Maximum likelihood estimation of skew-*t* copulas with its applications to stock returns

Toshinao Yoshiba[*]

Bank of Japan, Chuo-ku, Tokyo 103-8660, Japan

The Institute of Statistical Mathematics, Tachikawa, Tokyo 190-8562, Japan

November 11, 2015

Abstract

The multivariate Student-*t* copula family is used in statistical finance and other areas when there is tail dependence in the data. It often is a good-fitting copula but can be improved on when there is tail asymmetry. Multivariate skew-*t* copula families can be considered when there is tail dependence and tail asymmetry, and we show how a fast numerical implementation for maximum likelihood estimation is possible. For the copula implicit in the multivariate skew-*t* distribution of Azzalini and Capitanio (2003), the fast implementation makes use of (i) monotone interpolation of the univariate marginal quantile function and (ii) a reparametrization of the correlation matrix. The same techniques apply to the generalized hyperbolic skew-*t* copula. Our numerical approach is tested with simulated data with realistic parameters. A real data example involves the daily returns of three stock indices: the Nikkei225, S&P500, and DAX. We investigate both unfiltered returns and GARCH/EGARCH filtered returns comparing with the Azzalini–Capitanio skew-*t*, generalized hyperbolic skew-*t*, non-skewed Student-*t*, skew-Normal, and Normal copulas.

Keywords: skew-*t* distribution; copula; maximum likelihood estimation; tail asymmetry; tail dependence; generalized hyperbolic distribution.

AMS Subject Classifications: 62E17; 62H10; 62H20; 65C60

1. Introduction

Correlations among risk factors matter in financial portfolio risk management. When the risk factors are specified using asset returns, risk managers need to consider tail dependence, that is, more dependence in the joint tails than with the multivariate normal distribution. In this situation, the Student-*t* copula is frequently used in financial portfolio risk management. As McNeil *et al.* (2015) indicate, a pair of daily stock returns is well described by a bivariate Student-*t* distribution in many cases. Accordingly, Aas *et al.* (2009), Nikoloulopoulos *et al.* (2012) and others statistically

[*] E-mail: toshinao.yoshiba@boj.or.jp

adopt the Student-*t* copula as a pair copula family within the vine copula to fit multivariate stock returns. However, the Student-*t* copula is restrictive because of its symmetric dependence for the joint upper and lower tails. The tail asymmetry of stock returns, such as more dependence in the joint lower tail compared with the joint upper tail, is referred in the literature as Ang and Chen (2002); Longin and Solnik (2001). Patton (2006) refers to the asymmetric dependence of foreign exchange rate and applies the Joe–Clayton copula, which has two parameters adjusting the upper and lower tail dependences and is a modification of the BB7 copula of Joe (1997, 2014). In terms of simple tail asymmetric copulas with vines, the BB1 copula of Joe (1997, 2014) is used in Nikoloulopoulos *et al.* (2012). With this background, the skew-*t* copula is a good alternative to the Student-*t* copula if a fast computation is possible. Then, the skew-*t* copula can capture the asymmetric dependence of risk factors. The skew-*t* copula is defined by a multivariate skew-*t* distribution and its marginal distributions. As indicated in Kotz and Nadarajah (2004), various types of multivariate skew-*t* distributions have been proposed, implying that there are also various types of skew-*t* copulas.

To our knowledge, three types of skew-*t* copulas have been proposed. The first was described in Demarta and McNeil (2005) and is based on a multivariate version of the generalized hyperbolic (GH) skew-*t* distribution proposed by Barndorff-Nielsen (1977). We call it the GH skew-*t* copula. The second type was constructed by Smith *et al.* (2012) and is implied in the multivariate skew-*t* distribution proposed by Sahu *et al.* (2003). The multivariate skew-*t* distribution is formed from hidden truncation. Hidden truncation has received considerable attention as a method of constructing a skew elliptical distribution (Arnold and Beaver, 2004), as indicated in Smith *et al.* (2012). Among the multivariate skew-*t* distributions with hidden truncation, the distribution of Azzalini and Capitanio (2003) is the most popular. The third type of skew-*t* copula was mentioned by Joe (2006) and is implicit in the multivariate skew-*t* distribution of Azzalini and Capitanio (2003); we call it the AC skew-*t* copula. These skew-*t* copula families have rarely been used in applications, possibly because of numerical difficulties.

In this study, we indicate two computational problems that arise when estimating the parameters of the AC skew-*t* copula by maximum likelihood estimation (MLE) and suggest approaches to simplify the numerical procedure. The first problem is that the log-likelihood function includes univariate skew-*t* quantile functions, which involve solving equations with integration and this makes calculating a log-likelihood time consuming. The second problem is that the extended correlation matrix should be positive semi-definite. We solve the first problem by applying a monotone interpolator to the distribution functions. We solve the second problem by re-parameterizing the Cholesky decomposed triangular matrix with trigonometric functions. This keeps the diagonal elements of the extended correlation matrix to the value one. After estimating the benchmark parameters from the daily returns of three stock indices (Nikkei225, DAX, and

S&P500), we test our numerical procedure for maximum likelihood using simulated trivariate and higher-dimensional data with the realistic parameters.

For empirical studies, we compare the fits of the AC skew-*t*, GH skew-*t*, Student-*t*, skew-Normal, and Normal copulas. The numerical implementations for GH skew-*t* and skew-Normal are similar to that for AC skew-*t*. We investigate both unfiltered daily return and GARCH or EGARCH filtered daily return of the three stock indices: the Nikkei225, S&P500, and DAX. We find that the AC skew-*t* copula well describes the dependence structures of these returns.

The remainder of the paper is organized as follows. Section 2 derives the log-likelihood function of the AC skew-*t* copula and describes the two computational problems that occur when estimating the parameters by MLE. Section 3 shows how to overcome these two problems. Section 4 confirms that the implementation of the MLE algorithm works well using trivariate simulated data for the benchmark parameters. Section 5 investigates both unfiltered daily return and GARCH or EGRACH filtered daily returns for three stock indices: the Nikkei225, S&P500, and DAX by comparing the AC skew-*t*, GH skew-*t*, Student-*t*, skew-Normal, Normal copulas. Section 6 concludes the paper.

2. Problems of MLE for AC skew-*t* copulas

This section introduces the AC skew-*t* copula which involves the univariate marginal distribution from the *d*-variate skew-*t* distribution of Azzalini and Capitanio (2003). After deriving log-likelihood function, we indicate the two problems that occur when using MLE to estimate the parameters. We also mention that the same problems are also applied to GH skew-*t* copula.

2.1. AC skew-*t* copula

The AC skew-*t* copula is implicit in the standard *d*-variate skew-*t* distribution with the location vector, $\boldsymbol{\xi}^\top = (\xi_1, \dots, \xi_d) = (0, \dots, 0)$ and the scale vector, $\boldsymbol{\sigma}^\top = (\sigma_1, \dots, \sigma_d) = (1, \dots, 1)$. The random vector $\boldsymbol{X}$ of this distribution has the following joint density function at $\boldsymbol{x}$:

$$g(\boldsymbol{x}; \Omega, \boldsymbol{\alpha}, \nu) = 2t_{d,\nu}(\boldsymbol{x}; \Omega) T_{1,\nu+d}\left(\boldsymbol{\alpha}^\top \boldsymbol{x} \sqrt{\frac{\nu + d}{\boldsymbol{x}^\top \Omega^{-1} \boldsymbol{x} + \nu}}\right), \tag{1}$$

where $t_{d,\nu}(\boldsymbol{x}; \Omega)$ is the *d*-variate Student-*t* density with the correlation matrix Ω and the degrees of freedom ν and $T_{1,\nu}(\cdot)$ is the univariate Student-*t* distribution function with degrees of freedom ν. The Student-*t* density $t_{d,\nu}(\boldsymbol{x}; \Omega)$ is specified as:

$$t_{d,\nu}(\boldsymbol{x}; \Omega) = \frac{\Gamma\big((\nu + d)/2\big)}{(\pi\nu)^{d/2}\Gamma(\nu/2)|\Omega|^{1/2}}\left[1 + \frac{\boldsymbol{x}^\top \Omega^{-1} \boldsymbol{x}}{\nu}\right]^{-\frac{\nu+d}{2}}.$$

The *d*-variate skew-*t* distribution is denoted as $\mathrm{St}_d(0, \Omega, \boldsymbol{\alpha}, \nu)$.

The random vector $\boldsymbol{X}$ of $\mathrm{S}t_d(0,\Omega,\boldsymbol{\alpha},\nu)$ is represented as $\boldsymbol{X}=V^{-1/2}\boldsymbol{Y}$ where $\boldsymbol{Y}$ has a d-variate skew-Normal distribution and V has a Gamma distribution $G(\nu/2,\nu/2)$. The skew-Normal random vector $\boldsymbol{Y}$ with the skewness vector $\boldsymbol{\delta}=(\delta_1,\delta_2,\dots,\delta_d)^\top$ is constructed as

$$\boldsymbol{Y}=\begin{cases}\boldsymbol{Z} & \text{if } Z_0\geq 0,\\ -\boldsymbol{Z} & \text{if } Z_0<0,\end{cases} \tag{2}$$

where (d+1)-dimensional vector $(Z_0,\boldsymbol{Z}^\top)^\top$ has the standard (d+1)-variate Normal distribution $N_{d+1}(\mathbf{0},R)$. The extended correlation matrix R is defined by

$$R=\begin{pmatrix}1 & \boldsymbol{\delta}^\top\\ \boldsymbol{\delta} & \Omega\end{pmatrix}, \tag{3}$$

using the original correlation matrix Ω and the skewness vector $\boldsymbol{\delta}$. The transformed skewness vector $\boldsymbol{\alpha}$ appeared in the joint density (1) is given as

$$\boldsymbol{\alpha}=\frac{\Omega^{-1}\boldsymbol{\delta}}{\sqrt{1-\boldsymbol{\delta}^\top\Omega^{-1}\boldsymbol{\delta}}}, \tag{4}$$

using the original skewness vector $\boldsymbol{\delta}$ and the original correlation matrix Ω.

As Joe (2006) indicates, the j-th marginal distribution of $\mathrm{S}t_d(0,\Omega,\boldsymbol{\alpha},\nu)$ is $\mathrm{S}t_1(0{,}1,\zeta_j,\nu)$, with density

$$g_j(x_j;\zeta_j,\nu)=2t_{1,\nu}(x_j)T_{1,\nu+1}\left(\zeta_j x_j\sqrt{\frac{\nu+1}{x_j^2+\nu}}\right), \tag{5}$$

where $t_{1,\nu}(\cdot)$ is the univariate Student-*t* density with degrees of freedom ν and ζ_j is defined as

$$\zeta_j=\frac{\delta_j}{\sqrt{1-\delta_j^2}}, \tag{6}$$

using the original skewness parameter δ_j. This result is different from that of Kollo and Pettere (2010) who erroneously mention that j-th marginal distribution of the d-variate skew-*t* distribution $St_d(0,\Omega,\boldsymbol{\alpha},\nu)$ is $St_1(0{,}1,\alpha_j,\nu)$.

Hence, applying Sklar's theorem, the AC skew-*t* copula is given by

$$C_{\mathrm{S}t}(u_1,\dots,u_d;\Omega,\boldsymbol{\delta},\nu)=\mathrm{S}t_d(St_1^{-1}(u_1;0{,}1,\zeta_1,\nu),\dots,St_1^{-1}(u_d;0{,}1,\zeta_d,\nu);0,\Omega,\boldsymbol{\alpha},\nu). \tag{7}$$

Henceforth, we refer to the (i,j) element of the original correlation matrix Ω as ρ_{ij}.

2.2. Properties of the multivariate AC skew-*t* distribution

As shown in equation (2), the d-variate AC skew-*t* distribution is formed from hidden truncation, as is the case for the skew-*t* distribution of Sahu *et al.* (2003). In addition, the d-variate AC skew-*t* distribution is based on a general class of multivariate skew-elliptical distributions proposed by Branco and Dey (2001) and is the most popular multivariate skew-*t* distribution. Similar to the multivariate skew-*t* distribution of Sahu *et al.* (2003), the covariance of the multivariate skew-*t* distribution of Azzalini and Capitanio (2003) is finite if the degree of freedom parameter ν is greater than 2. For the Fisher information matrix, Arellano-Valle (2010) indicates that the *d*-variate AC skew-*t* distribution is non-singular if $\boldsymbol{\alpha} = \mathbf{0}$ while the *d*-variate skew-Normal distribution is singular if $\boldsymbol{\alpha} = \mathbf{0}$. For the details of the distribution and other related distributions, see Azzalini (2014).

Asymmetric tail dependence is a characteristic of interest for the skew-*t* copula. If $\delta < 0$, then the lower tail has a stronger tail dependence than the upper tail. Fig. 1 confirms this by plotting the contours of the joint densities for bivariate Student-*t* and the AC skew-*t* copula with $\rho_{12} = \rho_{21} = \rho = 0.5, \delta_1 = \delta_2 = \delta$, using standard normal margins. The minimum skewness value is $\delta = -\sqrt{(1+\rho)/2} \cong -0.866$ to keep the positive semi-definiteness of the extended correlation matrix R. Padoan (2011) derives the lower tail dependence λ_L and the upper tail dependence λ_U as

$$\lambda_L = F_{EST}\left(-a_{2,1}; 0,1, \alpha_2\sqrt{1-\rho^2}, -\tau_1, \nu+1\right) + F_{EST}\left(-a_{1,2}; 0,1, \alpha_1\sqrt{1-\rho^2}, -\tau_2, \nu+1\right),$$

$$\lambda_U = 2 - F_{EST}\left(a_{2,1}; 0,1, \alpha_2\sqrt{1-\rho^2}, \tau_1, \nu+1\right) - F_{EST}\left(a_{1,2}; 0,1, \alpha_1\sqrt{1-\rho^2}, \tau_2, \nu+1\right),$$

where $F_{EST}(\cdot)$ is the univariate extended skew-*t* cumulative distribution function with

$$a_{2,1} = \left\{\left(\frac{T_{1,\nu+1}(-\zeta_2\sqrt{\nu+1})}{T_{1,\nu+1}(-\zeta_1\sqrt{\nu+1})}\right)^{\frac{1}{\nu}} - \rho\right\}\sqrt{\frac{\nu+1}{1-\rho^2}},\ a_{1,2} = \left\{\left(\frac{T_{1,\nu+1}(-\zeta_1\sqrt{\nu+1})}{T_{1,\nu+1}(-\zeta_2\sqrt{\nu+1})}\right)^{1/\nu} - \rho\right\}\sqrt{\frac{\nu+1}{1-\rho^2}},$$

$\tau_1 = \sqrt{\nu+1}(\alpha_1 + \rho\alpha_2)$, and $\tau_2 = \sqrt{\nu+1}(\alpha_2 + \rho\alpha_1)$. The standard univariate extended skew-*t* cumulative distribution function with location parameter 0 and scale parameter 1 is given as:

$$F_{EST}(x; 0,1, \alpha, \tau, \nu) = \int_{-\infty}^{x} t_{1,\nu}(z)\frac{T_{1,\nu+1}\left((\alpha z+\tau)\sqrt{\frac{\nu+1}{\nu+z^2}}\right)}{T_{1,\nu}(\tau/\sqrt{1+\alpha^2})}dz.$$

Fung and Seneta (2010) also show the equivalent formula for the lower tail dependence. Fig. 2 plots the lower and upper tail dependence of AC skew-*t* copula for $\rho = 0.5$. We can see that the lower tail dependence becomes much stronger as the skewness parameter δ decreases. The difference between lower and upper tail dependence becomes larger as the degree of freedom parameter ν becomes smaller.

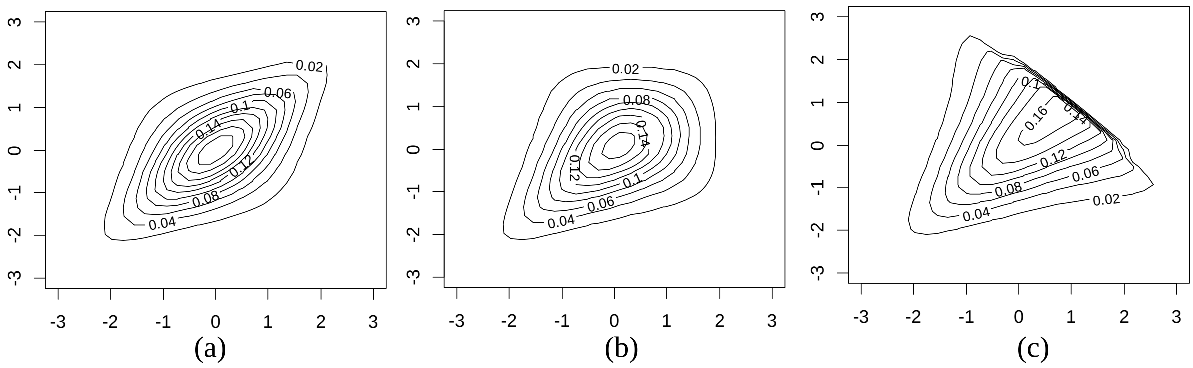

Fig. 1. Contour plot of bivariate distributions having standard normal margins and AC skew-*t* copula with $\delta_1 = \delta_2 = \delta$, $\rho = 0.5$ and $\nu = 3$: (a) $\delta = 0$, (b) $\delta = -0.7$, (c) $\delta = -0.866$.

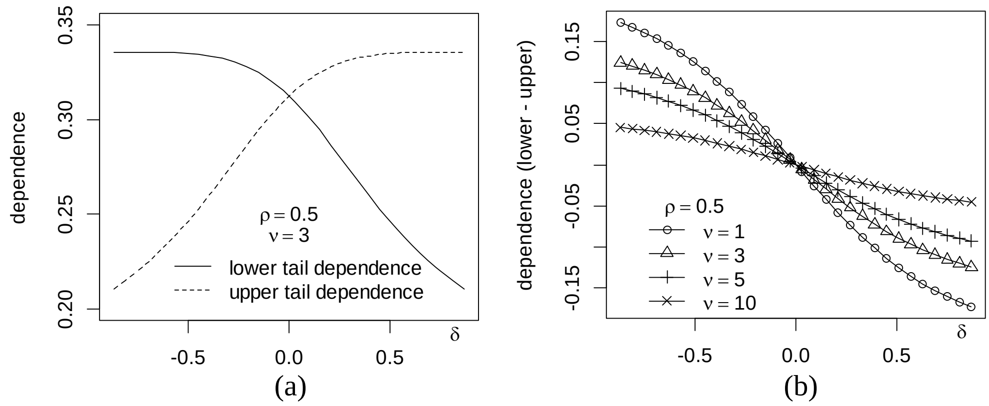

Fig. 2. Lower and upper tail dependence of AC skew-*t* copula having $\rho = 0.5$ with respect to (a) $\delta_1 = \delta_2 = \delta$ ($\nu = 3$) and (b) the difference between lower and upper tail dependence ($\nu = 1, 3, 5, 10$).

2.3. Log-likelihood function of AC skew-*t* copula

We assume that all univariate marginal distributions have been estimated and that data have been transformed to N observations u_i on $[0,1]^d$, for $i = 1, \ldots, N$, are given by the marginal distribution functions. The set of observations $\{u_1, \ldots, u_N\}$ is called a pseudo sample and can be obtained by applying the estimated univariate marginal distribution functions as probability integral transforms of the original sample. The estimation is called the two-stage estimation (see Joe, 2005 for its details and its asymptotic efficiency).

The log-likelihood function $l(\Omega, \boldsymbol{\delta}, \nu; \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N)$ is defined by

$$l(\Omega, \boldsymbol{\delta}, \nu; \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N) = \sum_{i=1}^{N} \ln c_{St}(\boldsymbol{u}_i; \Omega, \boldsymbol{\delta}, \nu), \tag{8}$$

using the density $c_{St}(\cdot)$ of the AC skew-*t* copula (7). The copula density $c_{St}(\cdot)$ in equation (8) is given as

$$c_{\mathrm{St}}(\boldsymbol{u}_i;\Omega,\boldsymbol{\delta},\nu)=\frac{\partial^d C_{\mathrm{St}}(u_{i1},\dots,u_{id};\Omega,\boldsymbol{\delta},\nu)}{\partial u_{i1}\cdots\partial u_{id}}=\frac{g(\boldsymbol{x}_i;\Omega,\boldsymbol{\alpha},\nu)}{\prod_{j=1}^{d} g_j(x_{ij};\zeta_j,\nu)},$$

where $\boldsymbol{x}_i=(x_{i1},\dots,x_{id})^\top$ is defined by

$$x_{ij}=\mathrm{St}_1^{-1}(u_{ij};0,1,\zeta_j,\nu). \tag{9}$$

Thus, the log-likelihood function is given as

$$l(\Omega,\boldsymbol{\delta},\nu;\boldsymbol{u}_1,\dots,\boldsymbol{u}_N)=\sum_{i=1}^{N}\left\{\ln g(\boldsymbol{x}_i;\Omega,\boldsymbol{\alpha},\nu)-\sum_{j=1}^{d}\ln g_j\big(x_{ij};\zeta_j,\nu\big)\right\}, \tag{10}$$

where $g(\boldsymbol{x}_i;\Omega,\boldsymbol{\alpha},\nu)$ is given by equation (1), $\boldsymbol{\alpha}$ is given by equation (4), and $g_j\big(x_{ij};\zeta_j,\nu\big)$ is given by equation (5).

2.4. Problems when estimating parameters using MLE

When maximizing the log-likelihood function $l(\Omega,\boldsymbol{\delta},\nu;\boldsymbol{u}_1,\dots,\boldsymbol{u}_N)$, we have two problems. First, the log-likelihood function given in equation (10) includes univariate skew-*t* quantile functions, as shown in equation (9). The quantile function should be applied $N\times d$ times, and this is a time-consuming calculation. The second problem is that the extended correlation matrix *R* in equation (3) should be positive semi-definite and numerical optimization with nonlinear constraints can be a complication.

2.5. GH skew-*t* copula

The two problems for AC skew-*t* copula in Section 2.4 are also relevant for the GH skew-*t* copula introduced by Demarta and McNeil (2005).

The random vector $\boldsymbol{X}$ of the based standard *d*-variate GH skew-*t* distribution has the following representation:

$$\boldsymbol{X}=\boldsymbol{\gamma}V^{-1}+V^{-1/2}\boldsymbol{Z}, \tag{11}$$

where V has a gamma distribution $G(\nu/2,\nu/2)$, and $\boldsymbol{Z}$ has a *d*-variate normal distribution $N_d(0,\Omega)$. Here, $\boldsymbol{\gamma}$ is a *d*-dimensional skewness parameter vector. This distribution is the multivariate version of the generalized hyperbolic skew-*t* distribution proposed by Barndorff-Nielsen (1977). If $\boldsymbol{\gamma}=\boldsymbol{0}$, then the implicit GH skew-*t* copula reduces to the Student-*t* copula.

As in the case of the AC skew-*t* copula, the log-likelihood function includes univariate GH skew-*t* quantile functions because the *j*-th marginal distribution of (11) is the univariate GH skew-*t* distribution with the skewness γ_j and the degree of freedom parameter ν.

We also have to keep positive semi-definiteness of the correlation matrix Ω of the random vector $\boldsymbol{Z}$ in equation (11) in the process of maximizing the log-likelihood function.

The covariance of the multivariate GH skew-*t* distribution with $\boldsymbol{\gamma}\neq\boldsymbol{0}$ is finite if the degree of freedom parameter ν is greater than 4, while that of Student-*t* distribution with $\boldsymbol{\gamma} = \mathbf{0}$ is finite if ν is greater than 2. The condition for the finite covariance of the distribution with skewness is stricter than that without skewness. Moreover, when ν goes to infinity, the GH skew-*t* distribution reduces to the Normal distribution. That is, the GH skew-*t* distribution does not nest the skew-Normal distribution. These are different limiting properties from multivariate AC skew-*t* distribution. For more information on this type, see also Aas and Haff (2006).

Christoffersen *et al.* (2012) applied this copula to weekly equity returns in both developed markets and emerging markets. They constrained the copula to have the same skewness parameter (i.e., $\gamma_j = \gamma$) for all *j*. They found the skewness parameter γ is significant in many cases.

3. Solutions to the MLE problems

This section describes how we overcome the two MLE problems discussed in the previous section.

3.1. A fast quantile function for the univariate skew-*t* distribution

An accurate quantile function for a univariate skew-*t* distribution is usually implemented in two steps. First, the distribution function is implemented as a numerical integration of the density. Second, the quantile is obtained using the iterative Newton method to equate the distribution function evaluated at the quantile to the given quantile probability level.

If we use an accurate quantile function, the calculation of 7,500 AC skew-*t* quantiles (N = 2,500 trivariate data values) with some fixed parameters with fractional ν takes more than eleven seconds using the statistical software R on an Intel i5-3230M (2.60GHz) processor running Microsoft Windows 7. This becomes time consuming when the dimension increases and many iterations for needed for numerical maximum likelihood.

One way to reduce the calculation time for quantiles of the univariate skew-*t* distribution is to use empirical quantiles with large random numbers (K). Christoffersen *et al*. (2012) use empirical quantiles with K = 100,000 to specify the GH skew-*t* copula because there is no closed-form quantile function for the univariate skew-*t* distribution. In financial applications, we usually calculate a lower-tail quantile (value at risk) for a portfolio. This quantile function needs to be accurate, especially in the tail. There is some debate on whether the empirical quantile with K = 100,000 random numbers is accurate enough for these applications.

A more efficient way to reduce the calculation time, while maintaining a degree of accuracy, is to use a monotone interpolator with m interpolating points (see Section 6.4 in Joe, 2014). Let $F(\cdot ; \zeta_j, \nu)$ be the distribution function of the univariate skew-*t* distribution of $St_1(0,1,\zeta_j,\nu)$. Note that the *j*-th variate of the pseudo sample has the values of $\{u_{1j}, \dots, u_{Nj}\}$. Let $p_{1j} = \min_{i=1,\dots,N} u_{ij}$,

$p_{mj} = \max_{i=1,\dots,N} u_{ij}$, and calculate $y_{1j} = F^{-1}(p_{1j}; \zeta_j, \nu)$, $y_{mj} = F^{-1}(p_{mj}; \zeta_j, \nu)$ using an accurate quantile function. Then, choose the interpolating points $y_{kj} = y_{1j} + (y_{mj} - y_{1j})\,(k-1)/(m-1)$ and calculate $p_{kj} = F(y_{kj}; \zeta_j, \nu)$, for $k = 2, \dots, m-1$. A monotone interpolator can be used with the table $\{(x_{1j}, p_{1j}), \dots, (x_{mj}, p_{m,j})\}$ to obtain quantile values $F^{-1}(p; \zeta_j, \nu)$ in $p \in [p_{1j}, p_{mj}]$. As a monotone interpolator, we use a piecewise cubic Hermite interpolating polynomial.

Table 1 compares the calculation time and accuracy of empirical quantiles and interpolating quantiles to those of accurate quantiles $\{F^{-1}(u_{11}; \zeta, \nu), \dots, F^{-1}(u_{Nd}; \zeta, \nu)\}$ with $d = 3$ and $\zeta_1 = \zeta_2 = \zeta_3 = \zeta$. The accurate quantiles are calculated by modified R codes based on the `sn` package (Azzalini, 2015). If we use empirical quantiles with K = 100,000 random numbers, the calculation is about 204 times faster than the accurate calculation for N = 2,500. On the other hand, the empirical quantiles have a mean absolute error (MAE) of 5.4×10^{-3} from the accurate quantiles. If we use empirical quantiles with K = 1,000,000 random numbers for N = 500, the calculation time is only three times faster than the accurate one. In this case, the empirical quantiles have an MAE of 1.6×10^{-3} from the accurate quantiles. If we use interpolating quantiles with m = 100 for N = 2,500, the quantiles have an MAE of 3.9×10^{-5} from the accurate quantiles. This calculation is about 438 times faster than the accurate calculation. In the case of m = 150, the quantiles have an MAE of 1.2×10^{-5} from the accurate quantiles. This calculation is about 309 times faster than the accurate calculation. Therefore, using a monotone interpolator is more accurate and faster than using empirical quantiles with large random numbers.

Table 1

Calculation time and accuracy of AC skew-*t* quantiles

Method	*K* or *m*	*N* = 2,500			*N* = 500		
		MAE	Time (sec.)	Speed	MAE	Time (sec.)	Speed
Accurate	–	4.3×10^{-7}	11.833	–	1.4×10^{-7}	2.391	–
Empirical	100,000	5.4×10^{-3}	0.058	203.7	5.1×10^{-3}	0.058	40.9
Empirical	1,000,000	1.5×10^{-3}	0.719	16.5	1.6×10^{-3}	0.715	3.3
Interpolate	100	3.9×10^{-5}	0.027	438.3	1.6×10^{-5}	0.027	88.5
Interpolate	150	1.2×10^{-5}	0.038	309.0	4.6×10^{-6}	0.035	68.5

Note: “Speed” of “empirical” and “interpolate” denote the ratio of the calculation time using the “accurate” quantiles to that of each method; the “MAE” denotes the mean absolute error from $\{F^{-1}(u_{11}; \zeta, \nu), \dots, F^{-1}(u_{N3}; \zeta, \nu)\}$. From Table 5, parameters ν and ζ are given as $\nu = 6.04$ and $\zeta = \delta/\sqrt{1-\delta^2} \cong -0.518$ using equation (6). “Time” and “MAE” are the means of 100 simulated samples.

If ν is positive integer, then a recursive iteration algorithm given by Theorem 1 and Remark 1 in Jamalizadeh *et al.* (2009) can be applied for the calculation of the distribution functions without numerical integrations. Table 2 compares the calculation time and accuracy of quantiles in the case of $\nu = 6$. The calculation of accurate quantiles for a positive integer ν is about fifty times faster than that for a fractional ν in the case of N = 2,500.

Table 2
Calculation time and accuracy of AC skew-*t* quantiles with integer ν ($\nu = 6$)

Method	K or m	N = 2,500			N = 500		
		MAE	Time (sec.)	Speed	MAE	Time (sec.)	Speed
Accurate	–	2.3×10^{-7}	0.217	–	8.2×10^{-8}	0.043	–
Empirical	100,000	5.5×10^{-3}	0.060	3.6	5.2×10^{-3}	0.058	0.7
Empirical	1,000,000	1.7×10^{-3}	0.718	0.3	1.5×10^{-3}	0.730	0.1
Interpolate	100	4.0×10^{-5}	0.003	65.7	1.4×10^{-5}	0.003	14.0
Interpolate	150	1.2×10^{-5}	0.005	40.1	4.2×10^{-6}	0.003	17.4

Note that the balance between calculation time and accuracy applies to all three types of skew-*t* copulas. As described earlier, Christoffersen *et al.* (2012) use empirical quantiles with K = 100,000 random numbers to specify the GH skew-*t* copula. Table 3 confirms the speed and accuracy of empirical quantiles. The GH skew-*t* empirical quantiles with K = 100,000 are twice faster than interpolating quantiles with m = 100 in the case of N = 2,500, however, they are less accurate with two decimal points in MAE. On the other hand, Smith *et al.* (2012) accurately calculate the marginal quantile for the multivariate skew-*t* distribution of Sahu *et al.* (2003) using the Newton method, which applies numerical integration to the distribution function. We can confirm the speed and accuracy of the accurate quantiles by Table 1 because the univariate marginal distribution of the multivariate skew-*t* by Sahu *et al.* (2003) is the same as that by Azzalini and Capitanio (2003).

Table 3
Calculation time and accuracy of GH skew-*t* quantiles

Method	K or m	N = 2,500			N = 500		
		MAE	Time (sec.)	Speed	MAE	Time (sec.)	Speed
Accurate	–	1.7×10^{-7}	1.018	–	5.4×10^{-8}	0.312	–
Empirical	100,000	5.9×10^{-3}	0.048	21.2	5.7×10^{-3}	0.048	6.6
Empirical	1,000,000	1.9×10^{-3}	0.641	1.6	1.8×10^{-3}	0.614	0.5
Interpolate	100	4.6×10^{-5}	0.075	13.5	1.9×10^{-5}	0.065	4.8
Interpolate	150	1.4×10^{-5}	0.090	11.3	5.4×10^{-6}	0.079	4.0

Note: From Table 6, parameters ν and γ are given as $\nu = 6.20$ and $\gamma = -0.17$. Time and MAE are the means of 100 simulated samples as Table 1.

3.2. Positive semi-definiteness for the extended correlation matrix

Since the extended correlation matrix R of the AC skew-*t* copula is symmetric and positive semi-definite, the matrix R can be Cholesky decomposed as

$$R = LL^{\top},$$

where L is a lower triangular matrix, given as

$$L = \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{d+1,1} & l_{d+1,2} & \cdots & l_{d+1,d+1} \end{pmatrix}.$$

Furthermore, the diagonal elements are all one and the non-diagonal elements are in $(-1,1)$, because the matrix R is a correlation matrix. Thus the elements of the lower triangular matrix L can be represented by $\theta_{i,j} \in [0,\pi)$ for $j = 1, \dots, i-2$ and $\theta_{i,i-1} \in [0,2\pi)$ for $i = 2, \dots, d+1$ as

$$\begin{aligned} l_{ii} &= \begin{cases} 1 & \text{for } i = 1, \\ \left(\prod_{k=1}^{i-2} \sin\theta_{i,k}\right) \sin\theta_{i,i-1} & \text{for } i = 2, \dots, d+1, \end{cases} \\ l_{ij} &= \left(\prod_{k=1}^{j-1} \sin\theta_{i,k}\right) \cos\theta_{i,j} \quad \text{for } j < i, \text{ and } i = 2, \dots, d+1, \end{aligned} \tag{12}$$

where $\left(\prod_{k=1}^{0} \sin\theta_{i,k}\right) \equiv 1$. We can confirm that the diagonal elements of the matrix R have the value one as follows:

$$\begin{aligned} (R)_{ii} = \sum_{j=1}^{i} l_{ij}^2 &= \sum_{j=1}^{i-2} l_{ij}^2 + \sin^2\theta_{i1} \cdots \sin^2\theta_{i,i-2} \left(\cos^2\theta_{i,i-1} + \sin^2\theta_{i,i-1}\right) \\ &= \sum_{j=1}^{i-3} l_{ij}^2 + \sin^2\theta_{i1} \cdots \sin^2\theta_{i,i-3} \left(\cos^2\theta_{i,i-2} + \sin^2\theta_{i,i-2}\right) \\ &= \cos^2\theta_{i1} + \sin^2\theta_{i1} = 1. \end{aligned}$$

It is clear that the absolute values of the non-diagonal elements in the matrix R do not exceed 1 because of the positive semi-definiteness.

The representation (12) corresponds to $\cos\theta_{i,j} = \rho_{i,j;1:(j-1)}$ for $j < i$ and $i = 2, \dots, d+1$, where $\rho_{i,j:1\cdots(j-1)}$ is the partial correlation between *i*-th variate and *j*-th one with 1st to $(j-1)$th variates held constant. See Lewandowski *et al.* (2009) and Joe (2014). Luo and Shevchenko (2010) use the Cholesky decomposed matrix to represent the correlation parameters of the grouped *t* copula with the constraint that $l_{ii}^2 = 1 - \sum_{j=1}^{i-1} l_{ij}^2$.

Now, the extended correlation matrix R is re-parameterized as θ_{ij} for $j = 1, \dots, i-1$, and $i = 2, \dots, d+1$ using equation (12). The number of parameters for θ_{ij} is $(d+1)d/2$ for $d \geq 2$.

This re-parameterization can be also applied to the correlation matrix Ω of the GH skew-*t* copula.

4. Implementation

Based on the solution to the MLE problems described in the previous section, we now test our solution after estimating benchmark parameters. See Appendix A–C for the implementation of MLE using the statistical software R. Here we assume equi-skewness setting: $\delta_1 = \delta_2 = \delta_3 = \delta$ for AC skew-*t*, $\gamma_1 = \gamma_2 = \gamma_3 = \gamma$ for GH skew-*t*. With this assumption, we are adding one extra parameter for joint tail asymmetry; it is reasonable when the log-likelihood is relatively flat over several skew parameters and a common joint tail skewness direction is suggested from the bivariate plots. We have also fitted the data without the equi-skewness assumption and this did not lead to an improvement in the Akaike information criterion.

4.1. Benchmark parameters

The MLE for the AC skew-*t* copula can be obtained by maximizing the log-likelihood function in equation (10) using piecewise cubic Hermite interpolating polynomials. The internal parameters are re-parameterized as θ_{ij} for $j = 1, \dots, i-1$ and $i = 2, \dots, d+1$, as shown in equation (12).

Before conducting the simulation, we estimate the trivariate ($d = 3$) skew-*t* copula for Nikkei225, S&P500, and DAX daily return data $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ from April 1, 2010 to March 31, 2015 using pseudo observations $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_N\}$. The pseudo observations $\{u_{1j}, \dots, u_{Nj}\}$ ($j = 1,2,3$) are constructed by a version of the empirical distribution function of *j*-th variate as:

$$u_{ij} = \frac{1}{N+1} \sum_{k=1}^{N} 1_{\{x_{kj} \le x_{ij}\}}\ , \qquad i = 1, \dots, N;\ \ j = 1,2,3. \tag{13}$$

Owing to trading time differences, the correlation between the Nikkei225 and the other two is weak. Therefore, we use one-day lagged data for the Nikkei225. Regarding the construction of pseudo observations by equation (13), see Section 7.5.2 on McNeil *et al.* (2015), for example. We estimate parameters as Table 4 on the setting of equi-skewness. We adopt the estimated parameters with rounded off to the second decimal place, as the benchmark parameters.

Table 4

Estimated benchmark parameters of the trivariate AC and GH skew-*t* copulas

	ρ_{21}	ρ_{31}	ρ_{32}	δ, γ	ν
AC skew-*t*	0.559	0.461	0.699	–0.464	6.039
GH skew-*t*	0.483	0.364	0.651	–0.168	6.201

4.2. Confirmation by simulation

We iterate the maximum likelihood estimation with 100 simulated pseudo samples of trivariate data, with N = 500 and 2,500. Each pseudo sample $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_N\}$ is generated from a simulated original sample $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ as equation (13). For comparison, we also calculate the MLE of the trivariate skew-*t* distribution for the 100 simulated samples in a similar way, assuming location parameters $\xi_j = 0$ and scale parameters $\sigma_j = 1$, for $j = 1, \dots, 3$. Table 5 summarizes the results of RMSE (root mean squared error) of estimated parameters and calculation time for AC skew-*t* copula. Table 6 summarizes those for GH skew-*t* copula. Similar to Table 1, these calculations were done on an Intel i5-3230M (2.60GHz) processor running Microsoft Windows 7. The calculation time is only for the parameter estimation without the standard error, although the sample code in Appendix A includes the procedure for the calculation of the standard errors.

Table 5

Root mean squared errors of AC skew-*t* estimated parameters and computational time

	N	RMSE ρ_{21}	ρ_{31}	ρ_{32}	δ	ν	Mean Time (sec.)
Copula	500	0.05	0.06	0.03	0.24	1.77	9.6
	2,500	0.04	0.05	0.03	0.15	1.42	12.0
Distribution	500	0.03	0.03	0.02	0.04	1.45	0.45
	2,500	0.02	0.02	0.02	0.03	1.42	0.78

Table 6

Root mean squared errors of GH skew-*t* estimated parameters and computational time

	N	RMSE ρ_{21}	ρ_{31}	ρ_{32}	γ	ν	Mean Time (sec.)
Copula	500	0.04	0.04	0.03	0.17	2.89	24.9
	2,500	0.02	0.02	0.01	0.08	2.33	29.9
Distribution	500	0.04	0.05	0.03	0.04	2.03	0.34
	2,500	0.03	0.03	0.02	0.03	1.65	0.91

Both in Table 5 and Table 6, RMSE of parameters decrease as the sample size N increases. We also see that RMSE of the skewness parameters δ in Table 5 and γ in Table 6 are larger than those of the correlation parameters ρ_{ij} in the copula parameters estimation, especially for N = 500. The skewness parameters δ and γ have an effect on both the marginal distributions and the copula. The pseudo sample does not include information on the effect on the marginal distributions. That is one of the reasons that the RMSEs of the skewness parameters are large.

5. Empirical results for three stock returns

We apply the proposed method to estimate the trivariate AC and GH skew-*t* copula for Nikkei225, S&P500, and DAX daily return data. We investigate whether the tail dependence, the parameter ν, is significant and the asymmetric dependence, the parameter δ or γ, is significant both for unfiltered returns and standardized residuals of GARCH(1,1) or EGARCH(1,1) by comparing estimated parameters with those of Student-*t*, skew-Normal, and Normal copulas. The pseudo sample $\{u_{1j}, \dots, u_{Nj}\}$ $(j = 1,2,3)$ is obtained as equation (13). For the same reason to estimate benchmark parameters, we use one-day lagged data for the Nikkei225.

5.1. Estimated copulas for unfiltered returns

Table 7 has the estimated parameters of the AC skew-*t*, GH Student-*t*, skew-Normal, and Normal copulas for unfiltered five-year daily return from April 1, 2010 to March 31, 2015 ($N = 1{,}188$). Table 8 is that for unfiltered ten-year daily return from April 1, 2005 to March 31, 2015 ($N = 2{,}367$). In both Table 7 and Table 8, the AC skew-*t* copula attains the lowest AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) among the five copula families and is selected both by the AIC and BIC. To ensure the significance of the skewness parameter, we apply likelihood ratio test with the null hypothesis $\delta = 0$ using the test statistic of the double of the difference between log-likelihood of the AC skew-*t* copula and that of the Student-*t* copula follows $\chi^2(1)$ under the null hypothesis. In Table 7, the test statistic is about 7.3, the p-value is 0.68%. In Table 8, the test statistic is about 15.9, the p-value is 0.01%. In both cases, the skewness parameter δ is significant at the 1% level.

Table 7

Estimated parameters for daily return from April 1, 2010 to March 31, 2015

	AC Skew-*t*	GH Skew-*t*	Student-*t*	Skew -Normal	Normal
ρ_{21}	0.559	0.483	0.492	0.642	0.488
ρ_{31}	0.461	0.364	0.376	0.558	0.369
ρ_{32}	0.699	0.651	0.655	0.751	0.644
δ, γ	–0.464	–0.168		–0.683	
ν	6.039	6.201	6.094		
log-likelihood	526.9	526.7	523.2	482.4	477.6
AIC	**–1043.8**	–1043.5	–1038.5	–956.7	–949.1
BIC	**–1018.4**	–1018.1	–1018.2	–936.4	–933.9

Table 8

Estimated parameters for daily return from April 1, 2005 to March 31, 2015

	AC Skew-*t*	GH Skew-*t*	Student-*t*	Skew -Normal	Normal
ρ_{21}	0.527	0.488	0.493	0.659	0.494
ρ_{31}	0.429	0.381	0.385	0.584	0.383
ρ_{32}	0.650	0.621	0.624	0.737	0.611
δ, γ	–0.357	–0.067		–0.709	
ν	3.536	3.661	3.623		
log-likelihood	1108.8	1105.7	1100.8	908.2	893.5
AIC	**–2207.6**	–2201.5	–2193.6	–1808.4	–1781.0
BIC	**–2178.7**	–2172.6	–2170.6	–1785.3	–1763.7

5.2. Estimated copulas for standardized residuals

In the GARCH type approach, daily returns of each stock are modeled as $r_t = \mu + \sigma_t z_t$. Here, z_t are called as standardized residuals. In GARCH(1,1), the local volatility σ_t is modeled as

$$\sigma_t^2 = \omega + \sigma_{t-1}^2(\beta + \alpha z_{t-1}^2). \tag{14}$$

In EGARCH(1,1), the local volatility σ_t is modelled as

$$\ln(\sigma_t^2) = \omega + \beta \ln(\sigma_{t-1}^2) + \alpha z_{t-1} + \gamma(|z_{t-1}| - E|z_{t-1}|). \tag{15}$$

EGARCH captures the asymmetric movement of volatility by the parameter α.

To save the space, we focus on the ten-year observation period from April 1, 2005 to March 31, 2015 (N = 2,367). Table 9 is the result of estimated parameters of AC skew-*t*, GH skew-*t*, Student-*t*, skew-Normal, and Normal copulas for the standardized residuals z_t in equation (14). Table 10 has the estimated parameters of the five copulas for the standardized residuals z_t in equation (15). For each margin, z_t is assumed to follow the univariate standard Normal distribution. The estimation is done by using `rugarch` package (Ghalanos, 2014).

Both in Table 9 for the result of GARCH(1,1) and in Table 10 for the result of EGARCH(1,1), the AC skew-*t* copula attains the lowest AIC and BIC among the five copula families. The test statistic of the likelihood ratio test with the null hypothesis $\delta = 0$ is about 9.7 in Table 9 and 12.0 in Table 10, the p-value is 0.18% and 0.05%, respectively. The skewness parameter δ is significant with the 1% level in both cases. The skewness parameters for both the AC and GH skew-*t* copulas are negative and this indicates that the tail skewness is in the direction of the joint lower tail, that is, more tail dependence in the joint lower tail than the joint upper tail. This matches what can be seen from the bivariate scatterplots of the filtered residuals.

Table 9

Estimated parameters for daily standardized residuals of GARCH(1,1) from April 1, 2005 to March 31, 2015

	AC Skew-*t*	GH Skew-*t*	Student-*t*	Skew -Normal	Normal
ρ_{21}	0.568	0.483	0.487	0.624	0.484
ρ_{31}	0.477	0.376	0.382	0.542	0.372
ρ_{32}	0.677	0.613	0.618	0.719	0.615
δ, γ	−0.511	−0.193		−0.651	
ν	9.297	9.520	9.434		
log-likelihood	926.4	924.8	921.6	891.7	885.6
AIC	**−1842.9**	−1839.5	−1835.2	−1775.4	−1765.2
BIC	**−1814.0**	−1810.7	−1812.1	−1752.3	−1747.9

Table 10
Estimated parameters for daily standardized residuals of EGARCH(1,1) from April 1, 2005 to March 31, 2015

	AC Skew-*t*	GH Skew-*t*	Student-*t*	Skew -Normal	Normal
ρ_{21}	0.589	0.472	0.480	0.627	0.478
ρ_{31}	0.502	0.364	0.373	0.549	0.368
ρ_{32}	0.695	0.613	0.621	0.727	0.618
δ, γ	–0.576	–0.281		–0.667	
ν	11.732	11.520	12.553		
log-likelihood	913.2	911.2	907.2	890.5	883.0
AIC	**–1816.4**	–1812.3	–1806.3	–1773.1	–1759.9
BIC	**–1787.5**	–1783.5	–1783.3	–1750.0	–1742.6

6. Conclusions

We have indicated two problems when using MLE to estimate parameters of skew-*t* copulas. First, practical MLE requires fast and accurate quantile calculations for a univariate skew-*t* distribution. Second, the correlation matrix should be kept positive semi-definite during the iterations for numerical optimization.

We have provided a solution to both problems and implemented in code for arbitrary dimensions (see Appendix A–C). We then confirm that the solution works by simulating a trivariate pseudo sample and estimating the parameter of the AC skew-*t* copula. We also show the solution can be applied to the GH skew-*t* copula. It is important to have a fast numerical approach for estimation of skew-*t* copulas. For finance data, one can sometimes see from bivariate scatterplots that there is joint tail asymmetry and tail dependence, and this suggests the use of models that extend the Student-*t* copula.

As the empirical studies for unfiltered and filtered daily return for the three stock indices; S&P500, DAX, and Nikkei225, we show the AC skew-*t* copula is effective in many cases compared with GH skew-*t*, Student-*t*, skew-Normal, and Normal copulas.

Acknowledgements

The author deeply appreciates Harry Joe who gave a lot of substantial suggestions including the idea of using a monotone interpolator to calculate quantiles quickly. The author is also grateful to Adelchi Azzalini, Hironori Fujisawa, Tsunehiro Ishihara, Shogo Kato, Satoshi Kuriki, Alexander J. McNeil, Gareth Peters, Pavel V. Shevchenko, Hideatsu Tsukahara, Toshiaki Watanabe, and Satoshi Yamashita for their helpful comments. The views expressed here are those of the author and do not necessarily reflect the official views of the Bank of Japan.

References

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009) "Pair-copula constructions of multiple dependence," *Insurance: Mathematics and Economics*, **44**(2), 182–198.

Aas, K. and I. H. Haff (2006) "The generalized hyperbolic skew Student's *t*-distribution," *Journal of Financial Econometrics*, **4**(2), 275–309.

Ang, A. and J. Chen (2002) "Asymmetric correlations of equity portfolios," *Journal of Financial Economics*, **63**(3), 443–494.

Arellano-Valle, R. B. (2010), "On the information matrix of the multivariate skew-*t* model," *Metron*, **68**(3), 371–386.

Arnold, B. C. and R. J. Beaver (2004) "Elliptical models subject to hidden truncation or selective sampling," in M. G. Genton ed. *Skew-Elliptical Distributions and Their Applications: A Journey Beyond Normality*, Chap.6, 101–112: Chapman & Hall/CRC.

Azzalini, A. (2014) *The Skew-Normal and Related Families*: Cambridge University Press.

Azzalini, A. (2015) *The R sn package: The skew-normal and skew-t distributions (version 1.2-0)*, Università di Padova, Italia.

Azzalini, A. and A. Capitanio (2003) "Distributions generated by perturbation of symmetry with emphasis on a multivariate skew *t*-distribution," *Journal of the Royal Statistical Society Series B*, **65**(2), 367–389.

Azzalini, A. and A. Dalla Valle (1996) "The multivariate skew-normal distribution," *Biometrika*, **83**(4), 715–726.

Barndorff-Nielsen, O. E. (1977) "Exponentially decreasing distributions for the logarithm of particle size," *Proceedings of the Royal Society of London Series A*, **353**(1674), 401–419.

Branco, M. D. and D. K. Dey (2001) "A general class of multivariate skew-elliptical distributions," *Journal of Multivariate Analysis*, **79**(1), 99–113.

Christoffersen, P., V. Errunza, K. Jacobs, and H. Langlois (2012) "Is the potential for international diversification disappearing? A dynamic copula approach," *Review of Financial Studies*, **25**(12), 3711–3751.

Demarta, S. and A. J. McNeil (2005) "The *t* copula and related copulas," *International Statistical Review*, **73**(1), 111–129.

Fung, T. and E. Seneta (2010) "Tail dependence for two skew *t* distributions," *Statistics & Probability Letters*, **80**(9–10), 784–791.

Ghalanos, A. (2014) *rugarch: Univariate GARCH models*, R package version 1.3-4.

Jamalizadeh, A., M. Khosravi, and N. Balakrishnan (2009) "Recurrence relations for distributions of a skew-*t* and a linear combination of order statistics from a bivariate-*t*," *Computational Statistics & Data Analysis*, **53**(4), 847–852.

Joe, H. and J. J. Xu (1996) "The estimation method of inference functions for margins for multivariate models," Technical Report No.166, Department of Statistics, University of British Columbia.

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

Joe, H. (2005) "Asymptotic efficiency of the two-stage estimation method for copula-based models," *Journal of Multivariate Analysis*, **94**, 401–419.

Joe, H. (2006) "Discussion of 'Copulas: tales and facts,' by Thomas Mikosch," *Extremes*, **9**(1), 37–41.

Joe, H. (2014), *Dependence Modeling with Copulas*: Chapman & Hall/CRC.

Kollo, T. and G. Pettere (2010) "Parameter estimation and application of the multivariate skew *t*-copula," in P. Jaworski, F. Durante, W. K. Hardle, and T. Rychlik eds. *Copula Theory and Its Applications*, Chap.15, 289–298: Springer.

Kotz, S. and S. Nadarajah (2004) *Multivariate t Distributions and Their Applications*: Cambridge University Press.

Lewandowski, D., D. Kurowicka, and H. Joe (2009) "Generating random correlation matrices based on vines and extended onion method," *Journal of Multivariate Analysis*, **100**(9), 1989–2001.

Luo, X. and P. V. Shevchenko (2010) "The *t* copula with multiple parameters of degrees of freedom: bivariate characteristics and application to risk management," *Quantitative Finance*, **10**(9), 1039–1054.

McNeil, A. J., R. Frey, and P. Embrechts (2015) *Quantitative Risk Management: Concepts, Techniques, and Tools*, Princeton University Press, revised ed.

Nikoloulopoulos, A. K., H. Joe, and H. Li (2012) "Vine copulas with asymmetric tail dependence and applications to financial return data," *Computational Statistics & Data Analysis*, **56**(11), 3659–3673.

Padoan, S. A. (2011) "Multivariate extreme models based on underlying skew-*t* and skew-normal distributions," *Journal of Multivariate Analysis*, **102**(5), 977–991.

Sahu, S. K., D. K. Dey, and M. D. Branco (2003) "A new class of multivariate skew distributions with applications to Bayesian regression models," *Canadian Journal of Statistics*, **31**(2), 129–150.

signal developers (2014) *signal: Signal processing*, R package version 0.7-4.

Smith, M. S., Q. Gan, and R. J. Kohn (2012) "Modelling dependence using skew *t* copulas: Bayesian inference and applications," *Journal of Applied Econometrics*, **27**(3), 500–522.

Appendix. Sample R code and some other empirical results

This appendix explains how to implement the MLE (maximum likelihood estimation) of skew-*t* copula parameters using the R statistical software. The skew-*t* copula is either the AC (Azzalini–Capitanio) or the GH (generalized hyperbolic) skew-*t* copula. It also shows some empirical results which are not referred in the main text.

A. Implementation of the MLE for AC skew-*t* copula

To implement MLE for the AC skew-*t* copula, we refer to the `sn` package for R which provides several functions to analyze multivariate skew-Normal and skew-*t* distributions. We modify the codes focusing on a standard multivariate AC skew-*t* distribution. The codes and some miscellaneous functions necessary for the standard multivariate AC skew-*t* distributions are collected in a file "sACstDef.R", which is described in Section C. This section describes the main implementation of the MLE for the AC skew-*t* copula using "sACstDef.R".

A.1. Main codes with functions for transforming parameters

The AC skew-*t* copula can be estimated by the maximum likelihood method by applying the `optim` function with the negative log-likelihood defined in the following box. Internal parameters are parameterized as $\theta_{21}, \theta_{31}, \dots, \theta_{d+1,1}, \theta_{32}, \dots, \theta_{d+1,2}, \dots, \theta_{d+1,d}, \ln(\nu-2)$. Transforming functions of the original parameters, $\{\rho_{21}, \dots, \rho_{d,d-1}\}$, $\boldsymbol{\delta}$ and ν to internal parameters $(\boldsymbol{\theta}, \ln(\nu-2))$, and vice versa, are implemented in the following way.

```
## AC skew-t copula estimation (MLE)
source("sACstDef.R");

## transforming original parameters to internal parameters
ACstIntPara <- function(rho,delta,nu){
  R <-  rhoToOmega(c(delta,rho));
  LTR <- t(chol(R));
  ndim <- nrow(LTR);
  theta <-  acos(LTR[2:ndim,1]);
  cumsin <- sin(theta)[-1];
  for(j in 2:(ndim-1)){
    thj <- acos(LTR[(j+1):ndim,j]/cumsin);
    theta <- c(theta,thj);
    cumsin <- (cumsin*sin(thj))[-1];
  }
  c(theta,log(nu-2.0));
}

## transforming internal parameters to original parameters
ACstOrgPara <- function(para){
  ntheta <- length(para)-1;
  theta <- para[1:ntheta];
  ndim <- (1+sqrt(1+8*ntheta))/2;
  LTR <- diag(ndim);
  LTR[-1,1] <- cos(theta[1:(ndim-1)]);
  cumsin <- sin(theta[1:(ndim-1)]);
  for(j in 2:(ndim-1)){
    LTR[j,j] <- cumsin[1];
    k <- (j-1)*(ndim-j/2)+1;
    thj <- theta[k:(k+ndim-j-1)];
```

```
    cumsin <- cumsin[-1];
    LTR[((j+1):ndim),j] <- cumsin*cos(thj);
    cumsin <- cumsin*sin(thj);
  }
  LTR[ndim,ndim] <- cumsin[1];
  R <- LTR %*% t(LTR);
  Omega <- R[-1,-1];
  delta <- R[1,-1];
  nu <- exp(para[ntheta+1])+2.0;
  list(rho = Omega[lower.tri(Omega)], delta = delta, nu = nu);
}


## negative log-likelihood for AC skew-t copula
ACstcopnll <- function(para, udat=NULL, rel.tol=1e-6, mpoints=150){
  dim <- ncol(udat);
  dp <- ACstOrgPara(para);
  delta <- dp$delta;
  zeta <- delta/sqrt(1-delta*delta);
  nu <- dp$nu;
  ix <- ipqsACst(udat,zeta,nu,mpoints=mpoints,rel.tol=rel.tol);
  ## Activate the following line instead of monotone interpolating
quantile
  ## function ipqsACst() to use accurate quantile function aqsACst()
  ## ix <- aqsACst(udat,zeta,nu,rel.tol=rel.tol);
  lm <- matrix(0,nrow=nrow(udat),ncol=dim);
  for(j in 1:dim){ lm[,j] <- ldsACst(ix[,j], zeta=zeta[j], nu=nu); }
  lc <- ldmsACst(ix,rho=dp$rho,delta=delta,nu=nu);
  -sum(lc)+sum(lm)
}

## MLE for AC skew-t copula using optim
ACstcop.mle <- function (udat = NULL, start = NULL, method = "Nelder-
Mead", rel.tol=1e-6, mpoints=150, ...){
  iniPar <- ACstIntPara(start$rho,start$delta,start$nu);
  fit <- optim(iniPar, ACstcopnll, method=method, hessian=FALSE,
udat=udat, rel.tol=rel.tol, mpoints=mpoints, ...);
  list(call = match.call(), dp = ACstOrgPara(fit$par), logL = -fit$value,
details=fit, nobs = nrow(udat), method = method);
}
```

A.2. Implementation of the MLE for a standard multivariate skew-*t* distribution

Similar to the AC skew-*t* copula, the multivariate standard skew-*t* distribution with an assumed location vector $\boldsymbol{\xi}^\top = (\xi_1, \dots, \xi_d) = (0, \dots, 0)$ and scale vector $\boldsymbol{\sigma}^\top = (\sigma_1, \dots, \sigma_d) = (1, \dots, 1)$ can be estimated by MLE as below.

```
## negative log-likelihood for standard (xi=0, omega=1)
## multivariate AC skew-t distribution
ACstdistnll <- function(para, xdat=NULL){
  dp <- ACstOrgPara(para);
  -sum(ldmsACst(xdat,rho=dp$rho,delta=dp$delta,nu=dp$nu));
}

## MLE for AC skew-t distribution with xi=0 and omega=1 using optim
ACstdist.mle <- function (xdat = NULL, start = NULL, method = "Nelder-
Mead", ...){
  iniPar <- ACstIntPara(start$rho,start$delta,start$nu);
  fit <- optim(iniPar, ACstdistnll, method=method, hessian=FALSE,
xdat=xdat, ...);
  list(call = match.call(), dp = ACstOrgPara(fit$par), logL = -fit$value,
details=fit, nobs = nrow(xdat), method = method);
}
```

A.3. Showing the result with the standard error of each estimate

To add the standard error for each estimate, we numerically calculate the hessian $H(\widehat{\boldsymbol{\theta}})$ of the log-likelihood function with original parameters $\boldsymbol{\theta} = (\Omega, \boldsymbol{\delta}, \nu)$ using the `numDeriv` library. The standard errors are obtained as the square roots of diagonal elements in $\{-H(\widehat{\boldsymbol{\theta}})\}^{-1}$.

```
library(numDeriv);

## log-likelihood of AC skew-t copula (for original parameters)
ACstcopLogLik <- function(x, udat=NULL, rel.tol=1e-6, mpoints=150){
  dim <- ncol(udat);
  lrho <- dim*(dim-1)/2;
  rho <- x[1:lrho];
  delta <- x[(lrho+1):(lrho+dim)];
  nu <- x[length(x)];
  zeta <- delta/sqrt(1-delta*delta);
  ix <- ipqsACst(udat,zeta,nu,mpoints=mpoints,rel.tol=rel.tol);
  ## Activate the following line instead of monotone interpolating
quantile
  ## function ipqsACst() to use accurate quantile function aqsACst()
  ## ix <- aqsACst(udat,zeta,nu,rel.tol=rel.tol);
  lm <- matrix(0,nrow=nrow(udat),ncol=dim);
  for(j in 1:dim){ lm[,j] <- ldsACst(ix[,j], zeta=zeta[j], nu=nu); }
  lc <- ldmsACst(ix,rho=rho,delta=delta,nu=nu);
  sum(lc)-sum(lm)
}

## estimated parameters with standard errors and the log-likelihood
showResSE <- function(fit,udat){
  dp <- fit$dp;
  para <- c(dp$rho,dp$delta,dp$nu);
  hess <- hessian(func=ACstcopLogLik, x=para, udat=udat);
  stdErrs <- sqrt(diag(solve(-hess)));
  resMat <- matrix(c(para,stdErrs),nrow=length(para),ncol=2);
  colnames(resMat) <- c("Estimate","Std. Error");
  dim <- ncol(udat);
  rhos <- NULL;
  for(i in 2:dim){ for(j in 1:(i-1)){ rhos <-
c(rhos,paste0("rho",i,j)) } }
  rownames(resMat) <- c(rhos,paste0("delta",1:dim),"nu");
  cat("Coefficients:\n");
  print(resMat);
  cat("-2 log L:",-2*fit$logL,"\n");
  cat("# of obs.:",nrow(udat),"\n");
}
```

A.4. Example of execution

With the previous functions in a file, "ACstcop.R", an example of executing the previous codes and the output from R is given as follows. Here, the file "MKdata.R" collects several functions to provide the daily return data $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ of three major stock indices: the Nikkei225, S&P500, and DAX, and the empirical distribution function values $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_N\}$. For the details, see Section D.

```
> ## Please specify the working directory by setwd()
> source("MKdata.R");
> source("ACstcop.R");
>
> ## unfiltered daily return of three major stock indices
> ## obsPeriod : 2010/4/1--2015/3/31
> ## dat$u is given by a version of empirical distribution function
> dat <- data3Stocks('2010-04-01::2015-03-31');
```

```
> dim <- ncol(dat$u);
>
> ## MLE (AC skew-t copula) ##
> start <- list(rho=numeric(dim*(dim-1)/2),delta=numeric(dim),nu=6);
> ## It would be better to add "trace=TRUE" option in control list
> system.time(stcopmle<-ACstcop.mle(dat$u, start=start,
control=list(reltol=1e-4)));
   user  system elapsed
  22.75    0.00   22.87
> system.time(showResSE(stcopmle,dat$u));
Coefficients:
         Estimate Std. Error
rho21   0.5352841 0.03958398
rho31   0.4917257 0.04990958
rho32   0.6757562 0.03441019
delta1 -0.5201134 0.18543992
delta2 -0.3175851 0.13795184
delta3 -0.6000269 0.16846268
nu      5.9767604 0.77364671
-2 log L: -1058.371
# of obs.: 1188
   user  system elapsed
  24.83    0.00   24.90
>
```

B. Implementation of the MLE with equi-skewness for the AC and GH skew-*t* copulas

Implementation of MLE with equi-skewness (or common skew parameter) of the skew-*t* copula is a special case of the skew-*t* copula. This section describes the implementations of the MLE with equi-skewness for the AC and GH skew-*t* copulas which are referred in the main document. The codes referred in Section B.1 and B.2 are collected in "ESstcop.R".

B.1. Main codes with equi-skewness for the AC skew-*t* copula

The AC skew-*t* copula with equi-skewness can be estimated by the maximum likelihood method by applying the `optim` function with defining negative log-likelihood using equi-skewness quantile functions collected in "ESqst.R" as follows. The internal parameters are parameterized as $\{\theta_{\cdot 1}, \theta_{32}, \dots, \theta_{d+1,2}, \dots, \theta_{d+1,d}\}$ and $\ln(\nu-2)$ while original parameters are given as δ, $\{\rho_{21}, \dots, \rho_{d,d-1}\}$, and ν.

```
## Equi-skewness AC skew-t copula estimation (MLE)
source("ESqst.R");

## transforming original parameters to internal parameters
ACEstIntPara <- function(rho,delta,nu){
  Omega <-  rhoToOmega(rho);
  LMat <- t(chol(Omega-delta*delta));
  ndim <- nrow(LMat);
  theta <-  acos(delta);
  cumsin <- sin(theta);
  for(j in 1:(ndim-1)){
    thj <- acos(LMat[(j+1):ndim,j]/cumsin);
    theta <- c(theta,thj);
    cumsin <- (cumsin*sin(thj))[-1];
  }
  c(theta,log(nu-2.0));
}
```

```
## transforming internal parameters to original parameters
ACEstOrgPara <- function(para){
  npara <- length(para);
  nrho <- npara-2;
  delta <- cos(para[1]);
  theta <- para[2:(nrho+1)];
  ndim <- (1+sqrt(1+8*nrho))/2;
  LMat <- diag(ndim);
  cumsin <- rep(sin(para[1]),length=ndim);
  k <- 1;
  for(j in 1:(ndim-1)){
    LMat[j,j] <- cumsin[1];
    thj <- theta[k:(k+ndim-j-1)];
    cumsin <- cumsin[-1];
    LMat[((j+1):ndim),j] <- cumsin*cos(thj);
    cumsin <- cumsin*sin(thj);
    k <- k + (ndim - j);
  }
  LMat[ndim,ndim] <- cumsin[1];
  Omega <- delta*delta + LMat %*% t(LMat);
  nu <- exp(para[npara])+2.0;
  list(rho = Omega[lower.tri(Omega)], delta = delta, nu = nu);
}


## negative log-likelihood for AC skew-t copula
ACEstcopnll <- function(para, udat=NULL, rel.tol=1e-6, mpoints=150){
  dim <- ncol(udat);
  dp <- ACEstOrgPara(para);
  delta <- dp$delta;
  zeta <- delta/sqrt(1-delta*delta);
  nu <- dp$nu;
  ix <- ipqsACEst(udat,zeta,nu,mpoints=mpoints,rel.tol=rel.tol);
  ## Activate the following line instead of monotone interpolating
quantile
  ## function ipqsACst() to use accurate quantile function aqsACst()
  ## ix <- aqsACst(udat,zeta,nu,rel.tol=rel.tol);
  lm <- ldsACst(ix, zeta=zeta, nu=nu);
  lc <-
ldmsACst(matrix(ix,ncol=dim),rho=dp$rho,delta=rep(delta,length=dim),nu=nu)
;
  -sum(lc)+sum(lm)
}

## MLE for AC skew-t copula using optim
ACEstcop.mle <- function (udat = NULL, start = NULL, method = "Nelder-
Mead", rel.tol=1e-6, mpoints=150, ...){
  iniPar <- ACEstIntPara(start$rho,start$delta,start$nu);
  fit <- optim(iniPar, ACEstcopnll, method=method, hessian=FALSE,
udat=udat, rel.tol=rel.tol, mpoints=mpoints, ...);
  list(call = match.call(), dp = ACEstOrgPara(fit$par), logL = -fit$value,
details=fit, nobs = nrow(udat), method = method);
}

## negative log-likelihood for standardized (xi=0, omega=1)
## multivariate AC skew-t distribution
ACEstdistnll <- function(para, xdat=NULL){
  dp <- ACEstOrgPara(para);
  dim <- ncol(xdat);
  -sum(ldmsACst(xdat,rho=dp$rho,delta=rep(dp$delta,length=dim),nu=dp$nu));
}

## MLE for AC skew-t distribution with xi=0 and omega=1 using optim
ACEstdist.mle <- function (xdat = NULL, start = NULL, method = "Nelder-
Mead", ...){
  iniPar <- ACEstIntPara(start$rho,start$delta,start$nu);
  fit <- optim(iniPar, ACEstdistnll, method=method, hessian=FALSE,
```

```
xdat=xdat, ...);
  list(call = match.call(), dp = ACEstOrgPara(fit$par), logL = -fit$value,
details=fit, nobs = nrow(xdat), method = method);
}
```

B.2. Main codes with equi-skewness for the GH skew-*t* copula

The GH skew-*t* copula with equi-skewness can be also estimated by the maximum likelihood method by applying the `optim` function with defining negative log-likelihood using equi-skewness quantile functions collected in "ESqst.R" as follows. While original parameters are given as $\{\rho_{21}, \dots, \rho_{d,d-1}\}$, γ and ν, the internal parameters are parameterized as $\{\theta_{21}, \dots, \theta_{d1}, \dots, \theta_{d,d-1}\}$, γ and $\ln(\nu - 2)$.

```
## Equi-skewness GH skew-t copula estimation (MLE)
source("ESqst.R");

## transforming original parameters to internal parameters ##
GHEstIntPara <- function(rho,gamma,nu){
  R <-  rhoToOmega(rho);
  LTR <- t(chol(R));
  dim <- nrow(LTR);
  theta <-  acos(LTR[2:dim,1]);
  cumsin <- sin(theta)[-1];
  if(dim>2){
    for(j in 2:(dim-1)){
      thj <- acos(LTR[(j+1):dim,j]/cumsin);
      theta <- c(theta,thj);
      cumsin <- (cumsin*sin(thj))[-1];
    }
  }
  c(theta,gamma,log(nu-2.0));
}

## transforming internal parameters to original parameters ##
GHEstOrgPara <- function(para){
  ntheta <- length(para)-2;
  dim <- (1+sqrt(1+8*ntheta))/2;
  theta <- para[1:ntheta];
  LTR <- diag(dim);
  LTR[-1,1] <- cos(theta[1:(dim-1)]);
  cumsin <- sin(theta[1:(dim-1)]);
  if(dim>2){
    for(j in 2:(dim-1)){
      LTR[j,j] <- cumsin[1];
      k <- (j-1)*(dim-j/2)+1;
      thj <- theta[k:(k+dim-j-1)];
      cumsin <- cumsin[-1];
      LTR[((j+1):dim),j] <- cumsin*cos(thj);
      cumsin <- cumsin*sin(thj);
    }
  }
  LTR[dim,dim] <- cumsin[1];
  Omega <- LTR %*% t(LTR);
  gamma <- para[ntheta+1];
  nu <- exp(para[ntheta+2])+2.0;
  list(rho = Omega[lower.tri(Omega)], gamma = gamma, nu = nu);
}

## negative log-likelihood for GH skew-t copula
GHEstcopnll <- function(para, udat=NULL, mpoints=150){
  dim <- ncol(udat);
  dp <- GHEstOrgPara(para);
  gamma <- dp$gamma;
  nu <- dp$nu;
```

```
  ix <- ipqsGHEst(udat,gamma,nu,mpoints=mpoints);
  ## Activate the following line instead of monotone interpolating
quantile
  ## function ipqsGHst() to use accurate quantile function aqsGHst()
  ## ix <- aqsGHst(udat,gamma,nu);
  ughyp <- ghyp(lambda=-nu/2, chi=nu, psi=0, sigma=1, gamma=gamma);
  lm <- dghyp(as.vector(ix), ughyp, logvalue=TRUE);
  mu <- rep(0,dim);
  mgamma <- rep(gamma,length=dim);
  mghyp <- ghyp(lambda=-nu/2, chi=nu, psi=0, mu=mu,
sigma=rhoToOmega(dp$rho), gamma=mgamma);
  lc <- dghyp(matrix(ix,ncol=dim), mghyp, logvalue=TRUE);
  -sum(lc)+sum(lm)
}

## MLE for GH skew-t copula using optim
GHEstcop.mle <- function (udat = NULL, start = NULL, method = "Nelder-
Mead", mpoints=150, ...){
  iniPar <- GHEstIntPara(start$rho,start$gamma,start$nu);
  fit <- optim(iniPar, GHEstcopnll, method=method, hessian=FALSE,
udat=udat, mpoints=mpoints, ...);
  list(call = match.call(), dp = GHEstOrgPara(fit$par), logL = -fit$value,
details=fit, nobs = nrow(udat), method = method);
}

## negative log-likelihood for multivariate standard GH skew-t
distribution
GHEstdistnll <- function(para, xdat=NULL){
  dp <- GHEstOrgPara(para);
  dim <- ncol(xdat);
  mu <- rep(0,dim);
  mgamma <- rep(dp$gamma,length=dim);
  mghyp <- ghyp(lambda=-dp$nu/2, chi=dp$nu, psi=0, mu=mu,
sigma=rhoToOmega(dp$rho), gamma=mgamma);
  -sum(dghyp(xdat, mghyp, logvalue=TRUE));
}

## MLE for multivariate standard GH skew-t distribution using optim
GHEstdist.mle <- function (xdat = NULL, start = NULL, method = "Nelder-
Mead", ...){
  iniPar <- GHEstIntPara(start$rho,start$gamma,start$nu);
  fit <- optim(iniPar, GHEstdistnll, method=method, hessian=FALSE,
xdat=xdat, ...);
  list(call = match.call(), dp = GHEstOrgPara(fit$par), logL = -fit$value,
details=fit, nobs = nrow(xdat), method = method);
}


## random number generator of multivariate standardized GH skew-t
distribution
rmsGHst <- function(n=1, rho, gamma, nu=Inf){
  d <- length(gamma);
  vv <- if(nu==Inf) 1 else rchisq(n,nu)/nu;
  Omega <- rhoToOmega(rho);
  z <- matrix(rnorm(n*d), n, d) %*% chol(Omega);
  gammaMat <- t(matrix(gamma, d, n));
  gammaMat/vv + z/sqrt(vv);
}


## random number generator of GH skew-t copula
rGHstcop <- function(n,rho,gamma,nu){
  d <- length(gamma);
  x <- rmsGHst(n,rho,gamma,nu);
  u <- matrix(0,n,d);
  for(j in 1:d){
    ughyp <- ghyp(lambda=-nu/2, chi=nu, psi=0, sigma=1, gamma=gamma[j]);
    u[,j] <- pghyp(x[,j], ughyp)
```

```
  }
  list(x=x,u=u);
}
```

B.3. Quantile functions for the AC and GH skew-*t* distributions with equi-skewness

Quantile functions for the AC and GH skew-*t* distributions with equi-skewness are implemented as follows using basic AC skew-*t* functions in "sACstDef.R" and GH library `ghyp`.

```
## Equi-skewness AC & GH skew-t quantiles
source("sACstDef.R");
library(ghyp);

## AC skew-t
## accurate quantiles for a standard (equi-delta) AC skew-t distribution
aqsACEst <- function(udat,zeta,nu,rel.tol=1e-6){
  ax <- qsACst(udat, zeta=zeta, nu=nu, rel.tol=rel.tol);
  matrix(ax,nrow=nrow(udat),ncol=ncol(udat));
}

## empirical quantiles with random sampling
## for a standard (equi-delta) AC skew-t distribution
rsqsACEst <- function(udat,zeta,nu,simNum){
  delta <- zeta/sqrt(1+zeta*zeta);
  ry <- sort(rsACst(simNum, delta, nu));
  rx <- ry[udat*(simNum-1)+1];
  matrix(rx,nrow=nrow(udat),ncol=ncol(udat));
}

## interpolating quantiles for a standard (equi-delta) AC skew-t
distribution
ipqsACEst <- function(udat,zeta,nu,mpoints=150,rel.tol=1e-6){
  minmaxu <- c(min(udat),max(udat));
  minmaxx <- qsACst(minmaxu, zeta=zeta, nu=nu, rel.tol=rel.tol);
  xx <- seq(minmaxx[1],minmaxx[2],length.out=mpoints);
  px <- sort(psACst(xx, zeta, nu, rel.tol=rel.tol));
  ix <- pchip(px, xx, as.vector(udat));
  matrix(ix,nrow=nrow(udat),ncol=ncol(udat));
}

## AC skew-t (integer nu)
## accurate quantiles for a standard (equi-delta) AC skew-t (integer nu)
aqsACIEst <- function(udat,zeta,nu,rel.tol=1e-6){
  ax <- qsACIst(udat, zeta=zeta, nu=nu, rel.tol=rel.tol);
  matrix(ax,nrow=nrow(udat),ncol=ncol(udat));
}

## interpolating quantiles for AC skew-t copula
## for positive integer nu (degree of freedom parameter)
ipqsACIEst <- function(udat,zeta,nu,mpoints=150,rel.tol=1e-6){
  minmaxu <- c(min(udat),max(udat));
  minmaxx <- qsACIst(minmaxu, zeta=zeta, nu=nu, rel.tol=rel.tol);
  xx <- seq(minmaxx[1],minmaxx[2],length.out=mpoints);
  px <- sort(psACIst(xx, zeta, nu, rel.tol=rel.tol));
  ix <- pchip(px, xx, as.vector(udat));
  matrix(ix,nrow=nrow(udat),ncol=ncol(udat));
}


## GH skew-t
## random number generator of GH skew-t copula
rGHEstcop <- function(n,rho,gamma,nu){
  vv <- if(nu==Inf) 1 else rchisq(n,nu)/nu;
  Omega <- rhoToOmega(rho);
  d <- nrow(Omega);
```

```
  z <- matrix(rnorm(n*d), n, d) %*% chol(Omega);
  gammaMat <- t(matrix(gamma, d, n));
  x <- gammaMat/vv + z/sqrt(vv);
  ughyp <- ghyp(lambda=-nu/2, chi=nu, psi=0, sigma=1, gamma=gamma);
  u <- matrix(pghyp(as.vector(x), ughyp),n,d);
  list(x=x,u=u);
}

## random number generator of standard GH skew-t distribution
rsGHst <- function(n=1, gamma, nu=Inf){
  vv <- if(nu==Inf) 1 else rchisq(n,nu)/nu;
  gamma/vv + rnorm(n)/sqrt(vv);
}

## accurate quantiles for (equi-gamma) GH skew-t
aqsGHEst <- function(udat,gamma,nu){
  ughyp <- ghyp(lambda=-nu/2, chi=nu, psi=0, sigma=1, gamma=gamma);
  ax <- qghyp(as.vector(udat), ughyp, method = "splines");
  matrix(ax,nrow=nrow(udat),ncol=ncol(udat));
}

## empirical quantiles with random sampling for (equi-gamma) GH skew-t
rsqsGHEst <- function(udat,gamma,nu,simNum){
  ry <- sort(rsGHst(simNum, gamma, nu));
  rx <- ry[udat*(simNum-1)+1];
  matrix(rx,nrow=nrow(udat),ncol=ncol(udat));
}

## interpolating quantiles for (equi-gamma) GH skew-t
ipqsGHEst <- function(udat,gamma,nu,mpoints=150){
  minmaxu <- c(min(udat),max(udat));
  ughyp <- ghyp(lambda=-nu/2, chi=nu, psi=0, sigma=1, gamma=gamma);
  minmaxx <- qghyp(minmaxu, ughyp, method = "splines");
  xx <- seq(minmaxx[1],minmaxx[2],length.out=mpoints);
  px <- sort(pghyp(xx, ughyp));
  ix <- pchip(px, xx, as.vector(udat));
  matrix(ix,nrow=nrow(udat),ncol=ncol(udat));
}
```

C. Basic functions for the standard multivariate AC skew-*t* distribution

This section describes the codes and some miscellaneous functions necessary for standard multivariate AC skew-*t* distributions. Those functions are based on the `sn` package for R which provides several functions to analyze multivariate skew-Normal and skew-*t* distributions. We modify the codes focusing on a standard multivariate AC skew-*t* distribution. That is because the implicit copula does not depend on the location vector and the scale vector. The codes presented in this section are collected in a file "sACstDef.R".

We use the `mnormt` library which provides distribution functions for the multivariate Student's-*t* and Normal distributions. We also use the `signal` library which provides the function of the piecewise cubic Hermite interpolating polynomial.

C.1. Standard multivariate AC skew-*t* density

For the standard multivariate AC skew-*t* distribution, the covariance matrix Ω is given by the correlations $\boldsymbol{\rho}$ which constructs the lower triangular of the covariance matrix. We provide the transformation function `rhoToOmega` of correlations $\boldsymbol{\rho}$ to Ω. Log-densities of the standard multivariate and univariate AC skew-*t* distribution are given in the following way.

```
library(mnormt)
library(signal)

## rho vector to Omega matrix ##
rhoToOmega <- function(rho){
  dim <- (sqrt(8*length(rho)+1)+1)/2;
  Omega <- diag(1/2,dim);
  Omega[lower.tri(Omega)] <- rho;
  Omega <- Omega + t(Omega);
  Omega;
}

## log-density of standard univariate AC skew-t distribution
ldsACst <- function (x, zeta, nu){
  pdf <- dt(x, df=nu, log=TRUE);
  cdf <- pt(zeta*x*sqrt((nu+1)/(x*x+nu)), df=nu+1, log.p=TRUE);
  logb(2) + pdf + cdf;
}

## density of standard univariate AC skew-t distribution
dsACst <- function (x, zeta, nu){
  pdf <- dt(x, df=nu);
  cdf <- pt(zeta*x*sqrt((nu+1)/(x*x+nu)), df=nu+1);
  2*pdf*cdf;
}

## log-density of standard multivariate AC skew-t distribution
ldmsACst <- function (x, rho, delta, nu){
  Omega <- rhoToOmega(rho);
  iOmega <- pd.solve(Omega, silent = TRUE, log.det = TRUE);
  if (is.null(iOmega)) return(NA)
  logDet <- attr(iOmega, "log.det");
  alpha <- iOmega %*% delta /sqrt(1-as.numeric(t(delta) %*% iOmega %*%
delta));
  d <- length(delta);
  x <- if (is.vector(x)) matrix(x, 1, d) else data.matrix(x);
  X <- t(x);
  Q <- apply((iOmega %*% X) * X, 2, sum);
  L <- as.vector(t(X/sqrt(diag(Omega))) %*% as.matrix(alpha));
  if (nu < 10000) {
    log.const <- lgamma((nu + d)/2) - lgamma(nu/2) - 0.5 * d * logb(nu);
    log1Q <- logb(1 + Q/nu)
  }
  else {
    log.const <- (-0.5 * d * logb(2) + log1p((d/2) * (d/2 - 1)/nu));
    log1Q <- log1p(Q/nu)
  }
  log.dmt <- log.const - 0.5 * (d * logb(pi) + logDet + (nu + d) * log1Q);
  log.pt <- pt(L * sqrt((nu + d)/(Q + nu)), df = nu + d, log.p = TRUE)
  logb(2) + log.dmt + log.pt
}
```

C.2. Distribution functions of standard univariate AC skew-*t* distributions

The distribution functions of standard AC skew-*t* distributions are implemented as follows. The function `pssn` is the distribution function of a standard skew-Normal which is described in Section C.7.

```
## cumulative density function of standard AC skew-t distribution
## with positive integer nu (degree of freedom parameter)
## The algorithm is given by Jamalizadeh, Khosravi, and Balakrishnan (2009)
```

```
psACst_int <- function (x, zeta=0, nu=Inf){
  if(nu != round(nu) | nu < 1) stop("nu not integer or not positive")
  if(nu == 1)
    atan(x)/pi + acos(zeta/sqrt((1+zeta^2)*(1+x^2)))/pi
    else { if(nu==2)
      0.5 - atan(zeta)/pi + (0.5 +
atan(x*zeta/sqrt(2+x^2))/pi)*x/sqrt(2+x^2)
    else
      (psACst_int(sqrt((nu-2)/nu)*x, zeta, nu-2) +
        psACst_int(sqrt(nu-1)*zeta*x/sqrt(nu+x^2), 0, nu-1) * x *
        exp(lgamma((nu-1)/2) +(nu/2-1)*log(nu)-0.5*log(pi)-lgamma(nu/2)
        -0.5*(nu-1)*log(nu+x^2)))
    }
}

## cumulative density function of standard multivariate AC skew-t
distribution
pmsACst <- function(x, zeta, nu, ...){
  if(any(abs(zeta) == Inf)) stop("Inf's in zeta are not allowed")
  d <- length(zeta)
  delta <- zeta/sqrt(1+zeta*zeta)
  Ocor <- diag(1,d)
  Obig <- matrix(rbind(c(1,-delta), cbind(-delta,Ocor)), d+1, d+1)
  z0 <- c(0,x)
  if(nu < .Machine$integer.max)
    p <- 2 * pmt(z0, mean=rep(0,d+1), S=Obig, df=nu, ...)
  else
    p <- 2 * pmnorm(z0, mean=rep(0,d+1), varcov=Obig, ...)
  p
}

## cumulative density function of standard AC skew-t distribution
## redefine pst on "sn" ver.1.2-0
psACst <- function (x, zeta, nu, ...){
  ok <- !(is.na(x) | (x==Inf) | (x==-Inf))
  y <- x[ok]
  if(abs(zeta) == Inf) {
    z0 <- replace(y, zeta*y < 0, 0)
    p <- pf(z0^2, 1, nu)
    return(if(zeta>0) p else (1-p))
  }
  fp <- function(v, zeta, nu, t.value)
     pssn(sqrt(v) * t.value, zeta) * dchisq(v * nu, nu) * nu
  if(round(nu)==nu){
    if(nu < (8.20 + 3.55* log(log(length(y)+1))))
      p <- psACst_int(y, zeta, nu)  # "method 4"
    else
      p <- pmsACst(y, zeta, nu, ...) # method 1
  }
  else{
    p <- numeric(length(y))
    upper <- 10 + 50/nu
    intdsst <- function(q) integrate(dsACst, -Inf, q, zeta, nu, ...)$value
    intfp <- function(q) integrate(fp, 0, Inf, zeta, nu, q, ...)$value
    idx2 <- (y<upper)
    idx3 <- !idx2
    p[idx2] <- sapply(y[idx2],intdsst)  # method 2
    p[idx3] <- sapply(y[idx3],intfp)  # method 3
  }
  pr <- rep(NA, length(x))
  pr[x==Inf] <- 1
  pr[x==-Inf] <- 0
  pr[ok] <- as.numeric(p)
  return(pmax(0,pmin(1,pr)))
}
```

C.3. Random number generator of AC skew-*t* copula

Random number generators of the standard AC skew-*t* distribution and AC skew-*t* copula are implemented as follows. The random number generator of the AC skew-*t* copula returns a list of an original sample $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ and the pseudo sample $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\}$.

```
## random number generator for standard univariate AC skew-t distribution
rsACst <- function(n=1, delta, nu=Inf){
  g <- if(nu==Inf) 1 else rchisq(n,nu)/nu;
  z <- delta * abs(rnorm(n)) + sqrt(1-delta*delta) * rnorm(n);
  z/sqrt(g);
}

## random number generator for standard multivariate AC skew-t distribution
rmsACst <- function(n=1, rho, delta, nu=Inf){
  d <- length(delta);
  g <- if(nu==Inf) 1 else rchisq(n,nu)/nu;
  zeta <- delta/sqrt(1-delta*delta);
  DD <- diag(sqrt(1+zeta*zeta));
  Ocor <- rhoToOmega(rho);
  Psi <- DD %*% (Ocor-outer(delta,delta)) %*% DD;
  Psi <- (Psi + t(Psi))/2;
  y <- matrix(rnorm(n*d), n, d) %*% chol(Psi);
  truncN <- abs(rnorm(n));
  truncN <- matrix(rep(truncN,d), ncol=d);
  z <- delta * t(truncN) + sqrt(1-delta*delta) * t(y);
  t(z)/sqrt(g);
}

## random number generator for AC skew-t copula
rACstcop <- function(n,rho,delta,nu,...){
  dim <- length(delta);
  x <- rmsACst(n=n,rho,delta,nu);
  u <- matrix(0,nrow=n,ncol=dim);
  zeta <- delta/sqrt(1-delta*delta);
  for(j in 1:dim){ u[,j] <- psACst(x[,j], zeta[j], nu,...); }
  list(x=x,u=u);
}
```

C.4. Accurate quantile function of a standard univariate AC skew-*t*

Based on the `qst` function of the `sn` package version 1.2-0, we modify the function as the following `qsACst`. The function is implemented by Newton method using the distribution function of the standard univariate AC skew-*t*. Because of the difference between the tolerance rate of the distribution function and that of the quantile function, the quantile function cannot exit the `while` loop with the default parameters in some cases.

To avoid the infinite loop, we modify the distribution function `psACst` to accept the optional parameters '...'. By setting the optional parameter `rel.tol = 1e-6`, for example, we can avoid the infinite loop. We also add `maxit` parameter to exit the loop when the number of iteration reaches the `maxit`. We also modify the function to return the `qt` value at zero skewness ζ.

```
## quantile function of standard AC skew-t distribution
## redefine qst on "sn" ver.1.2-0
qsACst <- function (p, zeta = 0, nu = Inf, tol = 1e-08, maxit = 30, ...){
  if (zeta == Inf)
    return(sqrt(qf(p, 1, nu)))
  if (zeta == -Inf)
    return(- sqrt(qf(1 - p, 1, nu)))
```

```
  if (zeta == 0)
    return(qt(p, nu))
  na <- is.na(p) | (p < 0) | (p > 1)
  abs.zeta <- abs(zeta)
  if (zeta < 0)
    p <- (1 - p)
  zero <- (p == 0)
  one <- (p == 1)
  x <- xa <- xb <- xc <- fa <- fb <- fc <- rep(NA, length(p))
  nc <- rep(TRUE, length(p))
  nc[(na | zero | one)] <- FALSE
  fc[!nc] <- 0
  xa[nc] <- qt(p[nc], nu)
  xb[nc] <- sqrt(qf(p[nc], 1, nu))
  fa[nc] <- psACst(xa[nc], abs.zeta, nu, ...) - p[nc]
  fb[nc] <- psACst(xb[nc], abs.zeta, nu, ...) - p[nc]
  regula.falsi <- FALSE
  it <- 0
  while ((sum(nc) > 0) & (it < maxit)) {
    xc[nc] <- if (regula.falsi)
      xb[nc] - fb[nc] * (xb[nc] - xa[nc])/(fb[nc] - fa[nc])
    else (xb[nc] + xa[nc])/2
    fc[nc] <- psACst(xc[nc], abs.zeta, nu, ...) - p[nc]
    pos <- (fc[nc] > 0)
    xa[nc][!pos] <- xc[nc][!pos]
    fa[nc][!pos] <- fc[nc][!pos]
    xb[nc][pos] <- xc[nc][pos]
    fb[nc][pos] <- fc[nc][pos]
    x[nc] <- xc[nc]
    nc[(abs(fc) < tol)] <- FALSE
    regula.falsi <- !regula.falsi
    it <- it + 1
  }
  x <- replace(x, zero, -Inf)
  x <- replace(x, one, Inf)
  q <- as.numeric(sign(zeta)* x)
  names(q) <- names(p)
  return(q)
}
```

C.5. Three methods to calculate quantiles

Three methods to calculate quantiles for a given pseudo sample $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N\}$ are implemented as follows. The first one is an accurate method using modified `qsACst` function given in C.4. The second one is empirical quantiles with random sampling. The third one uses a monotone interpolator.

For a monotone interpolator to calculate quantiles of a univariate skew-*t* distribution at high speed, the `signal` library for R provides a set of generally Matlab/Octave-compatible signal processing functions. We use the `pchip` function from this library for the piecewise cubic Hermite interpolating polynomial.[1]

```
## accurate quantiles for standard AC skew-t distribution
aqsACst <- function(udat,zeta,nu,rel.tol=1e-6){
  dim <- ncol(udat);
  ax <- matrix(0,nrow=nrow(udat),ncol=dim);
  for(j in 1:dim){
    ax[,j] <- qsACst(udat[,j], zeta=zeta[j], nu=nu, rel.tol=rel.tol);
```

[1] In some cases, the `pst` function of the `sn` package is not monotonically increasing because of the relative tolerance. We therefore sort the cumulative probabilities $p_1, \ldots, p_m$ for $x_1 < \cdots < x_m$.

```
  }
  ax
}

## empirical quantiles by random sampling for standard AC skew-t
distribution
rsqsACst <- function(udat,zeta,nu,simNum){
  dim <- ncol(udat);
  sx <- matrix(0,nrow=nrow(udat),ncol=dim);
  sy <- matrix(0,nrow=simNum,ncol=dim);
  delta <- zeta/sqrt(1+zeta*zeta);
  for(j in 1:dim){
     sy[,j] <- sort(rsACst(simNum, delta[j], nu));
     sx[,j] <- sy[udat[,j]*(simNum-1)+1,j];
  }
  sx
}


## interpolating quantiles for standard AC skew-t distribution
ipqsACst <- function(udat,zeta,nu,mpoints=150,rel.tol=1e-6){
  dim <- ncol(udat);
  ix <- matrix(0,nrow=nrow(udat),ncol=dim);
  for(j in 1:dim){
    minmaxu <- c(min(udat[,j]),max(udat[,j]));
    minmaxx <- qsACst(minmaxu, zeta=zeta[j], nu=nu, rel.tol=rel.tol);
    xx <- seq(minmaxx[1],minmaxx[2],length.out=mpoints);
    px <- sort(psACst(xx, zeta[j], nu, rel.tol=rel.tol));
    ix[,j] <- pchip(px, xx, udat[,j]);
  }
  ix
}
```

C.6. AC skew-*t* distribution with integer ν

For the AC skew-*t* distribution with integer ν, the cumulative densities (distribution functions) and quantiles can be calculated much faster than those for the AC skew-*t* distribution with fractional ν using the algorithm by Jamalizadeh *et al.* (2009). Those functions are implemented as follows.

```
## AC skew-t distribution with integer nu : CDF & quantiles
## cumulative density function of standard AC skew-t (integer nu)
psACIst <- function (x, zeta, nu, ...){
  ok <- !(is.na(x) | (x==Inf) | (x==-Inf))
  y <- x[ok]
  if(abs(zeta) == Inf) {
    z0 <- replace(y, zeta*y < 0, 0)
    p <- pf(z0^2, 1, nu)
    return(if(zeta>0) p else (1-p))
  }
  if(nu < (8.20 + 3.55* log(log(length(y)+1))))
    p <- psACst_int(y, zeta, nu)  # "method 4"
  else
    p <- pmsACst(y, zeta, nu, ...) # method 1
  pr <- rep(NA, length(x))
  pr[x==Inf] <- 1
  pr[x==-Inf] <- 0
  pr[ok] <- as.numeric(p)
  return(pmax(0,pmin(1,pr)))
}

## quantile function of standard AC skew-t distribution (integer nu)
qsACIst <- function (p, zeta = 0, nu = Inf, tol = 1e-08, maxit = 30, ...){
  if (zeta == Inf)
    return(sqrt(qf(p, 1, nu)))
  if (zeta == -Inf)
```

```
    return(- sqrt(qf(1 - p, 1, nu)))
  if (zeta == 0)
    return(qt(p, nu))
  na <- is.na(p) | (p < 0) | (p > 1)
  abs.zeta <- abs(zeta)
  if (zeta < 0)
    p <- (1 - p)
  zero <- (p == 0)
  one <- (p == 1)
  x <- xa <- xb <- xc <- fa <- fb <- fc <- rep(NA, length(p))
  nc <- rep(TRUE, length(p))
  nc[(na | zero | one)] <- FALSE
  fc[!nc] <- 0
  xa[nc] <- qt(p[nc], nu)
  xb[nc] <- sqrt(qf(p[nc], 1, nu))
  fa[nc] <- psACIst(xa[nc], abs.zeta, nu, ...) - p[nc]
  fb[nc] <- psACIst(xb[nc], abs.zeta, nu, ...) - p[nc]
  regula.falsi <- FALSE
  it <- 0
  while ((sum(nc) > 0) & (it < maxit)) {
    xc[nc] <- if (regula.falsi)
      xb[nc] - fb[nc] * (xb[nc] - xa[nc])/(fb[nc] - fa[nc])
    else (xb[nc] + xa[nc])/2
    fc[nc] <- psACIst(xc[nc], abs.zeta, nu, ...) - p[nc]
    pos <- (fc[nc] > 0)
    xa[nc][!pos] <- xc[nc][!pos]
    fa[nc][!pos] <- fc[nc][!pos]
    xb[nc][pos] <- xc[nc][pos]
    fb[nc][pos] <- fc[nc][pos]
    x[nc] <- xc[nc]
    nc[(abs(fc) < tol)] <- FALSE
    regula.falsi <- !regula.falsi
    it <- it + 1
  }
  x <- replace(x, zero, -Inf)
  x <- replace(x, one, Inf)
  q <- as.numeric(sign(zeta)* x)
  names(q) <- names(p)
  return(q)
}
```

C.7. Densities, distribution functions, and quantiles of standard skew-Normal

The log-densities, distribution functions, and quantiles of standard skew-Normal distributions are implemented as follows.

```
### For skew-Normal copula  ###
## log-density of standard univariate skew-Normal distribution
ldssn <- function(x,zeta){
 logN <- (-log(sqrt(2*pi)) -x^2/2);
 if(abs(zeta) < Inf)
   logS <- pnorm(zeta*x, log.p=TRUE)
 else
   logS  <- log(as.numeric(sign(zeta)*x > 0))
 as.numeric(logN + logS - log(0.5));
}

## log-density of standard multivariate skew-Normal distribution
ldmssn <- function(x, rho, delta){
  Omega <- rhoToOmega(rho);
  iOmega <- pd.solve(Omega, silent = TRUE, log.det = TRUE);
  if (is.null(iOmega)) return(NA)
  logDet <- attr(iOmega, "log.det");
  alpha <- iOmega %*% delta /sqrt(1-as.numeric(t(delta) %*% iOmega %*%
delta));
  d <- length(delta);
```

```
  x <- if (is.vector(x)) matrix(x, 1, d) else data.matrix(x);
  X <- t(x);
  Q <- colSums((iOmega %*% X) * X);
  L <- as.vector(t(X/sqrt(diag(Omega))) %*% as.matrix(alpha));
  logPDF <- (logb(2) - 0.5 * Q + pnorm(L, log.p = TRUE)
               - 0.5 * (d * logb(2 * pi) + logDet));
  logPDF;
}

## cumulative density function of standard skew-Normal distribution
pssn <- function(x, zeta=0, engine, ...){
  nx <- length(x)
  na <- length(zeta)
  if(missing(engine)) engine <-
    if(na == 1 & nx > 3 & all(zeta*x > -5))
      "T.Owen" else "biv.nt.prob"
  if(engine == "T.Owen") {
    if(na > 1) stop("engine='T.Owen' not compatible with other arguments")
    p <- pnorm(x) - 2 * T.Owen(x, zeta, ...)
  }
  else{ #  engine="biv.nt.prob"
    p <- numeric(nx)
    zeta <- cbind(x, zeta)[,2]
    delta <- zeta/sqrt(1+zeta*zeta)
    for(k in seq_len(nx)) {
      if(abs(zeta[k]) == Inf){
       p[k] <- if(zeta[k] > 0)
             2*pnorm(pmax(x[k],0)) - 1
           else
             2*pnorm(pmin(x[k],0))
      }
      else { # SNbook: formula (2.48), p.40
        R <- matrix(c(1, -delta[k], -delta[k], 1), 2, 2)
        p[k] <- 2 * biv.nt.prob(0, rep(-Inf,2), c(x[k], 0), rep(0, 2), R)
      }
    }
  }
  pmin(1, pmax(0, as.numeric(p)))
}

## quantile function of standard skew-Normal distribution
## using "NR" solver
qssn <- function(p, zeta = 0, tol = 1e-08, ...){
  max.q <- sqrt(qchisq(p,1));
  min.q <- -sqrt(qchisq(1-p,1));
  if(zeta == Inf)  return(as.numeric(max.q))
  if(zeta == -Inf) return(as.numeric(min.q))
  na <- is.na(p) | (p < 0) | (p > 1)
  zero <- (p == 0)
  one <- (p == 1)
  p <- replace(p, (na | zero | one), 0.5)
  delta <- zeta/sqrt(1+zeta^2);
  ex <- sqrt(2/pi)*delta;
  vx <- 1-ex^2;
  sx <- 0.5*(4-pi)*(ex^3);
  kx <- 2*(pi-3)*(ex^4);
  g1 <- sx/(vx^(3/2));
  g2 <- kx/(vx^2);
  x <- qnorm(p)
  x <- (x + (x^2 - 1) * g1/6 + x * (x^2 - 3) * g2/24 -
          x * (2 * x^2 - 5) * g1^2/36)
  x <- ex + sqrt(vx) * x
  px <- pssn(x, zeta=zeta, ...);
  max.err <- 1
  while (max.err > tol) {
    logPDF <- ldssn(x,zeta);
    x1 <- x - (px - p)/exp(logPDF);
    x <- x1
```

```
    px <- pssn(x, zeta=zeta, ...)
    max.err <- max(abs(px-p))
    if(is.na(max.err)) stop('failed convergence, try with solver="RFB"')
  }
  x <- replace(x, na, NA)
  x <- replace(x, zero, -Inf)
  x <- replace(x, one, Inf)
  q <- as.numeric(x)
  names(q) <- names(p)
  return(q)
}

## accurate quantiles for skew-Normal copula
aqssn <- function(udat,zeta){
  dim <- ncol(udat);
  ax <- matrix(0,nrow=nrow(udat),ncol=dim);
  for(j in 1:dim){
    ax[,j] <- qssn(udat[,j], zeta=zeta[j]);
  }
  ax
}

## interpolating quantiles for skew-Normal copula
ipqssn <- function(udat,zeta,mpoints=150){
  dim <- ncol(udat);
  ix <- matrix(0,nrow=nrow(udat),ncol=dim);
  for(j in 1:dim){
    minmaxu <- c(min(udat[,j]),max(udat[,j]));
    minmaxx <- qssn(minmaxu, zeta=zeta[j]);
    xx <- seq(minmaxx[1],minmaxx[2],length.out=mpoints);
    px <- sort(pssn(xx, zeta[j]));
    ix[,j] <- pchip(px, xx, udat[,j]);
  }
  ix
}

### from "sn" ver.1.2-0  ###
T.Owen <- function(h, a, jmax=50, cut.point=8){
  T.int <-function(h, a, jmax, cut.point){
     fui <- function(h,i) (h^(2*i))/((2^i)*gamma(i+1))
     seriesL <- seriesH <- NULL
     i  <- 0:jmax
     low<- (h <= cut.point)
     hL <- h[low]
     hH <- h[!low]
     L  <- length(hL)
     if (L > 0) {
       b    <- outer(hL, i, fui)
       cumb <- apply(b, 1, cumsum)
       b1   <- exp(-0.5*hL^2) * t(cumb)
       matr <- matrix(1, jmax+1, L) - t(b1)
       jk   <- rep(c(1,-1), jmax)[1:(jmax+1)]/(2*i+1)
       matr <- t(matr*jk) %*%  a^(2*i+1)
       seriesL  <- (atan(a) - as.vector(matr))/(2*pi)
     }
     if (length(hH) > 0)  seriesH <-
          atan(a)*exp(-0.5*(hH^2)*a/atan(a)) * (1+0.00868*(hH*a)^4)/(2*pi)
     series <- c(seriesL, seriesH)
     id <- c((1:length(h))[low],(1:length(h))[!low])
     series[id] <- series  # re-sets in original order
     series
  }
  if(!is.vector(a) | length(a)>1) stop("'a' must be a vector of length 1")
  if(!is.vector(h)) stop("'h' must be a vector")
  aa <- abs(a)
  ah <- abs(h)
  if(is.na(aa)) stop("parameter 'a' is NA")
  if(aa==Inf) return(sign(a)*0.5*pnorm(-ah)) # sign(a): 16.07.2007
```

```
  if(aa==0)   return(rep(0,length(h)))
  na  <- is.na(h)
  inf <- (ah == Inf)
  ah  <- replace(ah,(na|inf),0)
  if(aa <= 1)
    owen <- T.int(ah,aa,jmax,cut.point)
  else
    owen<- (0.5*pnorm(ah) + pnorm(aa*ah)*(0.5-pnorm(ah))
               - T.int(aa*ah,(1/aa),jmax,cut.point))
  owen <- replace(owen,na,NA)
  owen <- replace(owen,inf,0)
  return(owen*sign(a))
}
```

D. Tail dependences of the bivariate AC skew-*t* copula

Tail dependences of the bivariate AC skew-*t* copula refered in Section 2.2 in the main document are calculated by the following functions.

```
## density of extended standard univariate AC skew-t distribution
desACst <- function (z, alpha, tau, nu){
  pdf <- dt(z, df=nu);
  cdf1 <- pt((alpha*z+tau)*sqrt((nu+1)/(nu+z*z)), df=nu+1);
  cdf2 <- pt(tau/sqrt(1+alpha*alpha), df=nu);
  pdf*cdf1/cdf2;
}

## Tail dependencies of standard bivariate AC skew-t distribution
tdACst <- function (delta, rho, nu){
  r1 <- 1-rho*rho;
  sqr1 <- sqrt(r1);
  rd <- sum(delta*delta)-2*rho*delta[1]*delta[2];
  alpha <- (delta-rho*c(delta[2],delta[1]))/sqrt(r1+rd);
  zeta <- delta/sqrt(1-delta*delta);
  nu1 <- nu+1;
  sqnu1 <- sqrt(nu1);
  frac12 <- pt(-zeta[1]*sqnu1, df=nu1)/pt(-zeta[2]*sqnu1, df=nu1);
  avec <- (c(frac12,1/frac12)^(1/nu)-rho)*sqnu1/sqr1;
  al <- alpha*sqr1;
  tau <- sqnu1*(alpha+rho*c(alpha[2],alpha[1]));
  desACstl1 <- function (z){ desACst(z, al[2], -tau[1], nu1); }
  desACstl2 <- function (z){ desACst(z, al[1], -tau[2], nu1); }
  ltd1 <- integrate(desACstl1,-Inf,-avec[2])$value;
  ltd2 <- integrate(desACstl2,-Inf,-avec[1])$value;
  ltd <- ltd1+ltd2;
  desACstu1 <- function (z){ desACst(z, al[2], tau[1], nu1); }
  desACstu2 <- function (z){ desACst(z, al[1], tau[2], nu1); }
  utd1 <- integrate(desACstu1,avec[2],Inf)$value;
  utd2 <- integrate(desACstu2,avec[1],Inf)$value;
  utd <- utd1+utd2;
  c(ltd,utd);
}
```

The lower and upper tail dependences of the bivariate AC skew-*t* copulas plotted in Fig.1 in the main document are calculated as follows.

```
> rho <- .5;
> nu <- 3;
> minDelta <- -sqrt((1+rho)/2);
> minDelta;
[1] -0.8660254
> tdACst(c(0,0),rho,nu);
[1] 0.3125 0.3125
> tdACst(c(-0.7,-0.7),rho,nu);
```

```
[1] 0.3355492 0.2243023
> tdACst(c(minDelta,minDelta),rho,nu);
[1] 0.3354413 0.2107109
```

E. Filtered and unfiltered stock indices returns

The trivariate daily return data of Nikkei225, S&P500, and DAX are constructed as follows. The codes are collected in a file "MKdata.R". We use `xts` library to deal with time series data. For filtered return, we use `rugarch` library to obtain standardized residuals of various GARCH-type spec. "StockPrice.csv" is the daily stock price data from the end of 1994 obtained from Yahoo!Finance of Nikkei225 (^N225), S&P500 (^GSPC), and DAX (^GDAXI).

```
library(xts)
library(rugarch)

## construction of pseudo observations using an empirical distribution
mkPseudoDat <- function(orgdat){
  dim <- ncol(orgdat);
  N <- nrow(orgdat);
  u <- x <- matrix(0,nrow=N,ncol=dim);
  for(j in 1:dim){
    x[,j] <- orgdat[,j];
    Fx <- ecdf(x[,j]);
    u[,j] <- Fx(x[,j])*N/(N+1);
  }
  list(x=x,u=u);
}

## unfiltered daily return data of Nikkei225, S&P500, and DAX
data3Stocks <- function(obsperiod){
  SPdat<-as.xts(read.zoo("StockPrice.csv",header=T,sep=","));
  RSPdat<-diff(log(SPdat));
  RSPdat<-RSPdat['1995-01-04::'];
  SRSPdat<-cbind(lag(RSPdat$NK225,k=-1),RSPdat$SP500,RSPdat$DAX)[obsperiod];
  mkPseudoDat(SRSPdat);
}

## standardized residual data of GARCH spec for daily return
## of Nikkei225, S&P500, and DAX
stdRes3Stocks<- function(obsperiod,spec){
  SPdat<-as.xts(read.zoo("StockPrice.csv",header=T,sep=","));
  RSPdat<-diff(log(SPdat));
  RSPdat<-RSPdat['1995-01-04::'];
  SRSPdat<-cbind(lag(RSPdat$NK225,k=-1),RSPdat$SP500,RSPdat$DAX)[obsperiod];
  fit1 <- ugarchfit(spec=spec, data=SRSPdat$NK225);
  fit2 <- ugarchfit(spec=spec, data=SRSPdat$SP500);
  fit3 <- ugarchfit(spec=spec, data=SRSPdat$DAX);
  res1 <- residuals(fit1,standardize=TRUE);
  res2 <- residuals(fit2,standardize=TRUE);
  res3 <- residuals(fit3,standardize=TRUE);
  mkPseudoDat(cbind(res1,res2,res3));
}
```

F. MLE for Student-*t*, skew-Normal, Normal copulas

The maximum likelihood method for the Student-*t*, skew-Normal, Normal copulas can be implemented in the same way as the skew-*t* copula. This section shows the codes and an example of the comparison.

F.1. Transforming parameters and negative log-likelihood

To implement MLE for the Student-*t*, (equi-skewness) skew-Normal, Normal copulas, we define the functions of transforming parameters and the negative log-likelihood for each copula as follows. The codes are collected in a file "ESsntncop.R".

```
## skew-Normal (equi-skewness), t, Normal copula estimation (MLE)

source("sACstDef.R")

## skew-Normal copula
## interpolating quantiles for a standard (equi-delta) skew-Normal
distribution
ipqsesn <- function(udat,zeta,mpoints=150){
  minmaxu <- c(min(udat),max(udat));
  minmaxx <- qssn(minmaxu, zeta=zeta);
  xx <- seq(minmaxx[1],minmaxx[2],length.out=mpoints);
  px <- sort(pssn(xx, zeta));
  ix <- pchip(px, xx, as.vector(udat));
  ix
}

## transforming original parameters to internal parameters
esnIntTPara <- function(rho,delta){
  Omega <-  rhoToOmega(rho);
  LMat <- t(chol(Omega-delta*delta));
  ndim <- nrow(LMat);
  theta <-  acos(delta);
  cumsin <- sin(theta);
  for(j in 1:(ndim-1)){
    thj <- acos(LMat[(j+1):ndim,j]/cumsin);
    theta <- c(theta,thj);
    cumsin <- (cumsin*sin(thj))[-1];
  }
  theta;
}

## transforming internal parameters to original parameters
esnOrgTPara <- function(para){
  npara <- length(para);
  nrho <- npara-1;
  delta <- cos(para[1]);
  theta <- para[2:npara];
  ndim <- (1+sqrt(1+8*nrho))/2;
  LMat <- diag(ndim);
  cumsin <- rep(sin(para[1]),length=ndim);
  k <- 1;
  for(j in 1:(ndim-1)){
    LMat[j,j] <- cumsin[1];
    thj <- theta[k:(k+ndim-j-1)];
    cumsin <- cumsin[-1];
    LMat[((j+1):ndim),j] <- cumsin*cos(thj);
    cumsin <- cumsin*sin(thj);
    k <- k + (ndim - j);
  }
  LMat[ndim,ndim] <- cumsin[1];
  Omega <- delta*delta + LMat %*% t(LMat);
  list(rho = Omega[lower.tri(Omega)], delta = delta);
}

## negative log-likelihood for skew-Normal copula
## using interpolating quantiles
esncopnll <- function(para, udat=NULL, mpoints=150){
  dim <- ncol(udat);
```

```
  dp <- esnOrgTPara(para);
  delta <- dp$delta;
  zeta <- delta/sqrt(1-delta*delta);
  ix <- ipqsesn(udat,zeta,mpoints);
  lm <- ldssn(ix, zeta);
  lc <- ldmssn(matrix(ix,ncol=dim),rho=dp$rho,delta=rep(delta,length=dim));
  -sum(lc)+sum(lm)
}

## t copula
tIntTPara <- function(rho,nu){
  R <-  rhoToOmega(rho);
  LTR <- t(chol(R));
  ndim <- nrow(LTR);
  theta <-  acos(LTR[2:ndim,1]);
  cumsin <- sin(theta)[-1];
  if(ndim>=3){
    for(j in 2:(ndim-1)){
      thj <- acos(LTR[(j+1):ndim,j]/cumsin);
      theta <- c(theta,thj);
      cumsin <- (cumsin*sin(thj))[-1];
    }
  }
  c(theta,log(nu-2.0));
}

tOrgTPara <- function(tpara){
  ntpara <- length(tpara);
  eta <- tpara[ntpara];
  theta <- tpara[-ntpara];
  ntheta <- length(theta);
  ndim <- (1+sqrt(1+8*ntheta))/2;
  LTR <- diag(ndim);
  LTR[-1,1] <- cos(theta[1:(ndim-1)]);
  cumsin <- sin(theta[1:(ndim-1)]);
  if(ndim>=3){
    for(j in 2:(ndim-1)){
      LTR[j,j] <- cumsin[1];
      k <- (j-1)*(ndim-j/2)+1;
      thj <- theta[k:(k+ndim-j-1)];
      cumsin <- cumsin[-1];
      LTR[((j+1):ndim),j] <- cumsin*cos(thj);
      cumsin <- cumsin*sin(thj);
    }
  }
  LTR[ndim,ndim] <- cumsin[1];
  Omega <- LTR %*% t(LTR);
  nu <- exp(eta)+2.0;
  list(rho = Omega[lower.tri(Omega)], nu = nu);
}

## negative log-density for t copula
tcopnll <- function(tpara, udat=NULL){
  dp <- tOrgTPara(tpara);
  Omega <- rhoToOmega(dp$rho);
  nu <- dp$nu;
  dim <- ncol(udat);
  ax <- qt(udat, df=nu);
  lm <- dt(ax, df=nu, log=TRUE);
  lc <- dmt(matrix(ax,ncol=dim),S=Omega,df=nu,log=TRUE);
  -sum(lc)+sum(lm)
}


## Normal copula
nIntTPara <- function(rho){
  R <- rhoToOmega(rho);
  LTR <- t(chol(R));
```

```
  ndim <- nrow(LTR);
  theta <-  acos(LTR[2:ndim,1]);
  cumsin <- sin(theta)[-1];
  if(ndim>=3){
    for(j in 2:(ndim-1)){
      thj <- acos(LTR[(j+1):ndim,j]/cumsin);
      theta <- c(theta,thj);
      cumsin <- (cumsin*sin(thj))[-1];
    }
  }
  theta;
}

nOrgTPara <- function(theta){
  ntheta <- length(theta);
  ndim <- (1+sqrt(1+8*ntheta))/2;
  LTR <- diag(ndim);
  LTR[-1,1] <- cos(theta[1:(ndim-1)]);
  cumsin <- sin(theta[1:(ndim-1)]);
  if(ndim>=3){
    for(j in 2:(ndim-1)){
      LTR[j,j] <- cumsin[1];
      k <- (j-1)*(ndim-j/2)+1;
      thj <- theta[k:(k+ndim-j-1)];
      cumsin <- cumsin[-1];
      LTR[((j+1):ndim),j] <- cumsin*cos(thj);
      cumsin <- cumsin*sin(thj);
    }
  }
  LTR[ndim,ndim] <- cumsin[1];
  Omega <- LTR %*% t(LTR);
  Omega[lower.tri(Omega)];
}

## negative log-density for Normal copula
ncopnll <- function(theta, udat=NULL){
  Omega <- rhoToOmega(nOrgTPara(theta));
  dim <- ncol(udat);
  ax <- lm <- matrix(0,nrow=nrow(udat),ncol=dim);
  for(j in 1:dim){
    ax[,j] <- qnorm(udat[,j]);
    lm[,j] <- dnorm(ax[,j], log=TRUE);
  }
  lc <- dmnorm(ax,varcov=Omega,log=TRUE);
  -sum(lc)+sum(lm)
}
```

F.2. An example of comparison

Using the codes in Section F.1 collected in a file "ESsntncop.R", an example of comparison among the AC skew-*t*, GH skew-*t*, Student-*t*, skew-Normal, and Normal copulas is given as follows.

```
> ## Please specify the working directory by setwd()
> source("MKdata.R");
> source("ESstcop.R");
> source("ESsntncop.R");
>
> resTab <- function(res){
+   rowNames <-
c("rho21","rho31","rho32","delta/gamma","nu","logLik","AIC","BIC","Time","Conv.","Coun
t","irho21","irho31","irho32","idelta/igamma","inu");
+   colNames <- c("AC skew-t","GH skew-t","Student-t","skew-Normal","Normal")
+   resTab <- matrix(NA,nrow=length(rowNames),ncol=length(colNames));
+   nobs <- res$nobs;
+   fitres <- res$fits;
+   resTab[1:5,1] <- unlist(fitres[[1]]$dp);
+   resTab[1:5,2] <- unlist(fitres[[2]]$dp);
+   tRes <- unlist(fitres[[3]]$dp);
```

```
+   resTab[1:3,3] <- tRes[1:3];
+   resTab[5,3] <- tRes[4];
+   resTab[1:4,4] <- unlist(fitres[[4]]$dp);
+   resTab[1:3,5] <- unlist(fitres[[5]]$dp);
+   for(j in 1:length(colNames)){
+     fit <- fitres[[j]];
+     details <- fit$details;
+     stime <- fit$stime;
+     logLik <- fit$logL;
+     npara <- fit$npara;
+     AIC <- -2*logLik+2*npara;
+     BIC <- -2*logLik+log(nobs)*npara;
+     resTab[6:11,j] <- c(logLik,AIC,BIC,stime,details$convergence,details$counts[1]);
+   }
+   resTab[12:16,1] <- unlist(fitres[[1]]$inidp);
+   resTab[12:16,2] <- unlist(fitres[[2]]$inidp);
+   tini <- unlist(fitres[[3]]$inidp);
+   resTab[12:14,3] <- tini[1:3];
+   resTab[16,3] <- tini[4];
+   resTab[12:15,4] <- unlist(fitres[[4]]$inidp);
+   resTab[12:14,5] <- unlist(fitres[[5]]$inidp);
+   rownames(resTab) <- rowNames;
+   colnames(resTab) <- colNames;
+   return(resTab);
+ }
>
> mlesttsnn <- function(spec=NULL,obsPeriod){
+   ## obtain data
+   if(is.null(spec)) dat <- data3Stocks(obsPeriod)
+   else dat <- stdRes3Stocks(obsPeriod,spec);
+   dim <- ncol(dat$u);
+
+   ## MLE (t copula) ##
+   iniTdp <- list(rho=numeric(dim*(dim-1)/2),nu=6);
+   stimeT <- system.time(fit<-optim(tIntTPara(iniTdp$rho,iniTdp$nu), tcopnll,
udat=dat$u, control=list(reltol=1e-4)));
+   tdp <- tOrgTPara(fit$par);
+   tcopmle <- list(inidp = iniTdp, dp = tdp, logL = -fit$value, stime = stimeT[3],
npara = length(fit$par), details=fit);
+
+   ## MLE (AC skew-t copula) ##
+   iniSTdp1 <- list(rho=tdp$rho,delta=0,nu=tdp$nu);
+   iniSTPar <- ACEstIntPara(iniSTdp1$rho,iniSTdp1$delta,iniSTdp1$nu);
+   stimeAST1 <- system.time(fit<-optim(iniSTPar, ACEstcopnll, udat=dat$u,
control=list(reltol=1e-4)));
+   astcopmle1 <- list(inidp = iniSTdp1, dp = ACEstOrgPara(fit$par), logL = -
fit$value, stime = stimeAST1[3], npara = length(fit$par), details=fit);
+   astcopmle <- astcopmle1;
+   iniSTdp2 <- list(rho=numeric(dim*(dim-1)/2),delta=0,nu=6);
+   iniSTPar <- ACEstIntPara(iniSTdp2$rho,iniSTdp2$delta,iniSTdp2$nu);
+   stimeAST2 <- system.time(fit<-optim(iniSTPar, ACEstcopnll, udat=dat$u,
control=list(reltol=1e-4)));
+   astcopmle2 <- list(inidp = iniSTdp2, dp = ACEstOrgPara(fit$par), logL = -
fit$value, stime = stimeAST2[3], npara = length(fit$par), details=fit);
+   if(astcopmle2$logL > astcopmle$logL){ astcopmle <- astcopmle2; }
+
+   ## MLE (GH skew-t copula) ##
+   iniSTdp1 <- list(rho=tdp$rho,gamma=0,nu=tdp$nu);
+   iniSTPar <- GHEstIntPara(iniSTdp1$rho,iniSTdp1$gamma,iniSTdp1$nu);
+   stimeGST1 <- system.time(fit<-optim(iniSTPar, GHEstcopnll, udat=dat$u,
control=list(reltol=1e-4)));
+   gstcopmle1 <- list(inidp = iniSTdp1, dp = GHEstOrgPara(fit$par), logL = -
fit$value, stime = stimeGST1[3], npara = length(fit$par), details=fit);
+   gstcopmle <- gstcopmle1;
+   iniSTdp2 <- list(rho=numeric(dim*(dim-1)/2),gamma=0,nu=6);
+   iniSTPar <- GHEstIntPara(iniSTdp2$rho,iniSTdp2$gamma,iniSTdp2$nu);
+   stimeGST2 <- system.time(fit<-optim(iniSTPar, GHEstcopnll, udat=dat$u,
control=list(reltol=1e-4)));
+   gstcopmle2 <- list(inidp = iniSTdp2, dp = GHEstOrgPara(fit$par), logL = -
fit$value, stime = stimeGST2[3], npara = length(fit$par), details=fit);
+   if(gstcopmle2$logL > gstcopmle$logL){ gstcopmle <- gstcopmle2; }
+
+   ## MLE (Normal copula) ##
+   iniNdp <- list(rho=numeric(dim*(dim-1)/2));
+   stimeN <- system.time(fit<-optim(nIntTPara(iniNdp$rho), ncopnll, udat=dat$u,
control=list(reltol=1e-4)));
```

```
+   ndp <- list(rho=nOrgTPara(fit$par));
+   ncopmle <- list(inidp = iniNdp, dp = ndp, logL = -fit$value, stime = stimeN[3],
npara = length(fit$par), details=fit);
+
+   ## MLE (skew Normal copula) ##
+   iniSNdp1 <- list(rho=ndp$rho,delta=0);
+   iniSNPar <- esnIntTPara(iniSNdp1$rho,iniSNdp1$delta);
+   stimeSN <- system.time(fit<-optim(iniSNPar, esncopnll, udat=dat$u,
control=list(reltol=1e-4)));
+   sncopmle1 <- list(inidp = iniSNdp1, dp = esnOrgTPara(fit$par), logL = -fit$value,
stime = stimeSN[3], npara = length(fit$par), details=fit);
+   sncopmle <- sncopmle1;
+
+   iniSNdp2 <- list(rho=numeric(dim*(dim-1)/2),delta=0);
+   iniSNPar <- esnIntTPara(iniSNdp2$rho,iniSNdp2$delta);
+   stimeSN <- system.time(fit<-optim(iniSNPar, esncopnll, udat=dat$u,
control=list(reltol=1e-4)));
+   sncopmle2 <- list(inidp = iniSNdp2, dp = esnOrgTPara(fit$par), logL = -fit$value,
stime = stimeSN[3], npara = length(fit$par), details=fit);
+   if(sncopmle2$logL > sncopmle$logL){ sncopmle <- sncopmle2; }
+
+   ## result ##
+
list(start=dat$start,end=dat$end,nobs=nrow(dat$u),fits=list(astcopmle,gstcopmle,tcopml
e,sncopmle,ncopmle));
+ }
>
> ## GARCH(1,1) filterd return
> res <- mlesttsnn(ugarchspec(mean.model=list(armaOrder=c(0,0))),'2005-04-01::2015-03-
31');
> resTab(res)
                  AC skew-t     GH skew-t     Student-t   skew-Normal        Normal
rho21            0.5675724     0.4827798     0.4869604     0.6244610     0.4841349
rho31            0.4771940     0.3757118     0.3817211     0.5423083     0.3716737
rho32            0.6770061     0.6130115     0.6181123     0.7193701     0.6150727
delta/gamma     -0.5107352    -0.1928586            NA    -0.6506880            NA
nu               9.2967541     9.5198409     9.4340983            NA            NA
logLik         926.4473914   924.7724910   921.5758671   891.6771927   885.6059679
AIC          -1842.8947828 -1839.5449819 -1835.1517342 -1775.3543854 -1765.2119358
BIC          -1814.0478897 -1810.6980889 -1812.0742197 -1752.2768710 -1747.9037999
Time             6.5500000     5.6000000     4.3700000     1.3100000     0.2600000
Conv.            0.0000000     0.0000000     0.0000000     0.0000000     0.0000000
Count          132.0000000    62.0000000   179.0000000   191.0000000   154.0000000
irho21           0.0000000     0.4869604     0.0000000     0.0000000     0.0000000
irho31           0.0000000     0.3817211     0.0000000     0.0000000     0.0000000
irho32           0.0000000     0.6181123     0.0000000     0.0000000     0.0000000
idelta/igamma    0.0000000     0.0000000            NA     0.0000000            NA
inu              6.0000000     9.4340983     6.0000000            NA            NA
```

G. Estimated results with multi-skewness for the AC and GH skew-*t* copulas

This section describes estimated results with multi-skewness for the AC and GH skew-*t* copulas, where each variate has a different skewness parameter while the main document describes those for the equi-skewness case.

G.1. Estimated copulas for unfiltered returns

Table G-1 is the result of estimated parameters of the AC skew-*t*, GH skew-*t*, Student-*t*, skew-Normal, and Normal copulas for unfiltered five-year daily return from April 1, 2010 to March 31, 2015 (N = 1,188). Table G-2 is that for unfiltered ten-year daily return from April 1, 2005 to March 31, 2015 (N = 2,367).

In both Table G-1 and Table G-2, the AC skew-*t* copula attains the lowest AIC (Akaike Information Criterion) among the five copulas and is selected by the AIC. However, the Student-*t* copula attains the lowest BIC (Bayesian Information Criterion) and is selected by the BIC. To ensure the significance of the skewness parameter, we apply likelihood ratio test with the null hypothesis $\delta_1 = \delta_2 = \delta_3 = 0$ using the test statistic of the double of the difference between log-likelihood of the AC skew-*t* copula and that of the Student-*t* copula follows $\chi^2(3)$ under the null hypothesis. In Table G-1, the test statistic is about 11.9, the p-value is 0.0078. In Table G-2, the test statistic is about 17.2, the p-value is 0.0006. In both cases, the skewness parameter $\boldsymbol{\delta}$ is significant at the 1% level.

Table G-1 Estimated parameters for daily return from April 1, 2010 to March 31, 2015

	AC Skew-*t*	GH Skew-*t*	Student-*t*	Skew -Normal	Normal
ρ_{21}	0.535	0.486	0.492	0.413	0.488
ρ_{31}	0.492	0.365	0.376	0.384	0.369
ρ_{32}	0.676	0.654	0.655	0.522	0.644
δ_1, γ_1	–0.520	–0.132		–0.141	
δ_2, γ_2	–0.318	–0.042		0.426	
δ_3, γ_3	–0.600	–0.134		–0.272	
ν	5.977	5.646	6.094		
log-likelihood	529.2	528.3	523.2	483.5	477.6
AIC	**–1044.4**	–1042.6	–1038.5	–954.9	–949.1
BIC	–1008.8	–1007.0	**–1018.2**	–924.5	–933.9

Table G-2 Estimated parameters for daily return from April 1, 2005 to March 31, 2015

	AC Skew-*t*	GH Skew-*t*	Student-*t*	Skew -Normal	Normal
ρ_{21}	0.496	0.489	0.493	0.504	0.494
ρ_{31}	0.411	0.387	0.385	0.351	0.383
ρ_{32}	0.610	0.625	0.624	0.480	0.611
δ_1, γ_1	–0.190	–0.040		0.078	
δ_2, γ_2	–0.074	–0.040		0.375	
δ_3, γ_3	–0.405	–0.102		–0.340	
ν	3.436	3.692	3.623		
log-likelihood	1109.4	1109.1	1100.8	901.7	893.5
AIC	**–2204.9**	–2204.1	–2193.6	–1791.5	–1781.0
BIC	–2164.5	–2163.7	**–2170.6**	–1756.9	–1763.7

G.2. Estimated copulas for standardized residuals

We focus on the ten-year observation period from April 1, 2005 to March 31, 2015 (N = 2,367). Table G-3 is the result of estimated parameters of the AC skew-*t*, GH skew-*t*, Student-*t*, skew-Normal, and Normal copulas for the standardized residuals z_t of GARCH(1,1). Table G-4 is the result of estimated parameters of the four copulas for the standardized residuals z_t of

EGARCH(1,1). For each margin, z_t is assumed to follow the univariate standard Normal distribution.

Both in Table G-3 for the result of GARCH(1,1) and in Table G-4 for the result of EGARCH(1,1), the AC skew-t copula attains the lowest AIC among the five copulas and the Student-t copula attains the lowest BIC. The test statistic of the likelihood ratio test with the null hypothesis $\delta_1 = \delta_2 = \delta_3 = 0$ is about 11.4 in Table G-3, the p-value is 0.0097. The skewness parameter $\boldsymbol{\delta}$ is significant with the 1% level. The test statistic of the likelihood ratio is about 10.9 in Table G-4, the p-value is 0.0125. The skewness parameter $\boldsymbol{\delta}$ is significant at the 5% level.

Table G-3 Estimated parameters for daily standardized residuals of GARCH(1,1) from April 1, 2005 to March 31, 2015

	AC Skew-*t*	GH Skew-*t*	Student-*t*	Skew -Normal	Normal
ρ_{21}	0.516	0.483	0.487	0.502	0.484
ρ_{31}	0.451	0.375	0.382	0.417	0.372
ρ_{32}	0.635	0.616	0.618	0.599	0.615
δ_1, γ_1	–0.343	–0.103		–0.257	
δ_2, γ_2	–0.294	–0.154		–0.212	
δ_3, γ_3	–0.629	–0.291		–0.689	
ν	8.533	9.721	9.434		
log-likelihood	927.3	926.8	921.6	886.9	885.6
AIC	**–1840.6**	–1839.6	–1835.2	–1761.8	–1765.2
BIC	–1800.2	–1799.2	**–1812.1**	–1727.1	–1747.9

Table G-4 Estimated parameters for daily standardized residuals of EGARCH(1,1) from April 1, 2005 to March 31, 2015

	AC Skew-*t*	GH Skew-*t*	Student-*t*	Skew -Normal	Normal
ρ_{21}	0.589	0.474	0.480	0.573	0.478
ρ_{31}	0.484	0.366	0.373	0.419	0.368
ρ_{32}	0.685	0.611	0.621	0.644	0.618
δ_1, γ_1	–0.534	–0.211		–0.490	
δ_2, γ_2	–0.628	–0.288		–0.747	
δ_3, γ_3	–0.499	–0.299		–0.356	
ν	11.036	11.598	12.553		
log-likelihood	912.6	911.5	907.2	885.9	883.0
AIC	**–1811.2**	–1808.9	–1806.3	–1759.8	–1759.9
BIC	–1770.8	–1768.5	**–1783.3**	–1725.2	–1742.6