta Z + \epsilon \end{aligned} \tag{3.1}$$

We repeat the simulation and estimation of β over a grid of different values for the number of variables N and the sample size T. Each simulation consists of $C = 1000$ Monte-Carlo replications. The number of simulations in the SMM is set to $S = 25000$. Finally, we get a vector of estimates $\hat{\beta}_{t,n,c}$ with $t \in (100, 1000, 10000), n \in (2, 3, 10), c \in (1, \dots, C)$.

Figure 3.1 shows the results for the first study. For each combination of N and T the kernel density estimator over all Monte-Carlo simulations is plotted. The bias $\hat{b}_{t,n} = \frac{1}{C} \sum_{c=1}^{C} \hat{\beta}_{t,n,c} - \beta$ and the standard deviation is printed in the top left corner.

The density mass centers arround the true value of $\beta = 1.5$. For all simulations the bias is zero or close to it. As the sample size increases, the deviation gets smaller and the SMM estimator converges to the true parameter. But a small sample size clearly gives unreliable results. Having more variables improves the quality of the estimator. This is due to the fact that the number of latent variables and parameters is constant and does not increase with the number of

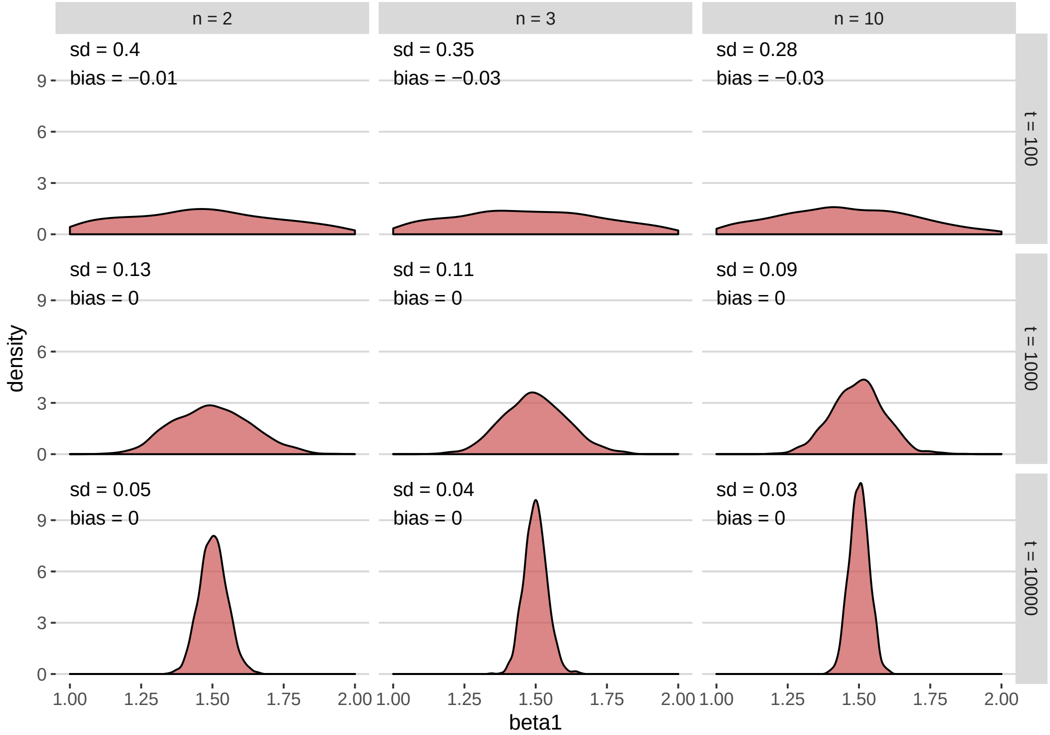


Figure 3.1: Approximated Monte-Carlo density estimators for $\hat{\beta}$ of an equidependence skew t - t factor copula model with $\beta = 1.5$, $S = 25000$, standard normal distributed marginals and different values for N and T. Each simulation is based on 1000 Monte-Carlo replications.

dimensions. Thus, with larger N more information is available to estimate the single parameter.

For the second study, a more sophisticated model is presented to illustrate the effectiveness of the approach even for high dimensional problems and complicated dependence structures. Analogous to the empirical examples in Manner, Stark, and Wied (2017) and Oh and Patton (2017) the DGP is based on a *bloc-equidependence* model as described at the end of section 2.2. The model consists of $N = 21$ standard normal distributed variables which are partitioned in $M = 3$ groups of equal size. For the factor copula, the single common latent factor is aggain skew-t distributed with strong assymetric tail-dependency. The three group specific latent factors and the error term are t-distributed. Due to the bloc-equidependence structure, the number of factor loadings reduces from $0.5 * N * (N - 1) = 210$ to just $2M = 6$. All distributional parameters are fixed. Therefore, only the factor loadings are about to be estimated such that $\theta = (\beta_1, \ldots, \beta_6)$.

We chose $T = 1500$, $S = 25 \times T$ and a breakpoint of the copula parameters at $t = 1000$. Before the break, $\theta_{pre} = (0, 1, 1, 0, 1, 1)$ and after the break $\theta_{post} = (1.5, 1, 1, 1.5, 1, 1)$. Thus, only the intra- and interdependence for the first group increases from 0 to 1.5 while the remaining loadings stay constant.

The test statistics and critical values are based on three different recursive calculations over a range from $t = 300$ to $T = 1500$: First, all factor loadings are estimated. Second, a subset of θ is estimated recursively while fixing the common and group specific factor for the first group at the full sample estimates. Hence, for this calculation $\theta = (0, \beta_2, \beta_3, 0.88, \beta_5, \beta_6)$. Finally, to perform the moments based test, the empirical dependence vectors are calculated wihtout estimating the copula parameters.

We expect, that a breakpoint is detected arround $t = 1000$ only for the first and third recursive calculation. For the second calculation no breakpoint should be detected since the factor loadings of the second and third group are not affected by the simulated structural break.

Figure 3.2 shows the test statistics of the three different break tests for a single recursive run of the simulation for $t = 300, \dots, 1500$. The horizontal solid line indicates the critical value. Each critical value is calculated using $B = 2000$ bootstrap samples as described in algorithm [2]. The vertical line indicates the theoretical breakpoint at $t = 1000$.

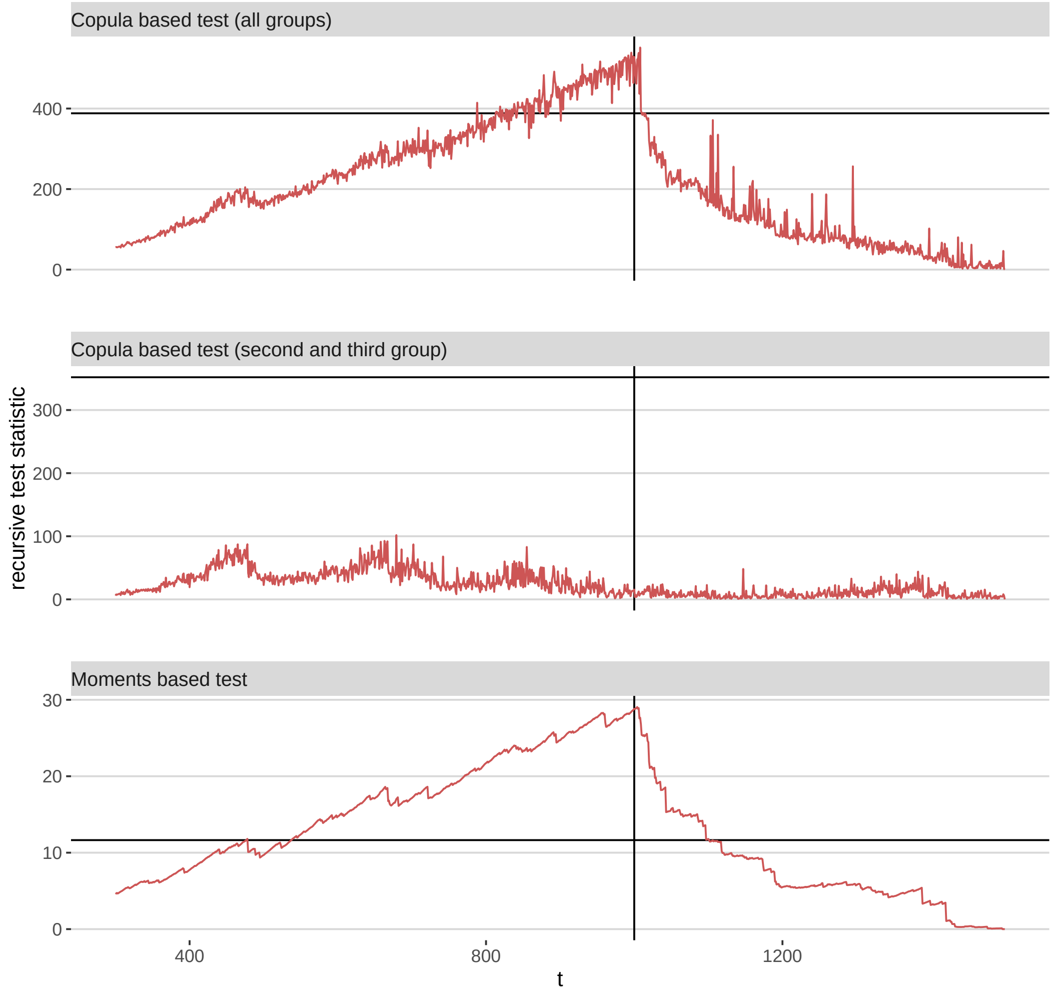


Figure 3.2: Illustration of a structural break test for a bloc-equidependence model with $N = 21$ and 3 groups of equal size. The theoretical breakpoint is at $t = 1000$ and is modelled as a change of the intra- and interdependency of the first group from 0 to 1.5. The first panel shows the copula based test based on all groups. The second panel the copula based test for only the second, and third group which are not affected by the structural break. The lower panel shows the moment based test.

As expected, both the full copula and the moments based test statistics reach their maximum arround $t = 1000$. In both cases the maximum lies above the critical value. The CV for the moments based test is relatively far away from the maximum while this is not true for the full copula based test. This could indicate that the moments based test is less sensitive while the copula based test is more restrictive. This observation is also confirmed in the empirical example in the following chapter. The copula based test statistics for the second and third group are clearly below their critical value. Hence, for this groups the null hypothesis of no parameter change cannot be rejected.

If the parameter vector is large, recursively estimating the copula model is computationally quite challenging and often results in numerical instabilities. Therefore, the test statistic can be substantially distorted which results in extreme outliers above the CV. Therefore, we advise to always perform a manual graphical analysis of the final test statistics. In addition, clear outliers can be detected by performing a smoothing method on the test statistics (e.g. running medians or smoothing splines).

Finally, table 3.1 shows the results for a copula model estimated on the full sample, before and after the theoretical breakpoint. For the model after the breakpoint the estimates are less precise due to the small sample size. This can also be seen in the relatively high value of the objective function Q.

coefficient	pre-break ($t \leq 1000$)	post-break ($t > 1000$)	full sample
β_1	0.00 (0.10)	1.96 (0.69)	0.00 (0.13)
β_2	1.17 (0.37)	0.88 (0.29)	1.28 (0.30)
β_3	0.96 (0.27)	0.83 (0.30)	0.99 (0.27)
β_4	0.03 (0.15)	1.55 (0.64)	0.88 (0.15)
β_5	0.92 (0.18)	1.01 (0.32)	0.92 (0.18)
β_6	1.04 (0.20)	1.13 (0.40)	1.09 (0.20)
Q	0.0008	0.0026	0.0025
T	1000	500	1500
S	37500	37500	37500

Table 3.1: Estimation results for the bloc-equidependence factor copula model before and after the breakpoint and for the full dataset. Standard errors in paranthesis (estimated with $B = 2000$ bootstrap samples).

4. Modelling topic dependencies over time with factor copulas

In this chapter we apply the previously discussed methods to a real aggregated dataset derived from the social media platform Facebook. First the dataset is presented. Second, we give a detailed description of the feature generation process and some descriptive overview. Third, the model setup is explaind and estimation results of various factor copula models applied to the residuals of the word frequencies are presented. Finally, we apply both the moments and copula based structural break test to the aggregated data.

4.1. The *btw17* social media dataset

The raw data consists of social media posts published by public pages on Facebook between January 2014 and December 2017. Using the official list of the candidates for the Bundestag election in 2017 (btw17), the account for each of the politicians was manually researched. Only candidates from the six factions in the *Bundestag* (CDU/CSU, SPD, Die Linke, Bündnis 90/ Die Grünen, AfD, FDP) are part of the study. Arround 84% of all 2516 candidates have an account on Facebook (see Stier et al. 2018, p. 16).

Due to API and privacy restrictions, only information from public pages can be accessed such that arround 52% of the social media accounts could be considered for the data collection. In addition to the candiates pages, 113 official pages from the political parties, both on the federal and regional level, were included.

The data collection took place on several days between 2017-11-21 and 2018-02-06. The web-scraping software is built on top of the *restfb* Java client library and makes calls to Facebook's official *Graph API* (**Facebook2018**; Allen and Bartels 2018). The posts are stored in a document orientated database on cloud-servers located in Germany. Besides the actual message, they contain a timestamp, the user-id of the author and the number of likes and shares the post has received uppon collecting it from the API.

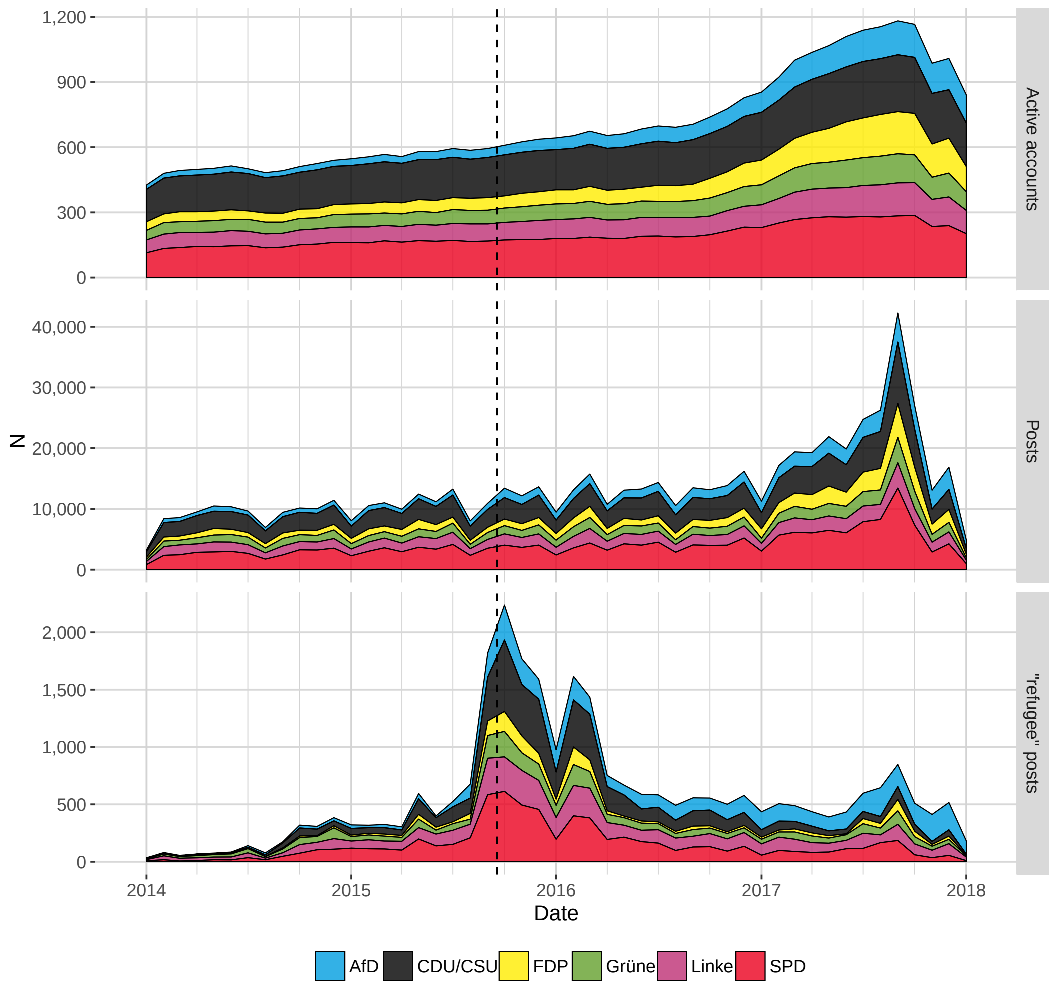
For this analysis, the data is restricted on textual posts only.[5]. This results in almost 664 thousand posts tagged with the party membership of their authors.

The first two panels of figure 4.1 show for each party the monthly number of active accounts and the monthly number of posts. An account was defined active if it has at least one post during the month considered.

In early 2014 approximately 500 accounts were active. This number increased steadily to roughly 750 accounts in mid 2016. From then until the election in

[5] A post can also consist of a foto, an album or an event.

September 2017 the number increased rapidly to almost 1200 active accounts followed by a drop after the election. Since we only collected accounts from politicians which were candidates in the btw17 the sharp rise of active accounts can be explained by the fact that many politicians opened an account just for the election campaign. After the election many candidates closed their accounts due to a failure in the elections. A similar pattern can be observed for the monthly number of posts.



Accounts and posts by party over time

Figure 4.1: Number of active accounts and number of posts per party and month. The bottom panel shows the number of posts matching the regular expression `flucht|fluecht`. The vertical line indicates the breakpoint detected by the moments based test.

The bottom panel of figure 4.1 shows the absolute number of monthly posts related to the topic “refugees”. We define a post to be refugee related if it matches the regular expression `flucht|fluecht`.[6] For example, a match occurs if a post contains the words *flüchtlingskrise* (refugee crisis), *fluchtursachen* (causes of flight), *flüchten* (to flee) or *flüchtlingsheime* (refugee hostels). A sharp rise of refugee related post can be observed in autumn and winter of 2015 and 2016. During this time many refugees entered Germany, fleeing from the war in Syria. This event is commonly labeled as the "german refugee crisis" in the media and the political discussions.

Table 4.1 summarizes the monthly numbers presented in figure **??**. For each party, it shows the overall number of posts, accounts, aggregated likes and shares for posts and the within party share of refugee related posts from 2014-01-01 to 2017-12-31. Allthough the right and left wing parties *AfD* and *Linke* have a relatively small number of accounts and posts they generate by far the greatest number of attention in terms of likes and shares. The social democratics party *SPD* has over twice more posts than the *AfD* but roughly generates only half of the likes and only a fifth of the shares. Looking at the share of refugee related posts one can see that both the far left and far right parties talk more about refugee related topics than the average.

party	posts	accounts	likes	shares	share of posts related to "refugees" (in %)
			(in Million)		
AfD	74724	162	18.36	7.59	6.96
CDU/CSU	169115	267	14.72	1.83	3.49
FDP	71083	201	6.32	0.77	2.39
Grüne	67188	139	4.03	1.28	4.62
Linke	84723	158	16.03	4.23	6.23
SPD	196805	290	10.11	1.57	3.66
All parties	663 638	1217	69.57	17.27	4.28

Table 4.1: Overall number of posts, active accounts, likes and shares over the observation period from "2014-01-01" - "2017-12-31". The last two columns show the fraction of posts matching the regular expression `flucht|fluecht`.

[6] Note that a posts's text was cleaned by removing links and stopwords, transforming umlauts and converting the text to lowercase letters.

4.2. Data processing and descriptive analysis

For each party we count the daily number of refugee related posts and divide it by the overall daily number of posts from this party. Thus, for each day, we get a relative within-party frequency of refugee related posts. The final dataset consists of $T = 1461$ rows and a column for each of the six parties. We use this statistic to approximate the importance of refugee related topics in the political discourse for this party. For example, a value close to 0.5 indicates that half of the political discourse from this party is refugee related.

A pairwise positive dependence between all parties could indicate that refugee related topics are equally relevant for all parties and that the discussions are driven by the same events. Whereas no dependence could indicate, that the importance is mainly party-specific and independent from outside discussions. Tail dependencies could indicate that only in the time of disruptive external events the importance of refugee related topics is equally high for alle parties.

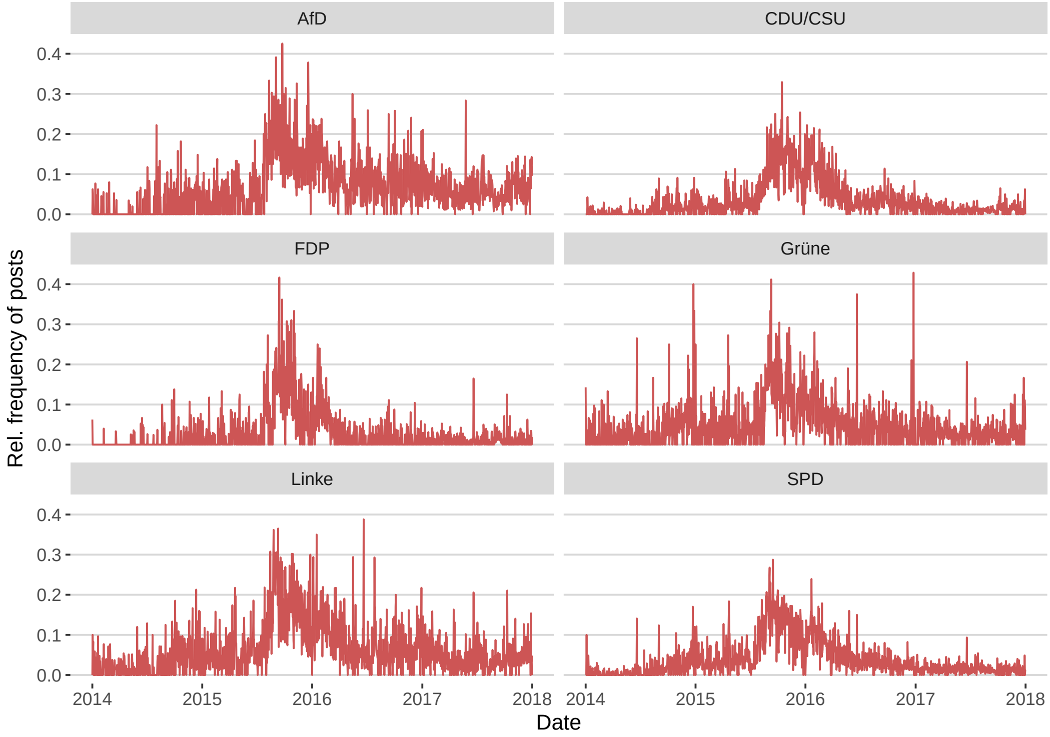
In the spirit of the copula time series model as discussed in 2.3 we first estimate univariate models for each of the time series. The factor copula dependency analysis is then performed on the residual information. By removing the time dependent conditional mean and standard deviation from the observations, it can then be assumed that the residuals are time invariant. The empirical distribution function is used to model the marginal distributions of the residuals.

We use a standard ARIMA-GARCH model for modelling the time series (Teräsvirta 2009). For the conditional variance we assume a GARCH(1, 1) process while specific ARIMA models are applied to model the conditional mean. The parts of the mean model are determined by running various models over a grid of different parameters. The model candidate with the lowest bayseian information criterion (BIC) is then choosen. Table 4.2 summarizes the final parameter values for the six time series model applied to the relative daily frequencies.

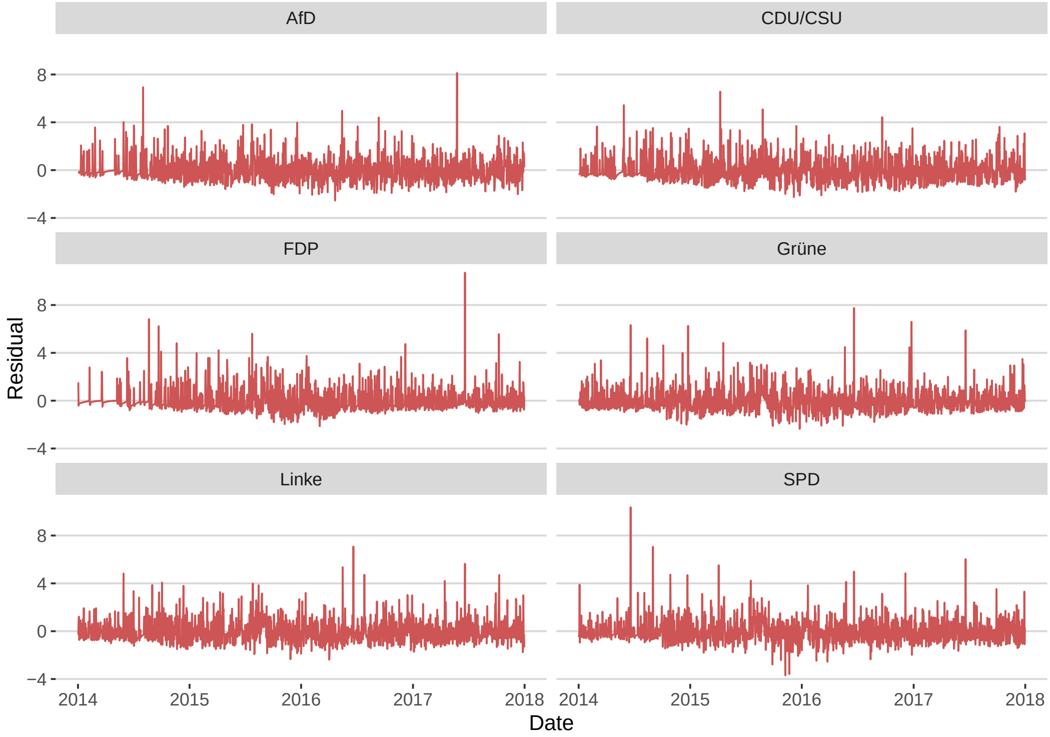
Party	AfD	CDU/CSU	FDP	Grüne	Linke	SPD
AR	4	1	2	3	2	1
I	1	1	1	1	1	1
MA	1	2	1	3	2	1

Table 4.2: ARIMA Model parameters for the ARIMA-GARCH(1,1) model used to estimate the standardized residuals.

Figure 4.2 shows the time series of relative frequencies and the residuals for each of the six parties after applying the ARIMA-GARCH(1, 1) models. Inline with the bottom panel of figure 4.1, refugee related topics start to become important in midth 2015 with peaks of up to 40% in late 2015. Espacially for the left and right wing parties, “Linke” and “AfD”, the topic stays important even after the crisis. The ARIMA-GARCH model seem to correctly fit the time series but some extreme positive outliers are prevalent.



(a) Relative daily within-party frequency of refugee related posts.



(b) Residuals of ARIMA-GARCH models applied to the time series.

Figure 4.2

Table 4.3 shows the overall pairwise dependency measures used also for the SMM procedure. The dependencies are very weak to weak.

	Rank -	Quantile-dependence			
Pairs	correlation	0.05	0.1	0.90	0.95
AfD-CDU/CSU	0.07	0.14	0.17	0.13	0.10
AfD-FDP	0.08	0.10	0.14	0.13	0.07
AfD-Grüne	0.03	0.07	0.16	0.09	0.10
AfD-Linke	0.08	0.07	0.15	0.19	0.08
AfD-SPD	0.09	0.11	0.16	0.16	0.08
CDU/CSU-FDP	0.17	0.16	0.25	0.18	0.14
CDU/CSU-Grüne	0.13	0.11	0.20	0.14	0.07
CDU/CSU-Linke	0.10	0.10	0.18	0.16	0.14
CDU/CSU-SPD	0.20	0.22	0.22	0.19	0.14
FDP-Grüne	0.10	0.08	0.19	0.15	0.10
FDP-Linke	0.10	0.10	0.16	0.17	0.10
FDP-SPD	0.12	0.19	0.23	0.18	0.07
Grüne-Linke	0.09	0.07	0.12	0.19	0.08
Grüne-SPD	0.12	0.14	0.21	0.16	0.14
Linke-SPD	0.15	0.16	0.16	0.23	0.21
Average	0.11	0.12	0.18	0.16	0.11

Table 4.3: Pairwise empirical dependencies for the six parties.

4.3. Results

First, various factor copula models are fitted to the complete dataset. Second, the best model is choosen and recurseviley applied to the data while fixing the distributional parameters at the complete sample estimates. A moments and copula based break test is performed. Finally, copula models are estimated for the pre-break and post-break period.

To dertermine the number of latent factors K, we analyze the ordered eigenvalues of the residual rank-correlation matrix, as described in Oh and Patton (2017, p. 148). One eigenvalue is above the threshold (see A.2). Thus, the analysis is restricted to factor copula models with one latent factor.

In figure 4.4 we summarize the results of applying various one-factor copula models to the complete sample of residuals. A restrictive equidependence and an unrestrictive model is performed.

	Equidependence			Unrestrictive		
	norm-norm	t-t	skewt-t	norm-norm	t-t	skewt-t
β_1	0.43	0.42	0.42	0.23	0.23	0.25
β_2	-	-	-	0.62	0.52	0.63
β_3	-	-	-	0.42	0.44	0.43
β_4	-	-	-	0.32	0.34	0.33
β_5	-	-	-	0.37	0.41	0.35
β_6	-	-	-	0.63	0.60	0.67
df	-	99	96	-	58	38
λ	-	-	-0.29	-	-	-0.36
Q	0.0036	0.0034	0.0031	0.1081	0.0979	0.0903

Table 4.4: Estimation results for different one-factor copula specifications.

Using the one-factor skew-t equidepndence model as

	Moments based test		Copula based test
	unrestrictive	restrictive	restrictive
test statistics	6.05	116.95	1.51
95%-CV	1.98	61.90	1.67 (p-value: 0.0685)
breakpoint	2015-09-18	2015-09-18	2015-03-13

Table 4.5: Summary of break point detection methods for the social media dataset.

coefficient	before ($t \leq 626$)	after ($t > 626$)	full model
β_1	0.39 (0.03)	0.41 (0.03)	0.42 (0.03)
λ	-0.55 (0.75)	-0.33 (0.54)	-0.29 (0.37)
Q	0.0029	0.0023	0.0031
T	626	835	1461
S	36525	36525	36525

Table 4.6: Estimation results for the btw17 dataset. A bloc-equidependence factor copula model is estimated before and after the breakpoint at $t = 626$. Standard errors in paranthesis (estimated with $B = 2000$ bootstrap samples). Degrees of freedom were fixed at the full model estimate of $df = 96$.

5. Discussion

To get better results much care has to be put into the optimiaztion algorithm and the numerical procedures. Frazier and Zhu (2017) proposes ways to use derivative based optimizataion procedures in cases of SMM where the objective is discontinuty.

Embrechts (n.d.) and Mikosch (2006) critizes the "hype" arround copulas.

A. Appendix

A.1. Notes on data and source code access

An online version of this thesis is publicly available as a git-repository under https://github.com/bonartm/factorcopula-thesis. The repository contains notes on how to install all dependencies. The source code files for the analyses are located online in the `source` folder.

Due to data restrictions by Facebook it is not possible to publish the original dataset of Facebook posts. However, the residuals of the ARIMA-GARCH models which were applied to the aggregated Facebook data are located online at `data/topics_residuals.rds`.

The methods for the simulation and estimation of factor copula models and the break test are not part of this repository. Instead, they are available via the R-package `factorcopula`. The package can be installed from Github. Further notes can be found under: https://github.com/bonartm/factorcopula (See also chapter 3).

For almost all estimation procedures, the HPC cluster of the Universiy of Cologne was utlizied (Achter et al. 2018). To simplify the workflow we wrote the R-package `cheopsr` which allows the execution of job scripts from within the local R environment. The package is available online at https://github.com/bonartm/cheopsr. To run the package, a Unix-like system and accesss rights to the HPC cluster are obligatory.

A.2. Additional figures

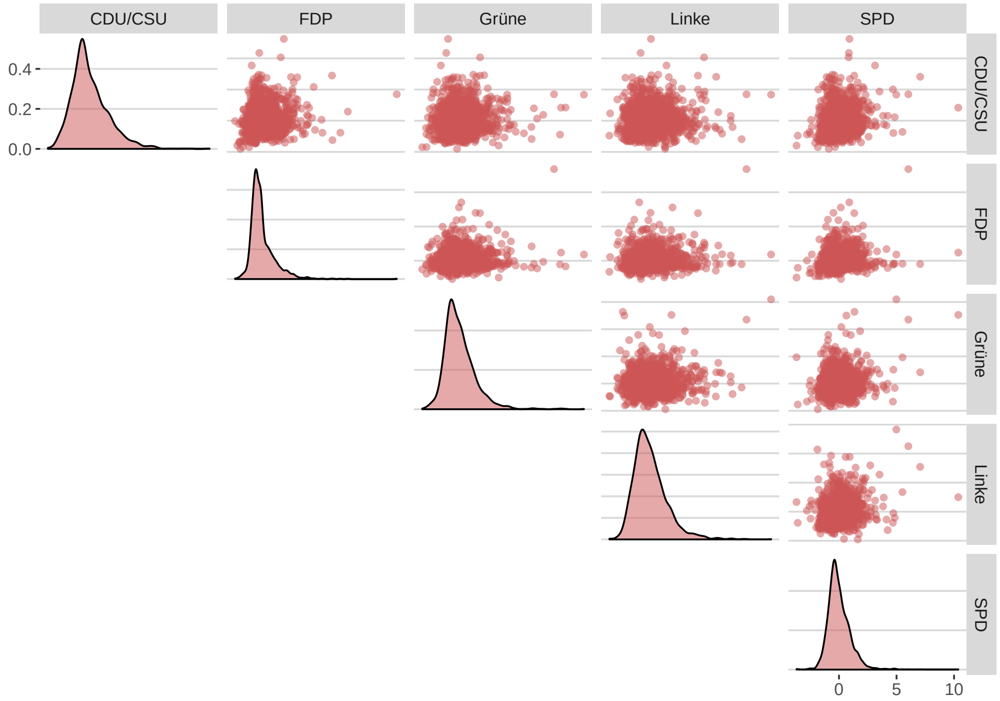


Figure A.1: Pairwise scatterplot of the estimated residuals.

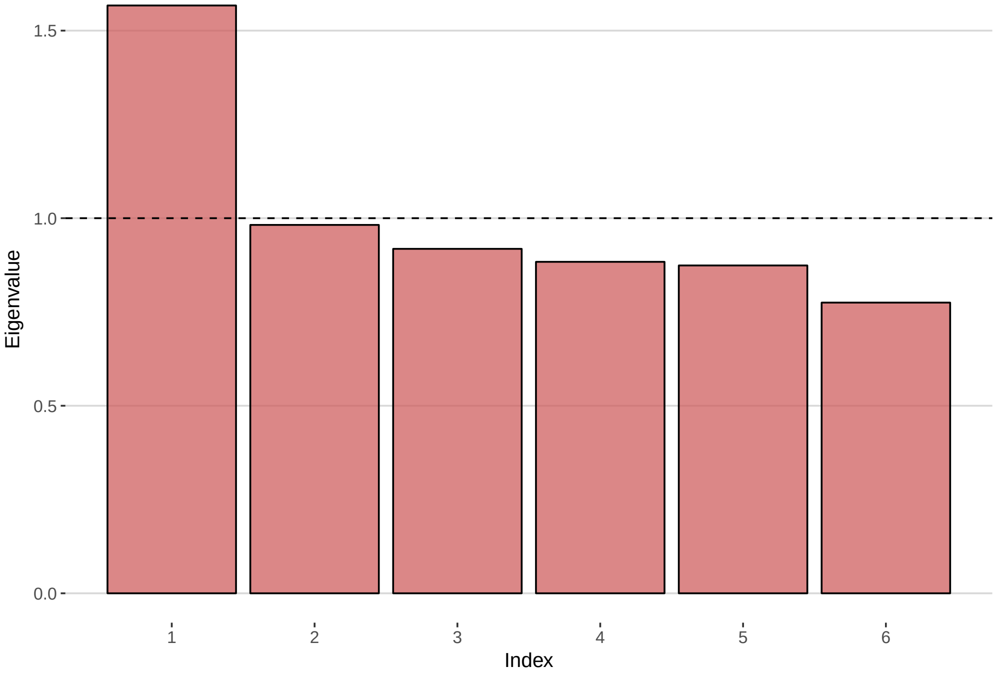


Figure A.2: Scree-plot of ranked eigenvalues based on the pairwise rank-correlation matrix.

B. References

Achter, Viktor, Stefan Borowski, Lech Nieroda, et al. (2018). *CHEOPS Cologne High Efficient Operating Platform for Science.* https://rrzk.uni-koeln.de/sites/rrzk/HPC_Projekte/CHEOPS_Brief_Instructions.pdf. [Online techreport; accessed 12-May-2018].

Allen, Mark and Norbert Bartels (2018). *RestFB - a pure Java Facebook Graph API client.* http://restfb.com. [Online documentation; accessed 12-May-2018].

Bonart, Malte (2018). *factorcopula - an R package for high dimensional depence modelling with factor copulas.* https://github.com/bonartm/factorcopula. [Online developement version; accessed 12-May-2018].

Chen, Xiaohong and Yanqin Fan (2006). "Estimation and model selection of semiparametric copula-based multivariate dynamic models under copula misspecification". In: *Journal of Econometrics* 135.1, pp. 125–154.

Embrechts, Paul. "Copulas: A Personal View". In: *Journal of Risk and Insurance* 76.3, pp. 639–650.

Frazier, David T. and Dan Zhu (2017). *Derivative-Based Optimization with a Non-Smooth Simulated Criterion.* https://arxiv.org/abs/1708.02365. [Online working paper; accessed 12-May-2018].

Joe, Harry (2015). *Dependence Modeling with Copulas.* Chapman & Hall/CRC Monographs on Statistics & Applied Probability 134. Taylor & Francis.

Johnson, Steven G. (2018). *The NLopt nonlinear-optimization package.* https://nlopt.readthedocs.io/en/latest/. [Online documentation; accessed 12-May-2018].

Krupskii, Pavel and Harry Joe (2013). "Factor copula models for multivariate data". In: *Journal of Multivariate Analysis* 120.C, pp. 85–101.

Kucherenko, Sergei and Yury Sytsko (2005). "Application of Deterministic Low-Discrepancy Sequences in Global Optimization". In: *Computational Optimization and Applications* 30.3, pp. 297–318.

Manner, Hans, Florian Stark, and Dominik Wied (2017). *Testing for Structural Breaks in Factor Copula Models.* https://www.wiwi.uni-due.de/fileadmin/fileupload/VWL-WIPO/WiWi-Kolloquium/Manner_Stark_Wied___2017_.pdf. [Online working paper; accessed 12-May-2018].

Mikosch, Thomas (2006). "Copulas: Tales and facts". In: *Extremes* 9.1, pp. 3–20.

Nelsen, Roger B. (1999). *An Introduction to Copulas.* Lecture notes in statistics 139. Berlin, Heidelberg: Springer.

Oh, Dong Hwan and Andrew J. Patton (2013). “Simulated Method of Moments Estimation for Copula-Based Multivariate Models”. In: *Journal of the American Statistical Association* 108.502, pp. 689–700.

– (2017). “Modeling Dependence in High Dimensions With Factor Copulas”. In: *Journal of Business & Economic Statistics* 35.1, pp. 139–154.

Patton, Andrew J. (2009). “Copula–Based Models for Financial Time Series”. In: *Handbook of Financial Time Series*. Ed. by Thomas Mikosch, Jens-Peter Kreiß, Richard A. Davis, et al. Berlin, Heidelberg: Springer, pp. 767–785.

Rowan, Thomas Harvey (1990). “Functional Stability Analysis of Numerical Algorithms”. PhD thesis. Austin, TX, USA.

Sempi, Carlo (2011). “Copulas”. In: *International Encyclopedia of Statistical Science*. Ed. by Miodrag Lovric. Berlin, Heidelberg: Springer, pp. 302–305.

Sklar, Abe (1959). “Fonctions de répartition à n dimensions et leurs marges”. In: *Publications de l’Institut de Statistique de L’Université de Paris* 8, pp. 229–231.

Stier, Sebastian, Arnim Bleier, Malte Bonart, et al. (2018). “Systematically Monitoring Social Media: the case of the German federal election 2017”. In: *GESIS Papers* 2018/04, p. 25.

Teräsvirta, Timo (2009). “An Introduction to Univariate GARCH Models”. In: *Handbook of Financial Time Series*. Ed. by Thomas Mikosch, Jens-Peter Kreiß, Richard A. Davis, et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 17–42.

Ypma, Jelmer (2014). *Introduction to nloptr: an R interface to NLopt*. https://cran.r-project.org/web/packages/nloptr/vignettes/nloptr.pdf. [Online documentation; accessed 12-May-2018].

C. Statutory Declaration

Eidesstattliche Versicherung

Hiermit versichere ich an Eides Statt, dass ich die vorliegende Arbeit selbstständig und ohne die Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden. Ich versichere, dass die eingereichte elektronische Fassung der eingereichten Druckfassung vollständig entspricht.

Köln, den 12. Mai 2018

(Malte Bonart)