Selection of Mixed Copula Model via Penalized Likelihood

Koushani Das and Siddharth Somani
May 6th, 2022

Abstract

The above study is based on the paper, Selection of Mixed Copula Model via Penalized Likelihood. We discuss the importance of copulas to understand the dependency among several variables. We study the properties of different copulas and establish the need to consider multiple copula models instead of a single copula. We discuss the method of fitting a penalized log likelihood function to estimate the copula models. We test our method on real data using US, Japan and UK market indices.

1. Introduction

To capture the complex dependence structure among variables, a copula approach has been used recently in many applied fields, in finance and economics. Copula modeling is used in hedge fund indices to study positive quadrant dependence found in financial indices, measure the portfolio value-at-risk, model the term structure of interest rates, study the dependence among international stock markets and exchange rate time series, cross-national dependence structure of asset returns in international financial markets; discover asymmetric dependence between two asset returns during market downturns and market upturns.

To describe the dependence structure more flexibly, researchers have proposed using a mixed copula which is a linear combination of several copula families. A combination of Gaussian, Gumbel, Clayton and Frank copula has been proposed to measure the dependence patterns across financial markets. The biggest advantage of a mixed copula model is that it can nest different copula shapes. For instance, Gaussian and Gumbel mixed copula can improve a single Gaussian dependence structure by allowing possible right tail dependence. Therefore, empirically, a mixed copula is more flexible to model dependence structure and can deliver better descriptions of dependence structure than an individual copula.

A motivation of this study comes from an analysis of real financial data, consisting of monthly measurements of international stock market indices: S&P500 (U.S.), FTSE 100 (UK), Nikkei (Japan), and Hang Seng (Hong Kong) from the Center for Research in Security Prices (CRSP). Of interest examining the existence of the comovement of returns among these four international stock markets, which is one of the popular topics in financial econometrics; see Hu (2006) and Chollete, Pena, and Lu (2005) for more details. Therefore, to study the comovement with various dependence structures, a copula approach is appropriate.

Further, we discuss how to choose an appropriate parametric copula because the distribution from which each data point is drawn is unknown. To attenuate this problem, there have been some efforts in the literature to choose an appropriate individual copula. The first is to propose a data-driven copula selection method via penalized likelihood plus a shrinkage operator.

The main goal is to select the **best mixed copula** among all candidate copulas to capture the dependence structure of the given data. The best mixed copula can be selected by choosing the one with the **highest likelihood**. When a fitted mixed copula contains some component copulas with small weights which imply small contribution to dependence structure, as expected, these components should not be included in the mixed copula. To filter out the components with small weights, some constraints should be added on all weights, such as penalty functions. Therefore, a likelihood function can be formulated as a form of a mixed copula including all candidate copulas. Furthermore, a penalized likelihood function is constructed by adding some appropriate penalty functions and constraints for weights. By maximizing the penalized likelihood function, copulas with small weights can be removed by a thresholding rule (shrinkage operator) and parameters remaining are estimated. In such a way, model selection and parameter estimation can be done simultaneously

We test the working of a model by fitting data generated by copula models as well as their mixtures against an identical distribution, as well as a misspecified model where one or more components are missing.

Finally, we try to fit the copula models for the Us, Uk and Japan markets. According to the type of copula one or more initial values are proposed and then final parameters are estimated. The estimated values are compared with those obtained by the pseudo-maximum likelihood method.

Review of Mixed Copula

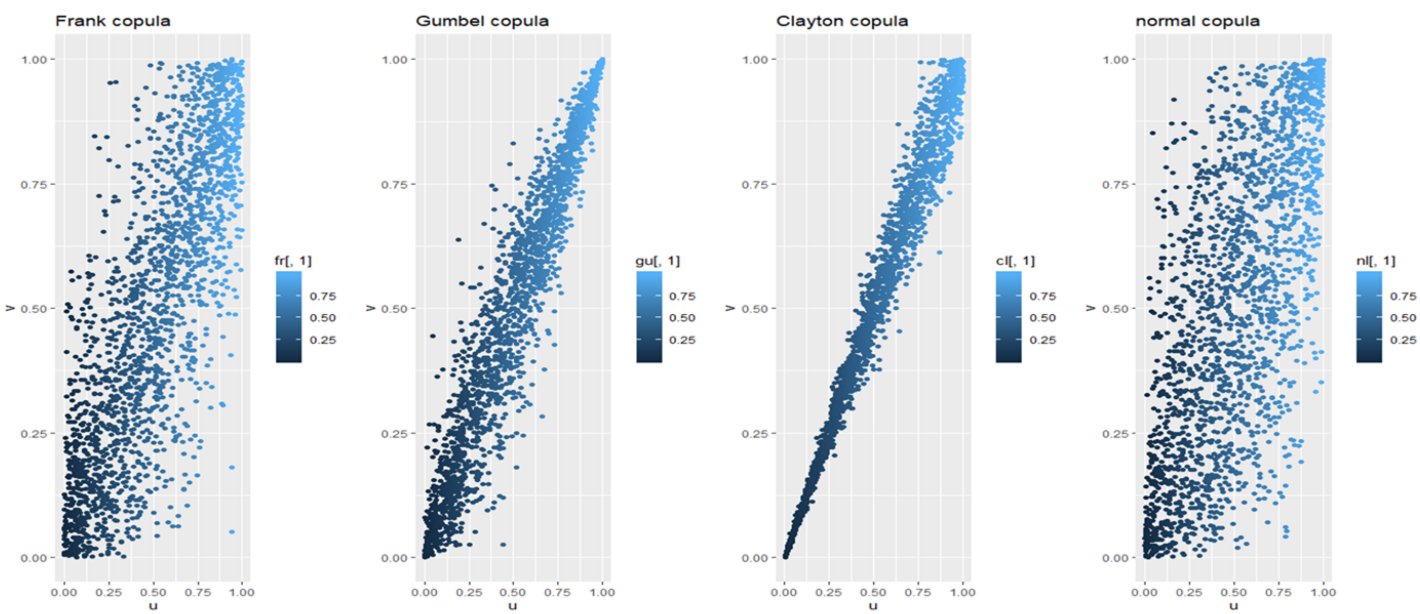
Let $\{ X_t \}$ be independent p-dimensional vectors of random variables with $X_t = (X_{t1},\ldots,X_{tp})$,. Let $f(x)$ and $F(x)$ be the joint density and distribution of X, respectively, and $f_j(x_j)$ and $F_j(x_j)$, $1 \leq j \leq p$, be the marginal density and distribution of X_j, respectively.

A mixed copula is a linear combination of several copula families. Mathematically, a mixed copula function is formulated as

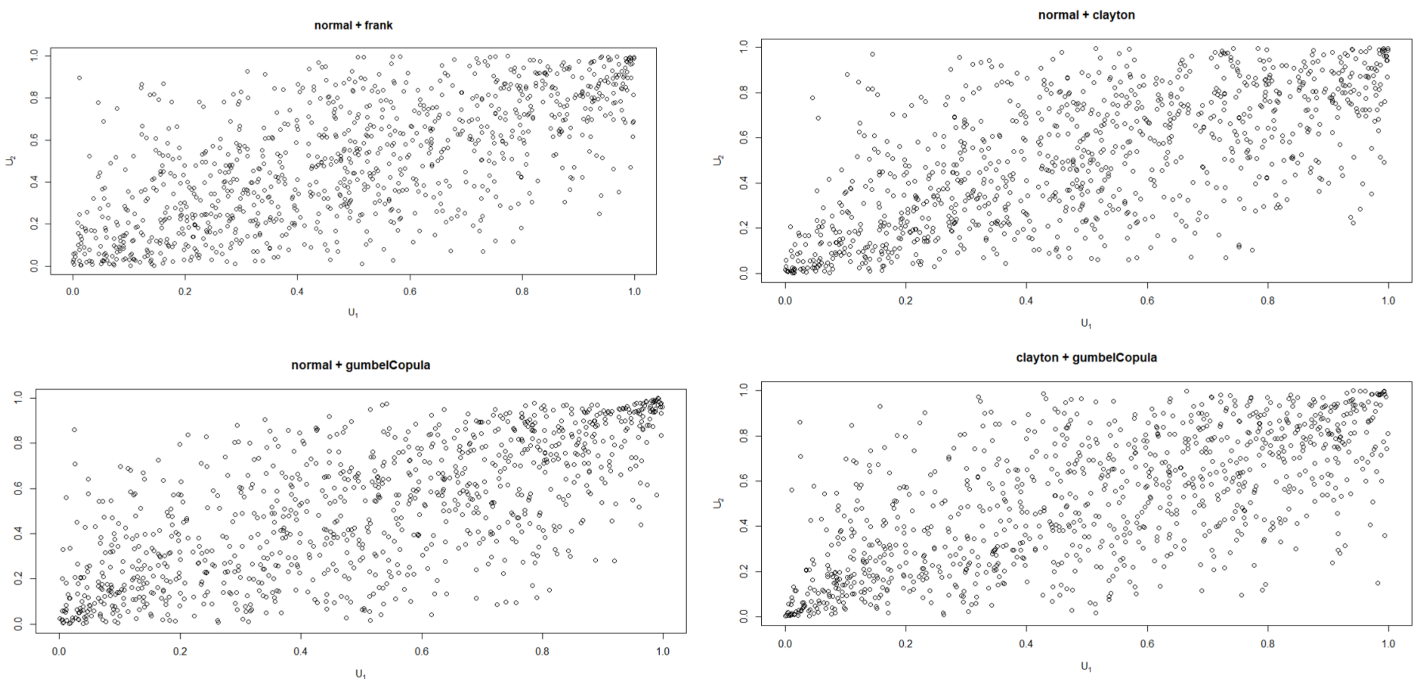
$$C(\mathbf{u};\boldsymbol{\theta}) = \sum_{k=1}^{s} \lambda_k\, C_k(\mathbf{u};\theta_k) = \sum_{k=1}^{s} \lambda_k\, C_k(F_1(x_1;\alpha_1), \ldots, F_p(x_p;\alpha_p);\theta_k),$$

where {C1(·),...,Cs(·)} is a set of basis copulas, which is a sequence of known copulas with unknown parameters $\{\theta_k\}$, $\{\lambda_k\}$ are the weights satisfying $0 \le \lambda_k \le 1$ and $\sum \lambda_k = 1$, and s is the number of candidate copulas. θ = (θ1,..., θs) is the vectorrepresents the degree of dependence, λ = (λ1,..., λs) is the vector of weights and α = (α1,..., αp) is the vector of marginal parameters for marginal distributions. Also, for any two random variables X and Y, copula dependent parameters θ can be transformed to the Kendall's τ proposed by Kendall and the Spearman's ρ proposed by Spearman

To understand the structure of copulas, including the tail dependencies we plotted single as well as mixed copula models in R



Plotting single Copula models in R



Plotting mixed copula models in R

2. Approach

In this section, we explain the selection and estimation procedures for a mixed copula model using **Penalized Likelihood method** with a **shrinkage parameter.**

Penalized likelihood methods augment the likelihood with a **penalty function,** which can be chosen to encode prior knowledge about the parameters or discourage undesirable estimates (e.g. small weights). Instead of doing a simple maximum likelihood estimation, we maximize the log-likelihood minus a penalty term, which depends on the model and generally increases with the number of parameters. By maximizing the likelihood function, the copulas with small weights are removed by a thresholding rule or shrinkage operator and parameters that remain are estimated. This approach is similar to the LASSO (L1 regularization) which enhances the prediction accuracy of the statistical model. In the study, marginal distributions are assumed to be known. Applying Sklar's theorem to (1) gives distribution function (2). Sklar's theorem states that any multivariate joint distribution can be written in terms of univariate marginal distributions.

$$\boxed{C(\mathbf{u};\boldsymbol{\theta}) = \sum_{k=1}^{s} \lambda_k C_k(\mathbf{u};\theta_k) = \sum_{k=1}^{s} \lambda_k C_k(F_1(x_1;\alpha_1), \ldots, F_p(x_p;\alpha_p);\theta_k),} \quad (1)$$

$$\downarrow$$

$$\boxed{H(x, y) = C(F(x), G(y)).}$$ **Sklar's theorem**

$$\downarrow$$

$$\boxed{F(\mathbf{x};\boldsymbol{\phi}) = \sum_{k=1}^{s} \lambda_k C_k(F_1(x_1;\alpha_1), \ldots, F_p(x_p;\alpha_p);\theta_k)} \quad (2)$$

We also get a **Joint density function** is given by (3) where $c_k(\mathbf{u}; \theta k) = \partial^p c_k(\mathbf{u}; \theta k)/\partial \mathbf{u}$ is the mixed partial derivative of the copula $C(\cdot)$ and we assume these copula densities $c_1(\cdot), \ldots, c_s(\cdot)$ exist

$$\boxed{f(\mathbf{x};\boldsymbol{\phi}) = \prod_{j=1}^{p} f_j(x_j;\alpha_j) \sum_{k=1}^{s} \lambda_k c_k(F_1(x_1;\alpha_1), \ldots, F_p(x_p;\alpha_p);\theta_k),} \quad (3)$$

When the sample is iid, the penalized log-likelihood takes the following form with a Lagrange multiplier term

$$\boxed{\begin{aligned} Q(\boldsymbol{\phi}) = {} & \sum_{t=1}^{T} \sum_{j=1}^{p} \ln f_j(X_{jt};\alpha_j) + \sum_{t=1}^{T} \ln \\ & \times \left[\sum_{k=1}^{s} \lambda_k c_k(F_1(X_{1t};\alpha_1), \ldots, F_p(X_{pt};\alpha_p);\theta_k) \right] \\ & - T \sum_{k=1}^{s} p_{\gamma_T}(\lambda_k) + \delta \left(\sum_{k=1}^{s} \lambda_k - 1 \right). \end{aligned}} \quad (4)$$

The first summand is the logarithm of the likelihood for marginal parameters and the second one is the logarithm of the likelihood for dependence parameters. The penalty function is $p_{\gamma T}$ and γT is the tuning parameter, which controls the complexity of the model. To avoid overfitting, the penalty function is applied only to the weight parameters $\{\lambda_k\}$ because some of them might be estimated as zero if they are insignificant in the model.

Estimation Procedure

For estimation, a **full maximum likelihood** approach is used to maximize the log of penalized likelihood function Q(ø) with respect to ø. We follow a **2 step** iterative algorithm for estimation procedure. First is estimating the marginal parameters (α) from the marginal likelihood. '**$\bar{\alpha}$**' is estimated by maximizing the likelihood corresponding to the marginal models not affected by the copula parameters θ as seen in Eq(5). The second step is optimizing the full likelihood with marginal parameters replaced by their estimators '$\bar{\alpha}$' from the first step in Eq(6)

$$\boxed{\sum_{t=1}^{T} \sum_{j=1}^{p} \ln f_j(x_{jt}; \alpha_j)} \quad (5) \qquad \boxed{Q(\bar{\boldsymbol{\alpha}}, \boldsymbol{\theta}) = L(\bar{\boldsymbol{\alpha}}, \boldsymbol{\theta}) - T \sum_{k=1}^{s} p_{\gamma_T}(\lambda_k) + \delta \left(\sum_{k=1}^{s} \lambda_k - 1 \right),} \quad (6)$$

where $L(\cdot)$ is the sum of the first two terms in Equation (4). Maximizing $Q(\bar{\boldsymbol{\alpha}}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ results in a two-step penalized likelihood estimator

$$\bar{\boldsymbol{\theta}} = (\bar{\boldsymbol{\theta}}^{\top}, \bar{\boldsymbol{\lambda}}^{\top})^{\top}$$

3. Simulation Study 1

For simulation studies, we look at 2 examples for Single Copula and a Mixed Copula Model to check if the proposed estimation works.

Single Copula Model

The first example consists of **single copula** functions and demonstrates the proposed estimation procedures work reasonably well in the finite sample case. The simulated example considers three sample sizes, T = **400, 700,** and **1000.** Observations are simulated from copula models Gaussian, Clayton, Gumbel, and Frank models by following the DGP and the penalized maximum likelihood estimators are computed. The Data-generating process (DGP) is a process that has the bivariate joint distribution and takes the form of a copula function and individual variables are normally distributed. The simulation is then repeated **1000** times

$$\boxed{(u_t, v_t) \sim \text{iid} \;\; C(u, v; \boldsymbol{\theta})} \quad \text{—-(7)}$$

Single Copula Model: Results

In Table 1, we see that as the sample size increases, the bias keeps decreasing for all the copula models reducing the sampling error.

Table 1. Biases and MSEs of the single copula parameter estimates in Example 1

T		Gaussian (λ_1, θ_1)	Clayton (λ_2, θ_2)	Gumbel (λ_3, θ_3)	Frank (λ_4, θ_4)
400	Bias	(−0.026, −0.004)	(0.000, 0.011)	(0.000, −0.518)	(−0.081, −0.104)
	MSE	(0.008, 0.001)	(0.001, 0.117)	(0.001, 0.604)	(0.026, 0.124)
700	Bias	(−0.009, −0.002)	(0.000, −0.003)	(0.001, −0.358)	(−0.020, −0.054)
	MSE	(0.002, 0.001)	(0.001, 0.079)	(0.001, 0.304)	(0.006, 0.024)
1000	Bias	(−0.010, −0.002)	(0.000, −0.002)	(0.000, −0.236)	(−0.015, −0.038)
	MSE	(0.004, 0.000)	(0.001, 0.034)	(0.001, 0.147)	(0.003, 0.014)

Table 2. Average estimated values of the unidentified copula parameters for single copula in Example 1 when $T = 1000$

Model	θ_{10}	θ_{20}	θ_{30}	θ_{40}	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$
Gaussian		5.000	6.100	6.900		4.842	6.063	6.922
		4.000	5.100	5.900		3.826	5.043	5.953
		6.000	7.100	7.900		5.846	7.076	7.908
Clayton	0.600		6.100	6.900	0.948		5.811	6.736
	0.400		5.100	5.900	0.745		4.700	5.933
	0.800		7.100	7.900	0.847		6.942	7.906
Gumbel	0.600	5.000		6.900	0.683	5.109		6.912
	0.400	4.000		5.900	0.349	4.198		5.928
	0.800	6.000		7.900	0.466	6.050		7.894
Frank	0.600	5.000	6.100		0.737	4.918	5.953	
	0.400	4.000	5.100		0.860	3.979	4.813	
	0.800	6.000	7.100		0.263	5.885	7.014	

Table 3. Percentage that the corresponding copula was chosen correctly (incorrectly) in Example 1

Model (×100)	T	Gaussian	Clayton	Gumbel	Frank
Gaussian	400	1.000	(0.000)	(0.000)	(0.092)
	700	1.000	(0.000)	(0.000)	(0.038)
	1000	1.000	(0.000)	(0.000)	(0.032)
Clayton	400	(0.010)	1.000	(0.000)	(0.000)
	700	(0.000)	1.000	(0.000)	(0.000)
	1000	(0.000)	1.000	(0.000)	(0.000)
Gumbel	400	(0.028)	(0.000)	1.000	(0.000)
	700	(0.000)	(0.000)	1.000	(0.000)
	1000	(0.000)	(0.000)	1.000	(0.000)
Frank	400	(0.289)	(0.000)	(0.000)	1.000
	700	(0.074)	(0.000)	(0.000)	1.000
	1000	(0.078)	(0.000)	(0.000)	1.000

[NOTE: θ*i*0 is initial values and is estimate for *i* = 1 to 4]

When the true coefficient parameters may be on the boundary of parameter space and dependence parameters are in an unidentified subset of parameter space, the limiting distribution for boundary parameters is abnormal and the likelihood estimator converges to an arbitrary value. In Table 2, we see that the estimated values are similar to the initial proposed parameter values which confirm that penalized likelihood estimator converges to an arbitrary value. In Table 3, we see that there is a **3.2%** chance that the **Frank model** will be selected when in reality the working model is **Gaussian**. The percentage of incorrectly choosing a model is very less. Hence the proposed estimation works.

Mixed Copula Model

All possible combinations of the **4 copulas** have the ability to capture most of the possible dependence structures. The Gaussian copula is widely used in financial fields, while the Frank copula exhibits no tail dependence like the Gaussian copula In contrast to the Gaussian and Frank copulas, the Clayton and Gumbel copulas exhibit asymmetric dependence The Clayton dependence is strong in the left tail, which implies that the Clayton copula is best suited for applications in which two variables are likely to decrease together. On the other hand, the Gumbel copula exhibits strong right tail dependence.

$$C(u, v; \boldsymbol{\theta}) = \lambda_1 C_{\mathrm{Ga}}(u, v; \theta_1) + \lambda_2 C_{\mathrm{Cl}}(u, v; \theta_2) + \lambda_3 C_{\mathrm{Gu}}(u, v; \theta_3) + \lambda_4 C_{\mathrm{Fr}}(u, v; \theta_4),$$

To demonstrate the proposed estimation procedures work reasonably well in the finite sample case for a mixed copula model, we will use three sample sizes, *T* = **400, 700,** and **1000**. Observations are simulated from **3** mixed copula models Gaussian+Clayton, Clayton+Gumbel, Gaussian+Frank, and the penalized maximum likelihood estimators were computed. The simulation was repeated **1000** times.

Mixed Copula Model: Results

Table 6. Biases and MSEs of the copula parameter estimates for mixed copula in Example 1

Model	T		(λ_1, θ_1)	(λ_2, θ_2)	(λ_3, θ_3)	(λ_4, θ_4)
Gaussian + Clayton	400	Bias	(−0.035, 0.015)	(−0.015, −0.045)		
		MSE	(0.019, 0.006)	(0.003, 0.107)		
	700	Bias	(−0.009, 0.009)	(−0.005, −0.050)		
		MSE	(0.004, 0.002)	(0.002, 0.064)		
	1000	Bias	(−0.002, 0.000)	(−0.003, 0.013)		
		MSE	(0.002, 0.001)	(0.001, 0.046)		
Clayton + Gumbel	400	Bias		(0.094, 0.541)	(−0.387, −0.852)	
		MSE		(0.050, 0.494)	(0.171, 3.367)	
	700	Bias		(−0.036, −0.253)	(−0.294, −0.717)	
		MSE		(0.013, 0.171)	(0.133, 1.733)	
	1000	Bias		(−0.001, −0.208)	(−0.241, −0.690)	
		MSE		(0.013, 0.144)	(0.108, 1.787)	
Gaussian + Frank	400	Bias	(0.083, −0.286)			(−0.085, 0.069)
		MSE	(0.054, 0.101)			(0.054, 0.061)
	700	Bias	(0.122, −0.272)			(−0.122, 0.144)
		MSE	(0.049, 0.076)			(0.049, 0.047)
	1000	Bias	(0.084, −0.272)			(−0.084, 0.130)
		MSE	(0.028, 0.076)			(0.028, 0.033)

Table 5. Percentage that the corresponding copula was chosen correctly (incorrectly) in Example 1

Model (×100)	T	Gaussian	Clayton	Gumbel	Frank
Gaussian + Clayton	400	0.952	1.000	(0.000)	(0.141)
	700	0.995	1.000	(0.000)	(0.062)
	1000	1.000	1.000	(0.009)	(0.010)
Clayton + Gumbel	400	(0.522)	1.000	0.528	(0.000)
	700	(0.378)	1.000	0.603	(0.000)
	1000	(0.203)	1.000	0.804	(0.000)
Gaussian + Frank	400	0.992	(0.000)	(0.011)	0.883
	700	1.000	(0.000)	(0.000)	0.904
	1000	1.000	(0.000)	(0.000)	0.982

We simulate three mixed copulas with only two components for the simulation study and just like single copula results we see that as the sample size increases, the sampling error decreases with a decrease in bias. In Table 5, we see that there is a **100%** probability to choose the correct mixed copula paired models. Since Gaussian and Frank models have similar dependence structures, there is a 100% chance to select Gaussian and Frank models. Thus the estimation procedure works well for the mixed model as well.

4. Simulation Study 2

The second simulation study is to check how the proposed selection method works if the working model is **misspecified** considering the following three misspecified models below when our candidate copulas are not included in all the actual components.

Misspecified Model

For the first model, data is generated from the single Gaussian copula (true model) but the working model is

$$C(u, v) = \lambda_1 C_{\mathrm{Cl}}(u, v) + \lambda_2 C_{\mathrm{Gu}}(u, v) + \lambda_3 C_{\mathrm{Fr}}(u, v).$$

For the second model, data are generated from the single Clayton copula (true model) but the working model is

$$\begin{aligned} C(u, v) &= \lambda_1 C_{\mathrm{Ga}}(u, v) + \lambda_2 C_{\mathrm{Sg}}(u, v) + \lambda_3 C_{\mathrm{Gu}}(u, v) \\ &\quad + \lambda_4 C_{\mathrm{Fr}}(u, v), \end{aligned}$$

For the third model, the data are generated from the mixed Clayton and Gumbel copula with equal weights but the working is

$$\begin{aligned} C(u, v) &= \lambda_1 C_{\mathrm{Ga}}(u, v) + \lambda_2 C_{\mathrm{Sg}} C(u, v) + \lambda_3 C_{\mathrm{Gu}}(u, v) \\ &\quad + \lambda_4 C_{\mathrm{Fr}}(u, v). \end{aligned}$$

The Clayton copula is absent from both Models II and III, while the survival Gumbel copula (Csg) which exhibits the same left tail dependence pattern as the Clayton copula is added to these two models.

Misspecified Model: Results

Table 8. Percentage that the corresponding copula function is selected and the estimated value of the related copula parameter θ in parentheses for Example 2

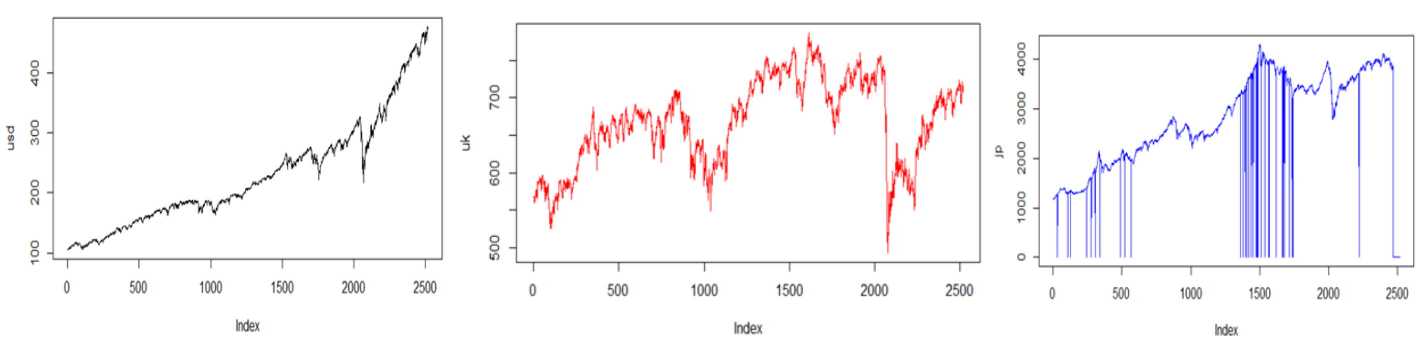
Model I	Clayton	Gumbel	Frank	
Percentage (Estimate of θ)	0.000	0.000	1.000 (7.107)	
Model II	Gaussian	SGumbel	Gumbel	Frank
Percentage (Estimate of θ)	0.000	0.996 (3.411)	0.000	0.152 (4.811)
Model III	Gaussian	SGumbel	Gumbel	Frank
Percentage (Estimate of θ)	0.000	0.994 (3.320)	0.884 (4.501)	0.000

In Model I, the Frank copula which is similar to the Gaussian copula is selected with **100%** for the missed Gaussian copula. In Model II**,** the working model includes the Gaussian, survival Gumbel, Gumbel, and Frank copulas without Clayton. The results show that the survival Gumbel copula exhibiting left tail dependence is chosen for all replications with **99.6%.** In Model III, the survival Gumbel is selected to replace the Clayton copula with **99.4%** selection rate, and the Gumbel copula appears with **88.4%**, The proposed estimation process can choose the best fit copula based on a combination of multiple copulas.

5. Real Example

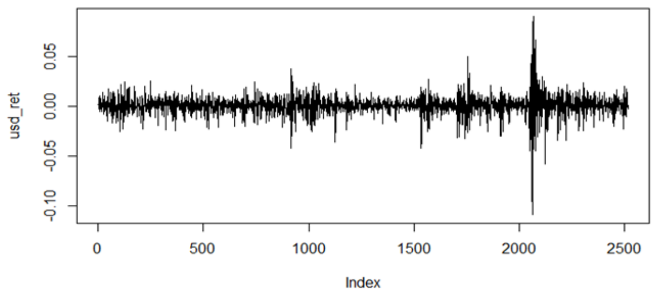
In this case study, we consider 3 international market indices **S&P 500** (US), **FTSE100** (UK) and **Nikkei** (JPY) market returns from 2012 till 2022.

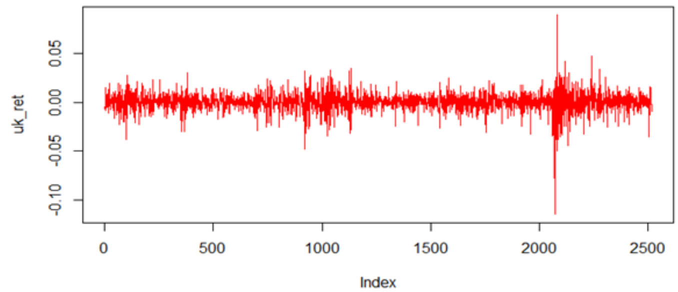
The main aim is to assess the comovement of returns among these 3 markets, compute the likelihood estimators of 3 pairs of markets and deduce the best-fit copula model with the highest likelihood. The first step is to check the **correlation** of markets using the linear correlation coefficient and Kendall's tau correlation. Second, we will be computing the copula parameters (initial and estimated parameters) of the international markets and, finally the log-likelihood of the copulas. The below graphs show the adjusted closing prices over a period of 10 years.



Adjusted Closing prices over 10 years

Results





The graphs above show the **returns** of the US and the UK markets over 10 years from 2012 to 2022.

	UK	JP
US	0.55(0.70)	0.38 (0.40)
UK		0.43 (0.37)
JP		

Chart 1: Correlation Analysis

Chart 1 shows the **linear correlation coefficient** and **Kendall's τs** in parentheses across 3 international markets. The results demonstrate that the US and the UK markets show a stronger correlation as compared to the US-JPY or UK-JPY markets.

Copulae are fitted on the basis of one or more specific parameters. According to the type of copula one or more initial values are proposed and then final parameters are estimated. Estimation of parameters is carried out by the method of pseudo-maximum likelihood method.

Copulas	Market Indices			
	parameters	US-UK	US-JP	UK-JP
Gaussian	ø1	0.51	0.51	0.51
	ø2	0.48	0.55	0.47
Clayton	ø1	0.85	0.85	0.85
	ø2	1.10	2.57	0.78
Gumbel	ø1	1.45	1.45	1.45
	ø2	1.33	2.40	1.41
Frank	ø1	3.30	3.30	3.30
	ø2	3.55	3.56	3.67

Chart 2 Estimates of Copula Parameters
(ø1= Initial Parameter) and ø2 = Estimated

Markets	Copula Models			
	Gaussian	Clayton	Gumbel	Frank
US-UK	714.4	1115.0	1074.5	1617.3
US-JP	329.2	500.8	159.1	364.5
UK-JP	306.9	264.3	267.1	376.2

Chart 3: Log-likelihood of Copula Models

Copula	Tail Dependency	US-UK	US-JP	UK-JP
Gaussian	Lower Bound	0.00	0.00	0.00
	Upper Bound	0.00	0.00	0.00
Clayton	Lower Bound	0.44	0.44	0.44
	Upper Bound	0.00	0.00	0.00
Gumbel	Lower Bound	0.00	0.00	0.00
	Upper Bound	0.39	0.38	0.34
Frank	Lower Bound	0.00	0.00	0.00
	Upper Bound	0.00	0.00	0.00

Chart 4: Tail Dependencies of copula models

In Chart 2, the results show that the estimated parameters approximate the initial proposed estimated values for all the four models and three market pairs. This proves the theory that the maximum likelihood estimator converges to an arbitrary value. In Chart 3, we see that the Frank model has the highest log-likelihood value of **1617** for the **US-UK** pair. Frank model has an asymmetric dependence structure which means variables are equally likely to be jointly low or jointly high.

For the **US-JPY** pair, we see that the Clayton model has the highest likelihood value of **500.8**. Clayton has an asymmetric structure which heavy left tail dependence which means if any 2 different markets crash together the degrees will be different. For the **UK-JPY** pair, Frank and Gaussian models have the highest log-likelihood value ~**376**. In general, the Frank and Gaussian model demonstrates to work better in financial markets as proposed by author as well.

6. Future Studies

The future studies are to generalize the proposed method and the related theory to the **time series** data and a more general setting like **finite mixture distribution**. It would also be very useful to investigate the case where the number of basis copulas s is large, say $s = s*$ going to infinity at a certain rate, theoretically and empirically. The computational implementation should also be addressed when the dimension of the copula function is high. Finally, the proposed method potentially can be applied to the analysis of multivariate financial data, such as multivariate **GARCH models** studied by Lee and Long (2009) and their extensions, and predictor selection in portfolio choice investigated by Cai, Peng, and Ren (2011),

References

1. Zongwu Cai & Xian Wang (2014), **“Selection of Mixed Copula Model via Penalized Likelihood”**
2. Anderson, T.W., and Darling, D.A. (1952), “**Asymptotic Theory of Certain Goodness-of-Fit Criteria Based on Stochastic Processes**,” Annals of Mathematical Statistics, 23, 193–212
3. Bouyé, E., Durrleman, V., Nikeghbali, A., Riboulet, G., and Roncalli, T. (2001), “**Copulas: An Open Field for Risk Management,**” Discussion Paper WP01-01, Paris, France: Groupe de Recherche Opérationnelle, Crédit-Lyonnais
4. Kendall tau calcualtions : https://rdrr.io/cran/CopulaCenR/man/tau_copula.html

Appendix

The R code is attached below along with the github link :

https://github.com/koushanidas/Selection-of-mixed-copula_/blob/main/Copula_estimation.txt

```
# Copula package
######################## Plotting the single copula models in ggplot
library(copula)
library(ggplot2)
library(grid)

install.packages("fCopulae")
library(fCopulae)
library(QRM)
set.seed(235)
# Build and plotting a Frank, a Gumbel and a Clayton single copula
frank <- frankCopula(dim = 2, param = 8)
gumbel <- gumbelCopula(dim = 3, param = 5.6)
clayton <- claytonCopula(dim = 4, param = 19)
# Select the copula
cp <- claytonCopula(param = c(3.4), dim = 2)
normal <- normalCopula(param = 0.7, dim = 2)
fr <- rCopula(2000, frank)
gu <- rCopula(2000, gumbel)
cl <- rCopula(2000, clayton)


# Plot the samples
p1 <- qplot(fr[,1], fr[,2], colour = fr[,1], main="Frank copula", xlab =
"u", ylab = "v")
p2 <- qplot(gu[,1], gu[,2], colour = gu[,1], main="Gumbel copula", xlab =
"u", ylab = "v")
p3 <- qplot(cl[,1], cl[,2], colour = cl[,1], main="Clayton copula", xlab =
"u", ylab = "v")

# Define grid layout to locate plots and print each graph^(1)
pushViewport(viewport(layout = grid.layout(1, 3)))
print(p1, vp = viewport(layout.pos.row = 1, layout.pos.col = 1))
print(p2, vp = viewport(layout.pos.row = 1, layout.pos.col = 2))
print(p3, vp = viewport(layout.pos.row = 1, layout.pos.col = 3))


###########This is mixed copula code - Clayton-Gumbel
cc <- claytonCopula(iTau(claytonCopula(), tau = 0.50)) # the first
component
gc <- gumbelCopula(iTau(gumbelCopula(),   tau = 0.50)) # the second
component
wts <- c(1/3, 2/3) # the corresponding weights
(mcg <- mixCopula(list(cc, gc), w = wts)) # the mixture copula

stopifnot(
  all.equal(   rho(mcg), wts[1] *    rho(cc) + wts[2] *    rho(gc)),
  all.equal(lambda(mcg), wts[1] * lambda(cc) + wts[2] * lambda(gc)))
lambda(mcg)
set.seed(127)
U <- rCopula(1000, copula = mcg) # sample from the mixture
wireframe2(mcg, FUN = dCopula, delta = 0.050) # density
contourplot2(mcg, FUN = pCopula) # copula
contourplot2(mcg, FUN = dCopula, cuts = 32, # density
```

```
             n.grid = 50, pretty = FALSE,
             col = adjustcolor(1, 1/3), alpha.regions = 3/4)
plot(U, xlab = quote(U[1]), ylab = quote(U[2])) # scatter plot


###########This is mixed copula code - normal-Gumbel
cc <- normalCopula(iTau(normalCopula(), tau = 0.50)) # the first component
gc <- gumbelCopula(iTau(gumbelCopula(),   tau = 0.50)) # the second
component
wts <- c(1/3, 2/3) # the corresponding weights
(mcg <- mixCopula(list(cc, gc), w = wts)) # the mixture copula

stopifnot(
  all.equal(   rho(mcg), wts[1] *    rho(cc) + wts[2] *    rho(gc)),
  all.equal(lambda(mcg), wts[1] * lambda(cc) + wts[2] * lambda(gc)))
lambda(mcg)
set.seed(127)
U <- rCopula(1000, copula = mcg) # sample from the mixture
wireframe2(mcg, FUN = dCopula, delta = 0.050) # density
contourplot2(mcg, FUN = pCopula) # copula
contourplot2(mcg, FUN = dCopula, cuts = 32, # density
             n.grid = 50, pretty = FALSE,
             col = adjustcolor(1, 1/3), alpha.regions = 3/4)
plot(U, xlab = quote(U[1]), ylab = quote(U[2])) # scatter plot
############################## Real data example

library(RColorBrewer)
cols <- brewer.pal(3, "BuGn")
pal <- colorRampPalette(cols)
exch<- read.csv(file="D:\\SPY.csv")
usd<-exch$Adj_Close_SPY
uk<-exch$Adj_Close_UK
jp<-exch$Adj_Close_JP
############ plotting closing prices of markets
plot(usd, type = "l", col = "black")
############ calculating returns of jpy
usd_ret <- diff(usd) / usd[- length(usd)]  # Calculate returns
uk_ret <- diff(uk) / uk[- length(uk)]  # Calculate returns
jp_ret <- diff(jp) / jp[- length(jp)]  # Calculate returns
########################Plotting the returns of markets
plot(usd_ret, type = "l", col = "black")
plot(jp_ret, type = "l", col = "blue")
############## Finding out correlation
cor(usd,uk)
cor(usd,jp)
cor(uk,jp)
################## Finding kendall's tau correlation
res<-cor.test(usd,uk, method="kendall")
res
res<-cor.test(uk,jp, method="kendall")
res
res<-cor.test(usd,jp, method="kendall")
res
```

```
###################################### estimating parameter and log
likelihood of copulas of international market data
####################USD-JP copula
val.ln <- cbind(exch[2],exch[3])
val.ln<- as.matrix(val.ln)
n<-nrow(val.ln)
summary(val.ln)
ro<-cor(val.ln)
ro<-ro[1,2]
Udata <- pobs(val.ln)
rotau<-Kendall(val.ln)
rotau<-rotau[1,2]
ParGum<-1/(1-rotau)
ParClay<-(2*rotau)/(1-rotau)
norm.cop <- normalCopula(ro, dim = 2, dispstr = "un")
###
EstimatedNormCop<-fitCopula(norm.cop,Udata, method="mpl")
EstimatedNormCop
logLik(EstimatedNormCop) #estimating log likelihood
AIC(EstimatedNormCop) ## estimatingthe AIC
BIC(EstimatedNormCop) #estimating BIC
##gaussian copula estimation
norm.cop <- normalCopula(0.50688, dim = 2, dispstr = "un")
TailDep<-tailIndex(norm.cop)
TailDep

###### CÃ³pula Gumbel #

gumb.cop0 <- gumbelCopula(ParGum, dim =2)
gumbCopEst<-fitCopula(gumb.cop0,Udata, method="mpl")
gumbCopEst
logLik(gumbCopEst)
AIC(gumbCopEst)
BIC(gumbCopEst)

# Dependencies on Gumbel copula
gumb.cop <- gumbelCopula(1.44915, dim =2)
TailDep<-tailIndex(gumb.cop)
TailDep

##############################
# Ajust clayton copula #

clay.cop0<- claytonCopula(param =ParClay, dim = 2)
ClayCopEst<-fitCopula(clay.cop0,Udata, method="mpl")
ClayCopEst
logLik(ClayCopEst)
AIC(ClayCopEst)
BIC(ClayCopEst)
#Dependencies on clayton copula
clay.cop<- claytonCopula(0.8450, dim = 2)
TailDep<-tailIndex(clay.cop)
TailDep
```

```
############################
# Ajuste frank copula #

frank.cop0<-frankCopula(param = NA_real_, dim = 2)
frank.cop0
FrankCopEst<-fitCopula(frank.cop0,Udata,method="mpl")
FrankCopEst
logLik(FrankCopEst)
AIC(FrankCopEst)
BIC(FrankCopEst)
#Dependencies on Frank copula
frank.cop<-frankCopula(3.3007, dim = 2)
TailDep<-tailIndex(frank.cop)
TailDep

######################### Estimate parameters for US-UK copula models
val.ln <- cbind(exch[2],exch[4])
val.ln<- as.matrix(val.ln)
n<-nrow(val.ln)
summary(val.ln)
ro<-cor(val.ln)
ro<-ro[1,2]
Udata <- pobs(val.ln)
rotau<-Kendall(val.ln)
rotau<-rotau[1,2]
ParGum<-1/(1-rotau)
ParClay<-(2*rotau)/(1-rotau)
############# Gaussian copula
norm.cop <- normalCopula(ro, dim = 2, dispstr = "un")
NormCopEst<-fitCopula(norm.cop,Udata, method="mpl")
NormCopEst
logLik(NormCopEst)
AIC(NormCopEst)
BIC(NormCopEst)
###### dependencies on Gaussian copula
norm.cop <- normalCopula(0.50688, dim = 2, dispstr = "un")
TailDep<-tailIndex(norm.cop)
TailDep

# Ajust - CÃ³pula Gumbel #
gumb.cop0 <- gumbelCopula(ParGum, dim =2)
gumbCopEst<-fitCopula(gumb.cop0,Udata, method="mpl")
gumbCopEst
logLik(gumbCopEst)
AIC(gumbCopEst)
BIC(gumbCopEst)
# Dependencies on Gumbel copula
gumb.cop <- gumbelCopula(1.44915, dim =2)
TailDep<-tailIndex(gumb.cop)
TailDep

##############################
# Ajust - CÃ³pula Clayton #
clay.cop0<- claytonCopula(param =ParClay, dim = 2)
```

```
ClayCopEst<-fitCopula(clay.cop0,Udata, method="mpl")
ClayCopEst
logLik(ClayCopEst)
AIC(ClayCopEst)
BIC(ClayCopEst)
#Dependencies of clayton copula
clay.cop<- claytonCopula(0.8450, dim = 2)
TailDep<-tailIndex(clay.cop)
TailDep

############################
# Ajust - CÃ³pula  Frank #
frank.cop0<-frankCopula(param = NA_real_, dim = 2)
FrankCopEst<-fitCopula(frank.cop0,Udata,method="mpl")
FrankCopEst
logLik(FrankCopEst)
AIC(FrankCopEst)
BIC(FrankCopEst)
#Dependencies of Frank copula
frank.cop<-frankCopula(3.3007, dim = 2)
TailDep<-tailIndex(frank.cop)
TailDep

######################################## Estimate parameters for UK-JP
copulas
val.ln <- cbind(exch[3],exch[4])
val.ln<- as.matrix(val.ln)
n<-nrow(val.ln)
summary(val.ln)
ro<-cor(val.ln)
ro<-ro[1,2]
Udata <- pobs(val.ln)
rotau<-Kendall(val.ln)
rotau<-rotau[1,2]
ParGum<-1/(1-rotau)
ParClay<-(2*rotau)/(1-rotau)
############## Normal copula
norm.cop <- normalCopula(ro, dim = 2, dispstr = "un")
NormCopEst<-fitCopula(norm.cop,Udata, method="mpl")
NormCopEst
logLik(NormCopEst)
AIC(NormCopEst)
BIC(NormCopEst)
######Dependencies of normal copula
norm.cop <- normalCopula(0.50688, dim = 2, dispstr = "un")
TailDep<-tailIndex(norm.cop)
TailDep


# Ajust - CÃ³pula Gumbel #
gumb.cop0 <- gumbelCopula(ParGum, dim =2)
gumbCopEst<-fitCopula(gumb.cop0,Udata, method="mpl")
gumbCopEst
logLik(gumbCopEst)
```

```
AIC(gumbCopEst)
BIC(gumbCopEst)

# Dependencies on copula gumbel
gumb.cop <- gumbelCopula(1.44915, dim =2)
TailDep<-tailIndex(gumb.cop)
TailDep

##############################
# Ajust CÃ³pula  Clayton #

clay.cop0<- claytonCopula(param =ParClay, dim = 2)
ClayCopEst<-fitCopula(clay.cop0,Udata, method="mpl")
ClayCopEst
logLik(ClayCopEst)
AIC(ClayCopEst)
BIC(ClayCopEst)

#Dependencies of clayton copula
clay.cop<- claytonCopula(0.8450, dim = 2)
TailDep<-tailIndex(clay.cop)
TailDep

############################
# Ajust Frank Copula
frank.cop0<-frankCopula(param = NA_real_, dim = 2)
FrankCopEst<-fitCopula(frank.cop0,Udata,method="mpl")
FrankCopEst
logLik(FrankCopEst)
AIC(FrankCopEst)
BIC(FrankCopEst)

#Dependencies of frank copula
frank.cop<-frankCopula(3.3007, dim = 2)
TailDep<-tailIndex(frank.cop)
TailDep
```