Maximum likelihood estimation of skew *t*-copula

Toshinao Yoshiba*

Bank of Japan, Chuo-ku, Tokyo 103-8660, Japan
The Institute of Statistical Mathematics, Tachikawa, Tokyo 190-8562, Japan
February 6, 2014

Abstract

We construct a copula from the multivariate skew *t*-distribution of Azzalini and Capitanio (2003). This copula can capture asymmetric and extreme dependence between variables, and it is one of the few that is effective when the number of dimensions is high. However, two problems arise when estimating the parameters by maximum likelihood estimation. Here, we solve these problems and provide a concrete maximum likelihood estimation algorithm. We test our solution by simulating trivariate data with realistic parameters. The parameters are estimated from the daily returns of three stock indices: the SP500, DAX, and Nikkei225.

Keywords: Skew *t*-distribution; Copula; Maximum likelihood estimation

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. In this situation, Student's *t*-copula is frequently used in financial portfolio risk management. However, Student's *t*-copula is restrictive because of its symmetric dependence at both the upper and lower tails. Therefore, we apply the skew *t*-copula to capture the asymmetric dependence of risk factors.

The skew *t*-copula is defined by a multivariate skew *t*-distribution and its marginal distribution. 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*-copula.

To the best of the author's knowledge, three types of skew *t*-copula 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). 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

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

the multivariate skew t-distributions with hidden truncation, the distribution of Azzalini and Capitanio (2003) is more popular than that of Sahu *et al.* (2003). The third type of skew t-copula was proposed by Joe (2006) and is based on the multivariate skew t-distribution of Azzalini and Capitanio (2003). Kollo and Pettere (2010) tried to construct the third type, but they were unsuccessful.

In this study, we construct the skew t-copula of Azzalini and Capitanio (2003) and indicate two problems that arise when estimating the parameters by maximum likelihood estimation (MLE). The first problem is that the log-likelihood function includes univariate skew t-quantile functions, which makes calculating a log-likelihood extremely 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 (SP500, DAX, and Nikkei225), we test our solution by simulating trivariate data with the realistic parameters.

The remainder of the paper is organized as follows. Section 2 introduces the three types of skew t-copula referred to in previous research. Section 3 derives the log-likelihood function of the skew t-copula of Azzalini and Capitanio (2003) and describes the two problems that occur when estimating the parameters by MLE. Section 4 shows sophisticated solutions to these two problems. Then, Section 5 estimates the parameters using MLE for trivariate simulated data after estimating the benchmark parameters. Finally, Section 6 concludes the paper.

2. Three types of skew *t*-copula

This section summarizes the three types of skew t-copula, focusing on the d-variate random vector $\boldsymbol{X}$ representations of the corresponding skew t-distribution. Because we focus on the copulas, we set the location vector to $\boldsymbol{\xi}^{\mathrm{T}} = (\xi_1, \ldots, \xi_d) = (0, \ldots, 0)$ and the scale vector to $\boldsymbol{\sigma}^{\mathrm{T}} = (\sigma_1, \ldots, \sigma_d) = (1, \ldots, 1)$.

2.1. The GH skew *t*-copula

The first skew t-copula is described in the work of Demarta and McNeil (2005). The random variable for this copula has the following representation:

$$\mathbf{X} = \boldsymbol{\gamma} h(V) + V^{-1/2}\mathbf{Z}. \tag{1}$$

In particular, Demarta and McNeil (2005) focus on the following special case:

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

where $V \sim G(\nu/2, \nu/2)$ and $\mathbf{Z} \sim N_d(0, \Sigma)$. Here, $\boldsymbol{\gamma}$ is the *d*-variate skewness parameter vector. If $\boldsymbol{\gamma} = 0$, then the skew *t*-copula reduces to Student's *t*-copula. This copula is based on the multivariate version of the generalized hyperbolic (GH) skew *t*-distribution proposed by Barndorff-Nielsen (1977). Therefore, we refer to this type as the GH skew *t*-copula. 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, to be significant in many cases.

2.2. The skew *t*-copula of Smith *et al.* (2012)

The second skew *t*-copula was constructed by Smith *et al.* (2012). The random variable for this copula has the following representation:

$$\mathbf{X} = V^{-1/2}(\Gamma \,|\, \mathbf{W} \,| + \mathbf{Z}), \tag{3}$$

where $V \sim G(\nu/2, \nu/2)$, $\mathbf{Z} \sim N_d(0, \Sigma)$, and $\mathbf{W} \sim N_d(0, I)$. Here, $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_d)$ is a $d \times d$ diagonal matrix that denotes the skewness parameter. This skew *t*-copula is implied in the skew *t*-distribution proposed by Sahu *et al.* (2003). If $\Gamma = 0$, then the skew *t*-copula reduces to Student's *t*-copula. In equation (3), $\Gamma \,|\, \mathbf{W} \,| + \mathbf{Z}$ is also represented as $\mathbf{Y} \,|\, \mathbf{W} > 0$, where $\mathbf{Y} = \Gamma \mathbf{W} + \mathbf{Z}$. In this sense, the skew *t*-distribution is formed from hidden truncation. Arnold and Beaver (2004) indicate that this method has received considerable attention in the last 20 years as a method of constructing a skew elliptical distribution.

Smith *et al.* (2012) applied this type of copula to regional spot prices in the Australian electricity market and to ordinal exposure measures for 15 major websites. They showed that the skew *t*-copula substantially outperforms the symmetric Student's *t*-copula in both cases. For parameter estimation, they proposed a Bayesian Markov chain Monte Carlo approach rather than using maximum likelihood estimation.

2.3. The skew *t*-copula of Azzalini and Capitanio (2003)

The third type of skew *t*-copula is implied by the *d*-variate skew *t*-distribution of Azzalini and Capitanio (2003). The random variable for this distribution has the following representation:

$$\mathbf{X} = V^{-1/2}\mathbf{Y}, \tag{4}$$

where $V \sim G(\nu/2, \nu/2)$. Here, $\mathbf{Y}$ is the *d*-variate skew normal distributed random vector and has the following representation with a skewness vector of $\boldsymbol{\zeta}^{\mathrm{T}} = (\zeta_1, \ldots, \zeta_d)$:[1]

$$Y_j = \delta_j \mid Z_0 \mid + \sqrt{1-\delta_j^2} Z_j , \tag{5}$$

where $Z_0 \sim N(0,1)$ and $\mathbf{Z} \sim N_d(0, \Psi)$. Here, Ψ is a $d \times d$ correlation matrix. The distribution of $\mathbf{Y}$ is denoted as $\mathbf{Y} \sim \mathrm{SN}_d(\boldsymbol{\zeta}, \Psi)$.[2] The density of $\mathbf{Y}$ is given as

$$f(\mathbf{y}) = 2\phi_d(\mathbf{y}; \Omega)\Phi(\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{y}) , \tag{6}$$

where

$$\boldsymbol{\zeta}^{\mathrm{T}} \equiv (\zeta_1, \ldots, \zeta_d) = (\frac{\delta_1}{\sqrt{1-\delta_1^2}}, \ldots, \frac{\delta_d}{\sqrt{1-\delta_d^2}}) , \tag{7}$$

$$\Delta = \mathrm{diag}(\frac{1}{\sqrt{1+\zeta_1^2}}, \ldots, \frac{1}{\sqrt{1+\zeta_d^2}}) , \tag{8}$$

$$\Omega = \Delta(\Psi + \boldsymbol{\zeta}\boldsymbol{\zeta}^{\mathrm{T}})\Delta , \tag{9}$$

$$\boldsymbol{\alpha}^{\mathrm{T}} = \frac{\boldsymbol{\zeta}^{\mathrm{T}}\Psi^{-1}\Delta^{-1}}{\sqrt{1+\boldsymbol{\zeta}^{\mathrm{T}}\Psi^{-1}\boldsymbol{\zeta}}} . \tag{10}$$

The random vector $\mathbf{X}$ is denoted as $\mathbf{X} \sim \mathrm{St}_d(0, \Omega, \boldsymbol{\alpha}, \nu)$.

The representation of the *d*-variate skew normal random vector $\mathbf{Y}$ in equation (5) is called the "transformation method." Applying the "conditioning method" described in Azzalini and Capitanio (2003), the $\mathbf{Y}$ is represented as

$$\mathbf{Y} = \begin{cases} \mathbf{Z} & \text{if } Z_0 > 0, \\ -\mathbf{Z} & \text{if } Z_0 < 0, \end{cases} \quad \begin{pmatrix} Z_0 \\ \mathbf{Z} \end{pmatrix} \sim N_{d+1}\left\{0, \begin{pmatrix} 1 & \boldsymbol{\delta}^{\mathrm{T}} \\ \boldsymbol{\delta} & \Omega \end{pmatrix}\right\}. \tag{11}$$

As shown in equation (11), the skew *t*-distribution is also formed from hidden truncation, as was the skew *t*-distribution of Sahu *et al.* (2003). In addition, the skew *t*-distribution of Azzalini and Capitanio (2003) is more popular than that of Sahu *et al.* (2003).

We define the extended correlation matrix, *R*, as

$$R = \begin{pmatrix} 1 & \boldsymbol{\delta}^{\mathrm{T}} \\ \boldsymbol{\delta} & \Omega \end{pmatrix}. \tag{12}$$

The marginal skewness parameter, $\boldsymbol{\zeta}^{\mathrm{T}} = (\zeta_1, \ldots, \zeta_d)$, is given in equation (7). Conversely, the skewness parameter, $\boldsymbol{\delta}^{\mathrm{T}} = (\delta_1, \ldots, \delta_d)$, is given as

---

[1] $\mathbf{Y}$ has a common skewness scalar factor, $| Z_0 |$, for its *d* variates, as shown in equation (5). On the other hand, the corresponding random vector in Sahu *et al.* (2003), shown in equation (3), has a different skewness factor, $| W_j |$, for each *j*-th variate.

[2] We adopt the notation of equation (2.7) in Azzalini and Dalla Valle (1996).

$$\delta_j = \frac{\zeta_j}{\sqrt{1+\zeta_j^2}} . \tag{13}$$

The scale vector $\boldsymbol{\sigma}^{\mathrm{T}} = (\sigma_1, \ldots, \sigma_d)$ and the correlation matrix Ω in equation (12) form a variance–covariance matrix, $\tilde{\Omega}$, as follows:

$$\tilde{\Omega} = \mathrm{diag}(\sigma_1, \ldots, \sigma_d)\Omega\,\mathrm{diag}(\sigma_1, \ldots, \sigma_d) . \tag{14}$$

With the location vector $\boldsymbol{\xi}^{\mathrm{T}} = (\xi_1, \ldots, \xi_d)$, $\tilde{\mathbf{X}} = \boldsymbol{\xi} + V^{-1/2}\tilde{\mathbf{Y}}$ has the distribution $\mathrm{St}_d(\boldsymbol{\xi}, \tilde{\Omega}, \tilde{\boldsymbol{\alpha}}, \nu)$, where the d-variate skew normal random vector, $\tilde{\mathbf{Y}}$, is constructed as

$$\tilde{\mathbf{Y}} = \begin{cases} \tilde{\mathbf{Z}} & \text{if } Z_0 > 0, \\ -\tilde{\mathbf{Z}} & \text{if } Z_0 < 0, \end{cases} \quad \begin{pmatrix} Z_0 \\ \tilde{\mathbf{Z}} \end{pmatrix} \sim N_{d+1}\left\{0, \begin{pmatrix} 1 & \boldsymbol{\delta}^{\mathrm{T}} \\ \boldsymbol{\delta} & \tilde{\Omega} \end{pmatrix}\right\}, \tag{15}$$

and

$$\tilde{\boldsymbol{\alpha}} = \frac{\tilde{\Omega}^{-1}\boldsymbol{\delta}}{\sqrt{1-\boldsymbol{\delta}^{\mathrm{T}}\tilde{\Omega}^{-1}\boldsymbol{\delta}}} . \tag{16}$$

Conversely,

$$\boldsymbol{\delta} = \frac{\tilde{\Omega}\tilde{\boldsymbol{\alpha}}}{\sqrt{1+\tilde{\boldsymbol{\alpha}}^{\mathrm{T}}\tilde{\Omega}\tilde{\boldsymbol{\alpha}}}} . \tag{17}$$

3. The construction of the Azzalini and Capitanio (2003) skew *t*-copula

Since the multivariate skew *t*-distribution of Azzalini and Capitanio (2003) is the most popular method of constructing skew-symmetric distributions, this section constructs their skew *t*-copula by specifying the marginal distribution from the *d*-variate skew *t*-distribution. After deriving log-likelihood function, we then indicate the two problems that occur when using MLE to estimate the parameters.

3.1. The specification of the skew *t*-copula

The skew *t*-copula of Azzalini and Capitanio (2003) is given as follows. Their *d*-variate skew *t*-distribution, $\mathrm{St}_d(0, \Omega, \boldsymbol{\alpha}, \nu)$, has the following density function at $\mathbf{x}$:

$$g(\mathbf{x}) = 2t_{d,\nu}(\mathbf{x};\Omega)T_{1,\nu+d}\left(\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{x}\sqrt{\frac{\nu+d}{\mathbf{x}^{\mathrm{T}}\Omega^{-1}\mathbf{x}+\nu}}\right), \tag{18}$$

where $t_{d,\nu}(\mathbf{x};\Omega)$ is the density function of the *d*-variate Student's *t*-distribution. This function has correlation matrix Ω and degrees of freedom, ν, and is specified as:

$$t_{d,\nu}(\mathbf{x};\Omega) = \frac{\Gamma((\nu+d)/2)}{(\pi\nu)^{d/2}\Gamma(\nu/2)\,|\,\Omega\,|^{1/2}}\left[1+\frac{\mathbf{x}^{\mathrm{T}}\Omega^{-1}\mathbf{x}}{\nu}\right]^{-(\nu+d)/2}. \tag{19}$$

$T_{1,\nu}(\cdot)$ is the distribution function of the univariate Student's *t*-distribution, with degrees of freedom ν. As Joe (2006) indicates, the *j*-th marginal distribution is $\mathrm{St}_1(0,1,\zeta_j,\nu)$, with density

$$g_j(x_j) = 2t_{1,\nu}(x_j)T_{1,\nu+1}\left(\zeta_j x_j\sqrt{\frac{\nu+1}{x_j^2+\nu}}\right), \tag{20}$$

where $t_{1,\nu}(x)$ is the density function of the univariate Student's *t*-distribution, with degrees of freedom ν.

Kollo and Pettere (2010) explain that *j*-th marginal distribution of the *d*-variate skew *t*-distribution, $\mathrm{St}_d(0,\Omega,\boldsymbol{\alpha},\nu)$, is $\mathrm{St}_1(0,1,\alpha_j,\nu)$. However, in fact, the *j*-th marginal distribution is $\mathrm{St}_1(0,1,\zeta_j,\nu)$, as shown in equations (5) and (13).

The skew *t*-copula is given by

$$C_{st}(u_1,\ldots,u_d;\Omega,\boldsymbol{\delta},\nu) = \mathrm{St}_d(\mathrm{St}_1^{-1}(u_1;0,1,\zeta_1,\nu),\ldots,\mathrm{St}_1^{-1}(u_d;0,1,\zeta_d,\nu);0,\Omega,\boldsymbol{\alpha},\nu), \tag{21}$$

where $\zeta_1,\ldots,\zeta_d$ are defined in equation (7) and

$$\boldsymbol{\alpha} = \frac{\Omega^{-1}\boldsymbol{\delta}}{\sqrt{1-\boldsymbol{\delta}^{\mathrm{T}}\Omega^{-1}\boldsymbol{\delta}}}. \tag{22}$$

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

If $\boldsymbol{\delta} < 0$, then the lower tail has a higher tail dependence than the upper tail. Fig. 1 confirms this by plotting the contours of the joint densities for Student's *t*-copula and the skew *t*-copula, using standard normal margins.[3]

[3] The values of both the upper and lower tail dependence can be calculated using the formula given in Fung and Seneta (2010).

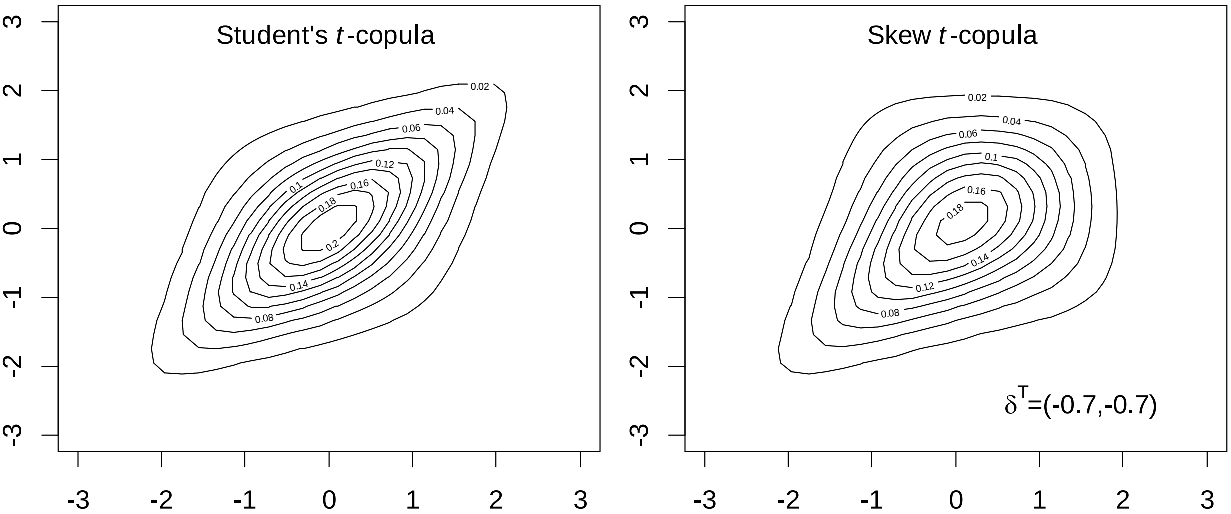

Fig. 1. Contour plot of bivariate distributions using Student's t-copula (left) and the skew t-copula (right) with standard normal margins; the common parameters are ρ =0.5 and ν =3.

3.2. The log-likelihood function

It is assumed that all marginal distributions have been estimated and that N independent observations, $\mathbf{u}_i$, uniformly distributed on $[0,1]^d$, for $i = 1,\ldots, N$, are given by the marginal distribution functions. The set of observations, $\{\mathbf{u}_1,\ldots,\mathbf{u}_N\}$, is called a pseudo sample and can be obtained by applying the estimated marginal distribution function to the original sample.

The log-likelihood function, $l(\theta)$, for $\theta = (\Omega, \boldsymbol{\delta}, \nu)$, is defined by

$$l(\theta) = \sum_{i=1}^{N} \ln\left(\frac{\partial C_{st}(\mathbf{u}_i; \Omega, \boldsymbol{\delta}, \nu)}{\partial \mathbf{u}_i} \right). \tag{23}$$

The partial derivative in equation (23) is calculated as

$$\frac{\partial C_{st}(\mathbf{u}; \Omega, \boldsymbol{\delta}, \nu)}{\partial \mathbf{u}} = \frac{\partial C_{st}(u_1,\ldots,u_d; \Omega, \boldsymbol{\delta}, \nu)}{\partial u_1 \cdots \partial u_d} = \frac{g(\mathbf{x})}{\prod_{j=1}^{d} g_j(x_j)}, \tag{24}$$

using equations (18) and (20), where $\mathbf{x}^{\mathrm{T}} = (x_1,\ldots,x_d)$ and $x_i = \mathrm{St}_1^{-1}(u_i; 0, 1, \zeta_j, \nu)$. Thus, $l(\theta)$ is given as

$$l(\theta) = \sum_{i=1}^{N} l_i(\theta), \tag{25}$$

$$\begin{aligned} l_i(\theta) = {} & \ln\left(\frac{2\Gamma((\nu+d)/2)}{(\pi\nu)^{d/2}\Gamma(\nu/2)} \right) - \frac{1}{2}\ln|\Omega| - \frac{\nu+d}{2}\ln\left(1 + \frac{\mathbf{x}_i^{\mathrm{T}}\Omega^{-1}\mathbf{x}_i}{\nu} \right) \\ & + \ln\left[T_{1,\nu+d}\left(\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{x}_i \sqrt{\frac{\nu+d}{\mathbf{x}_i^{\mathrm{T}}\Omega^{-1}\mathbf{x}_i + \nu}} \right) \right] - \sum_{j=1}^{d} \ln g_j(x_{ij}; \zeta_j, \nu), \end{aligned} \tag{26}$$

where

$$\mathbf{x}_i^{\mathrm{T}} = (x_{i1},\ldots,x_{id}),\, \mathbf{u}_i^{\mathrm{T}} = (u_{i1},\ldots,u_{id}),\, x_{ij} = \mathrm{St}_1^{-1}(u_{ij};0,1,\zeta_j,\nu)\,. \tag{27}$$

The functions $g_j(x_{ij};\zeta_j,\nu)$ in equation (26) are given by equation (20), and $\boldsymbol{\alpha}$ is given by equation (22).

3.3. Problems when estimating parameters using MLE

When maximizing the log-likelihood function, $l(\theta)$, we have two problems. First, the log-likelihood function given in equation (25) includes univariate skew *t*-quantile functions, as shown in equation (27). The quantile function should be applied *Nd* times, which is a time-consuming calculation. The second problem is that the extended correlation matrix, *R*, in equation (12) should be positive semi-definite.

4. Solutions to the MLE problems

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

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

A 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 searched using the Newton method to equate the given probability to the distribution function.

If we use the accurate quantile function, the log-likelihood calculation for some fixed parameters takes more than fifteen seconds on an Intel Core i-7-3520M processor running Microsoft Windows 8 for N = 2,500 trivariate data values. This is time consuming.

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.

One sophisticated way to reduce the calculation time, while maintaining a degree of accuracy, is to use a monotone interpolator with m interpolating points.[4] Let $F(\cdot;\zeta,\nu)$ be the distribution function for the univariate skew t-distribution, $\mathrm{St}_1(0,1,\zeta_j,\nu)$. Note that the j-th variate of the pseudo sample is $u_{1j},\ldots,u_{Nj}$. Let $p_1 = \min_{i=1,\ldots,N} u_{ij}$ and $p_m = \max_{i=1,\ldots,N} u_{ij}$, and calculate $x_1 = F^{-1}(p_1;\zeta_j,\nu)$ and $x_m = F^{-1}(p_m;\zeta_j,\nu)$ using an accurate quantile function. Then, choose $x_k = x_1 + (x_m - x_1)(k-1)/(m-1)$ and calculate $p_k = F(x_k;\zeta_j,\nu)$, for $k = 2,\ldots,m-1$. A monotone interpolator can be used with the table $((x_1,p_1),\cdots,(x_m,p_m))$ to obtain $F(x;\zeta,\nu)$ and $F^{-1}(p;\zeta,\nu)$ for other values in $x \in [x_1,x_m]$ and $p \in [p_1,p_m]$, respectively. 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. The software used here was the sn package, version 0.4-18, released on May 1, 2013. If we use empirical quantiles with K = 100,000 random numbers, the calculation for the log-likelihood is about 60 times faster than the accurate calculation for N = 2,500. On the other hand, the empirical quantiles have a mean absolute error (MAE) of 6.0×10^{-3} from the accurate quantiles. If we use empirical quantiles with K = 1,000,000 random numbers, the calculation for the log-likelihood is twice as fast as the accurate calculation. In this case, the empirical quantiles have an MAE of 1.9×10^{-3} from the accurate quantiles. If we use interpolating quantiles with m = 100, the quantiles have an MAE of 4.3×10^{-5} from the accurate quantiles. This calculation is about 220 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 170 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 quantiles for a log-likelihood using sn version 0.4-18

Method	K or m	N = 2,500			N = 500		
		MAE	Time (sec.)	Speed	MAE	Time (sec.)	Speed
Accurate	–	1.8×10^{-14}	16.42	–	3.3×10^{-15}	2.79	–
Empirical	100,000	6.0×10^{-3}	0.28	59.0	5.9×10^{-3}	0.26	10.8
Empirical	1,000,000	1.9×10^{-3}	3.10	5.3	1.9×10^{-3}	3.00	0.9
Interpolate	100	4.3×10^{-5}	0.07	220.5	1.4×10^{-5}	0.07	40.3
Interpolate	150	1.2×10^{-5}	0.10	169.3	4.2×10^{-6}	0.09	30.9

Note: "Speed" of "empirical" and "interpolate" denote the ratio of the calculation time using the accurate quantiles to that of each method; the "MAE" of "accurate" denotes the mean absolute error $\{F^{-1}(u_{11};\zeta_1,\nu),\ldots,F^{-1}(u_{Nd};\zeta_d,\nu)\}$ from $\{x_{11},\ldots,x_{Nd}\}$, with d = 3. Parameters $\zeta_1,\ldots,\zeta_3$ are given by equation (7) using "true parameters," $\delta_1,\ldots,\delta_3$, from Table 4; $\nu = 5$. Time and MAE are the means of 100 simulated samples.

[4] The idea of using a monotone interpolator belongs to Harry Joe.

Version 1.0-0 of the `sn` package was released on January 6, 2014. Using this version of the package reduces the calculation time when using the accurate quantile function, `qst`, although it seems to be less accurate than version 0.4-18.[5] Table 2 compares the calculation times and accuracy of the empirical quantiles and interpolating quantiles to those of the accurate quantiles, using version 1.0-0 of the `sn` package. As shown, calculating the empirical quantiles with K = 100,000 random numbers is slower than when using the accurate quantiles. However, calculating the interpolating quantiles with m = 150 is still faster than when using the accurate quantiles.

Table 2

Calculation time and accuracy of quantiles for a log-likelihood using `sn` version 1.0-0

Method	K or m	N = 2,500			N = 500		
		MAE	Time (sec.)	Speed	MAE	Time (sec.)	Speed
Accurate	–	3.1×10^{-6}	0.240	–	6.6×10^{-8}	0.068	–
Empirical	100,000	6.0×10^{-3}	0.253	0.95	5.9×10^{-3}	0.302	0.23
Empirical	1,000,000	1.9×10^{-3}	3.038	0.08	1.9×10^{-3}	3.636	0.02
Interpolate	100	4.6×10^{-5}	0.0211	11.4	1.4×10^{-5}	0.0254	2.7
Interpolate	150	1.6×10^{-5}	0.0215	11.2	4.2×10^{-6}	0.0274	2.5

Note that the balance between calculation time and accuracy applies to all three types of skew t-type copulas. As described earlier, Christoffersen *et al.* (2012) use empirical quantiles with K = 100,000 random numbers to specify the GH skew t-copula. 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.

4.2. Positive semi-definiteness for the extended correlation matrix

Since the extended correlation matrix, R, is symmetric and positive semi-definite, the matrix R can be decomposed as

$$R = LL^{\mathrm{T}}, \tag{28}$$

where L is a lower triangular matrix, given as

$$L = \begin{pmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{d+1,1} & l_{d+1,2} & \cdots & l_{d+1,d+1} \end{pmatrix}. \tag{29}$$

Furthermore, the diagonal elements are all one and the non-diagonal elements are in (–1,1), because

[5] The default accuracy of `qst` is 10^{-8} in the probability scale. The quantiles with version 0.4-18 is more accurate than required.

matrix R is a correlation matrix. Thus the elements of the lower triangular matrix, L, can be represented as

$$
\begin{aligned}
l_{ii} &= \begin{cases} 1 & \text{for } i = 1, \\ \left(\prod_{k=1}^{i-2} \sin\theta_{ik}\right)\sin\theta_{i,i-1} & \text{for } i = 2,\ldots,d+1, \end{cases} \\
l_{ij} &= \left(\prod_{k=1}^{j-1} \sin\theta_{ik}\right)\cos\theta_{ij} \quad \text{for } j < i, \text{and } i = 2,\ldots,d+1,
\end{aligned}
\tag{30}
$$

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

$$
\begin{aligned}
(R)_{ii} &= \sum_{j=1}^{i} l_{ij}^2 = \sum_{j=1}^{i-2} l_{ij}^2 + \sin^2\theta_{i1}\cdots\sin^2\theta_{ii-2}(\cos^2\theta_{ii-1} + \sin^2\theta_{ii-1}) \\
&= \sum_{j=1}^{i-3} l_{ij}^2 + \sin^2\theta_{i1}\cdots\sin^2\theta_{ii-3}(\cos^2\theta_{ii-2} + \sin^2\theta_{ii-2}) = \cdots = \cos^2\theta_{i1} + \sin^2\theta_{i1} = 1.
\end{aligned}
\tag{31}
$$

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

Now, the extended correlation matrix, R, is re-parameterized as θ_{ij} for $j = 1,\ldots,i-1$, and $i = 2,\ldots,d+1$ using equation (30). The number of parameters for θ_{ij} is $(d+1)d/2$ for $d \geq 2$.

5. Implementation

Based on the solution to the MLE problems described in the previous section, we now test our solution after estimating benchmark parameters.

5.1. Benchmark parameters

Before conducting the simulation, we set realistic parameters. We estimate the trivariate skew t-distribution for SP500, DAX, and Nikkei225 daily return data over a period of five years from September 2003 to August 2008.[6] The estimated parameters are shown in Table 3.[7]

---

[6] 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.

[7] The estimation is performed using the `mst.mle` function in the `sn` package version 0.4-18 of R. The function estimates the parameters $\boldsymbol{\xi}$, ν, the skewness vector $\boldsymbol{\alpha}$, and the covariance matrix $\tilde{\Omega}$. The parameters $\boldsymbol{\sigma}$, ρ_{ij} in Table 3 are converted from the covariance matrix, $\tilde{\Omega}$. The skewness vectors $\boldsymbol{\delta}$ are given by the estimated skewness vector $\boldsymbol{\alpha}$ and the estimated covariance matrix $\tilde{\Omega}$ using equation

Table 3

Estimated benchmark parameters of a trivariate skew *t*-distribution

ξ_1	ξ_2	ξ_3	σ_1	σ_2	σ_3	
0.004	0.002	0.005	0.011	0.007	0.009	
ρ_{21}	ρ_{31}	ρ_{32}	δ_1	δ_2	δ_3	ν
0.459	0.392	0.546	–0.362	–0.334	–0.481	4.883

The location parameters, ξ_j, and the scale parameters, σ_j, are determined by the marginal distributions. Therefore, we set the correlation parameters, ρ_{ij}, and the skewness parameters, δ_j, to the values given in Table 3, rounded off to the second decimal place, as the true parameter values. We also set the degrees of freedom parameter, ν, to 5 (i.e., rounded off after the decimal point in Table 3).

5.2. Confirmation by simulation

The MLE for the skew *t*-copula can be obtained by maximizing the log-likelihood function in equation (25) 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 (30).[8]

We iterate the above estimation with 100 simulated pseudo samples of trivariate data, with *N* = 500 and 2,500. One of these samples is a pseudo sample, $\{\mathbf{u}_1,\dots,\mathbf{u}_N\}$. The pseudo sample $\{\mathbf{u}_1,\dots,\mathbf{u}_N\}$ is generated from a simulated original sample $\{\mathbf{x}_1,\dots,\mathbf{x}_N\}$ as shown in equation (27). For comparison, we also calculate the MLE of the trivariate skew *t*-distribution for the 100simulated samples in a similar way, assuming location parameters $\xi_j = 0$ and scale parameters $\sigma_j = 1$, for $j = 1,\dots,3$. Table 4 summarizes the results. These calculations were done on an Intel Core i-7-3520M processor running Microsoft Windows 8 and using the `sn` package version 1.0-0.

(17).

[8] See the Appendix for the implementation of MLE using R.

Table 4

Estimated parameters using the proposed method

		N	ρ_{21}	ρ_{31}	ρ_{32}	δ_1	δ_2	δ_3	ν	Time (sec.)
True parameters			0.46	0.39	0.55	–0.36	–0.33	–0.48	5.00	
Copula	mean	500	0.45	0.37	0.54	–0.33	–0.34	–0.45	5.28	43.2
		2,500	0.45	0.38	0.54	–0.34	–0.31	–0.46	5.15	47.8
	standard deviation	500	0.07	0.07	0.06	0.24	0.25	0.26	1.14	18.9
		2,500	0.03	0.04	0.03	0.14	0.13	0.12	0.68	35.9
Distribution	mean	500	0.46	0.39	0.54	–0.36	–0.32	–0.46	5.40	0.18
		2,500	0.45	0.39	0.55	–0.36	–0.33	–0.47	5.22	0.81
	standard deviation	500	0.04	0.05	0.04	0.05	0.06	0.05	1.15	0.06
		2,500	0.03	0.03	0.02	0.03	0.04	0.03	0.98	0.27

Here, we see that the mean of both the correlation parameters, ρ_{ij}, and the skewness parameters, δ_j, are very close to the true parameter values.

We also see that the standard deviations of the skewness parameters, δ_j, are larger than those of the correlation parameters, ρ_{ij}, in the copula parameters estimation, especially for N = 500. The skewness parameters 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 standard deviations of the skewness parameters are large. The standard deviations decrease as the sample size N increases.

Therefore, it has been confirmed that the proposed method works well both in terms of mean and standard deviation.

6. Conclusions

After summarizing the construction of three types of skew *t*-copula, we derive the skew *t*-copula of Azzalini and Capitanio (2003).

When using MLE to estimate parameters, we have indicated two problems that arise. First, practical MLE requires fast and accurate quantile calculations for a univariate skew *t*-distribution. Second, extended correlation matrix should remain positive semi-definite during the estimation process.

In this study, we provided a solution to both problems. We then confirm that the solutions work by simulating a trivariate pseudo sample and estimating the parameter of the skew *t*-copula.

In practical applications of the skew t-copula, estimation comes first. This applies to all three types of copula. After establishing the estimations, we compare the different types of skew t-copula using various data of interest.

Acknowledgements

The author is grateful to Adelchi Azzalini, Hironori Fujisawa, Tsunehiro Ishihara, Harry Joe, Toshiaki Watanabe, and Satoshi Yamashita for their useful comments. The views expressed here are those of the author and do not necessarily reflect the official views of the Bank of Japan. Any errors are the author's own.

References

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

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

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.

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.

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.

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

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.

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

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 codes

This appendix describes the implementation of the MLE (maximum likelihood estimation) of skew *t*-copula parameters using R. It includes the procedure for the MLE of a multivariate skew-*t* distribution, and miscellaneous functions necessary for the MLE of skew-*t* copula.

A.1. Implementation of the MLE for skew-*t* copula

On January 6, 2014, version 1.0-0 of the `sn` package for R was released. This package provides several functions to analyze skew Normal and skew *t*-distributions. Multivariate skew *t*-copula can be estimated by the maximum likelihood method by applying the `sn` library in the following way. Internal parameters are parameterized as $\theta_{21}, \theta_{31}, \ldots, \theta_{d+1,1}$, $\theta_{32}, \ldots, \theta_{d+2,2}, \ldots$, $\theta_{d+1,d}$, $\ln(\nu - 2)$.

```
# skew-t coplua estimation (MLE) using "sn" ver.1.0-0
library(sn)
library(signal)

## redefine qst on "sn" ver.1.0-0 here

## negative log-likelihood for multivariate skew-t copula
## udat[1:n,1:dim] : pseudo sample (N observations for [0,1]^dim)
stcopnll <- function(para, udat=NULL){
  mpoints <- 150;
  dp <- stOrgPara(para);
  delta <- dp$delta;
  zeta <- delta/sqrt(1-delta*delta);
  dim <- length(delta);
  Omega <- diag(dim);
  Omega[upper.tri(Omega)] <- Omega[lower.tri(Omega)] <- dp$rho;
  iOmega <- solve(Omega);
  alpha <- iOmega %*% delta /sqrt(1-(t(delta) %*% iOmega %*% delta)[1,1]);
  nu <- dp$nu;
  ix <- ipqst(udat,zeta,nu,mpoints,rel.tol=1e-6);
## Activate the following line instead of monotone interpolating quantile
## function ipqst() to use accurate quantile function aqst()
## ix <- aqst(udat,zeta,nu,mpoints);
  lm <- matrix(0,nrow=nrow(udat),ncol=dim);
  for(j in 1:dim){ lm[,j] <- dst(ix[,j], alpha=zeta[j], nu=nu, log=TRUE); }
  lc <- dmst(ix,Omega=Omega,alpha=alpha,nu=nu,log=TRUE);
  -sum(lc)+sum(lm)
}


stcop.mle <- function (udat, start = NULL, gr = NULL, ...,
     method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN", "Brent"),
     lower = -Inf, upper = Inf,
     control = list(), hessian = FALSE)
{
  iniPar <- stIntPara(start$rho,start$delta,start$nu);
  method <- match.arg(method);
  fit <- optim(iniPar, stcopnll, method=method, control=control, udat=udat);
  list(call = match.call(), dp = stOrgPara(fit$par), logL = -fit$value,
details=fit, nobs = nrow(udat), method = method);
}


## show estimated parameters and the log-likelihood ##
showResult <- function(fit){
  dp <- fit$dp;
  list(rho=dp$rho,delta=dp$delta,nu=dp$nu,logL=fit$logL);
```

```
}

## setting ##
smN <- 2500;
smrho <- c(0.46,0.39,0.55);
smdelta <- c(-0.36,-0.33,-0.48);
smdf <- 5;

## simulation ##
set.seed(1);
dat <- rstcop(smN,smrho,smdelta,smdf);
dim <- ncol(dat$u);
start <- list(rho=numeric(dim*(dim-1)/2),delta=numeric(dim),nu=6);
## It would be better to add "trace=TRUE" option in control list
system.time(stcopmle<-stcop.mle(dat$u, start=start,
control=list(reltol=1e-4,trace=TRUE)))
showResult(stcopmle);
```

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

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

```
## negative log-likelihood for multivariate skew-t distribution (xi=0,
omega=1)
## xdat[1:n,1:dim] : sample (N observations for [0,1]^dim)
stdistnll <- function(para, xdat=NULL){
  dp <- stOrgPara(para);
  delta <- dp$delta;
  dim <- length(delta);
  Omega <- diag(dim);
  Omega[upper.tri(Omega)] <- Omega[lower.tri(Omega)] <- dp$rho;
  iOmega <- solve(Omega);
  alpha <- iOmega %*% delta /sqrt(1-(t(delta) %*% iOmega %*% delta)[1,1]);
  -sum(dmst(xdat,Omega=Omega,alpha=alpha,nu=dp$nu,log=TRUE));
}


stdist.mle <- function (xdat, start = NULL, gr = NULL, ...,
     method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN", "Brent"),
     lower = -Inf, upper = Inf,
     control = list(), hessian = FALSE)
{
  iniPar <- stIntPara(start$rho,start$delta,start$nu);
  method <- match.arg(method);
  fit <- optim(iniPar, stdistnll, method=method, control=control, xdat=xdat);
  list(call = match.call(), dp = stOrgPara(fit$par), logL = -fit$value,
details=fit, nobs = nrow(xdat), method = method);
}

system.time(stdistmle<-stdist.mle(dat$x, start=start,
control=list(reltol=1e-4)));
showResult(stdistmle);
```

A.3. Random number generator of skew *t*-copula

Random number generator of skew *t*-copula is implemented as the following function. The function returns a list of an original sample $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and the pseudo sample $\{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$.

```
## random number generator of skew t-copula
rstcop <- function(n,rho,delta,nu,...){
  dim <- length(delta);
  zeta <- delta/sqrt(1-delta*delta);
  Omega <- diag(dim);
  Omega[upper.tri(Omega)] <- Omega[lower.tri(Omega)] <- rho;
  iOmega <- solve(Omega);
  alpha <- iOmega %*% delta /sqrt(1-(t(delta) %*% iOmega %*% delta)[1,1]);
  x <- rmst(n=n,Omega=Omega,alpha=alpha,nu=nu);
  u <- matrix(0,nrow=n,ncol=dim);
  for(j in 1:dim){ u[,j] <- pst(x[,j], alpha=zeta[j], nu=nu,...); }
  list(x=x,u=u);
}
```

A.4. Functions for transforming parameters

Functions for transforming the original parameters, ρ_{ij}, $\boldsymbol{\delta}$ and ν to internal parameters $(\boldsymbol{\theta}, \ln(\nu-2))$, and vice versa, are implemented in the following way.

```
## transforming original parameters to internal parameters ##
stIntPara <- function(rho,delta,nu){
  ndim <- length(delta)+1;
  R <- diag(ndim);
  for(i in 2:ndim){
    R[i,1] <- R[1,i] <- delta[i-1];
    if(i>=3){ for(j in 2:(i-1)){ R[i,j] <- R[j,i] <-
rho[i-ndim+(j-1)*(ndim-2-(j-2)/2)]; } }
  }
  LTR <- t(chol(R));
  Mtheta <- matrix(0,nrow=ndim,ncol=ndim);
  for(i in 2:ndim){
    Mtheta[i,1] <- acos(LTR[i,1]);
    cumsin <- sin(Mtheta[i,1]);
    if(i>=3){ for(j in 2:(i-1)){
        Mtheta[i,j] <- acos(LTR[i,j]/cumsin);
        cumsin <- cumsin*sin(Mtheta[i,j]); }
    }
  }
  c(Mtheta[lower.tri(Mtheta)],log(nu-2.0));
}


## transforming internal parameters to original parameters ##
stOrgPara <- function(para){
  ntheta <- length(para)-1;
  theta <- para[1:ntheta];
  ndim <- (1+sqrt(1+8*ntheta))/2;
  LTR <- diag(ndim);
  for(i in 2:ndim){
    LTR[i,1] <- cos(theta[i-1]);
    cumsin <- sin(theta[i-1]);
    if(i>=3){ for(j in 2:(i-1)){
        k <- i+ndim*(j-1)-j*(j+1)/2;
        LTR[i,j] <- cumsin*cos(theta[k]);
        cumsin <- cumsin*sin(theta[k]); }
    }
    LTR[i,i] <- cumsin;
  }
  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);
}
```

A.5. Overwrite `qst` function on `sn` ver.1.0-0

In version 1.0-0 of the `sn` package, the `qst` function should be overwritten as follows to avoid an infinite loop and to obtain the `qt` value at zero skewness ζ (`alpha = 0`).

The default relative tolerance of the numerical integrate function used in the internal `pst` function has lower accuracy than the default tolerance of the `qst` function. Therefore, the `qst` function cannot exit the `while` loop with the default parameters in some cases.

To avoid the infinite loop, we modifies the internal `pst` functions 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`.

To obtain the `qt` value at zero skewness ζ (`alpha = 0`), we define `Sign` function instead of internal `sign` function.

```
## redefine qst on "sn" ver.1.0-0
qst <- function (p, xi = 0, omega = 1, alpha = 0, nu = Inf, tol = 1e-08, maxit
= 30, ...)
{
    if (length(alpha) > 1)
        stop("'alpha' must be a single value")
    if (length(nu) > 1)
        stop("'nu' must be a single value")
    if (nu <= 0)
        stop("nu must be non-negative")
    if (nu == Inf)
        return(qsn(p, xi, omega, alpha))
    if (nu == 1)
        return(qsc(p, xi, omega, alpha))
    if (alpha == Inf)
        return(xi + omega * sqrt(qf(p, 1, nu)))
    if (alpha == -Inf)
        return(xi - omega * sqrt(qf(1 - p, 1, nu)))
    na <- is.na(p) | (p < 0) | (p > 1)
    abs.alpha <- abs(alpha)
    if (alpha < 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] <- pst(xa[nc], 0, 1, abs.alpha, nu, ...) - p[nc]
    fb[nc] <- pst(xb[nc], 0, 1, abs.alpha, 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] <- pst(xc[nc], 0, 1, abs.alpha, 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)
    Sign <- function(x) sign(x)+ as.numeric(x==0)
    q <- as.numeric(xi + omega * Sign(alpha)* x)
    names(q) <- names(p)
    return(q)
}
```

A.6. Three methods to calculate quantiles

Three methods to calculate quantiles for a given pseudo sample $\{\mathbf{u}_1,\ldots,\mathbf{u}_N\}$ are implemented as follows. The first one is an accurate method using modified `qst` function given in A.5. The second one is empirical quantiles with random sampling. The third one uses 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.[9]

```
## accurate quantiles ##
aqst <- function(udat,zeta,nu, ...){
  dim <- ncol(udat);
  ax <- matrix(0,nrow=nrow(udat),ncol=dim);
  for(j in 1:dim){
    ax[,j] <- qst(udat[,j], alpha=zeta[j], nu=nu, ...);
  }
  ax
}


## empirical quantiles with random sampling ##
rsqst <- function(udat,zeta,nu,simNum){
  dim <- ncol(udat);
  sx <- matrix(0,nrow= nrow(udat),ncol=dim);
  sy <- matrix(0,nrow=simNum,ncol=dim);
  for(j in 1:dim){
     sy[,j] <- sort(rst(simNum, alpha=zeta[j], nu=nu));
     sx[,j] <- sy[udat[,j]*(simNum-1)+1,j];
  }
  sx
}


## interpolating quantiles ##
ipqst <- function(udat,zeta,nu,mpoints, ...){
  dim <- ncol(udat);
  ix <- matrix(0,nrow=nrow(udat),ncol=dim);
  for(j in 1:dim){
    minx <- qst(min(udat[,j]), alpha=zeta[j], nu=nu, ...);
    maxx <- qst(max(udat[,j]), alpha=zeta[j], nu=nu, ...);
```

[9] 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 $x_1 < \cdots < x_m$.

```
    xx <- seq(minx,maxx,length.out=mpoints);
    px <- sort(pst(xx, alpha=zeta[j], nu=nu, ...));
    ix[,j] <- pchip(px, xx, udat[,j]);
  }
  ix
}
```

A.7. Example of execution

With the previous functions in a file, "stcopula.R", an example of executing the previous codes and the correspondence from R is given as follows.

```
> ## Please specify the directory where "stcopula.R" is located by setwd()
> ## read R codes from a file ##
> source("stcopula.R");
> ## setting ##
> smN <- 2500;
> smrho <- c(0.46,0.39,0.55);
> smdelta <- c(-0.36,-0.33,-0.48);
> smdf <- 5;
> ## simulation ##
> set.seed(1);
> dat <- rstcop(smN,smrho,smdelta,smdf);
> dim <- ncol(dat$u);
> start <- list(rho=numeric(dim*(dim-1)/2),delta=numeric(dim),nu=6);
> ## It would be better to add "trace=TRUE" option in control list
> system.time(stcopmle<-stcop.mle(dat$u, start=start,
control=list(reltol=1e-4)));
   user  system elapsed
  46.42    0.00   46.42

> showResult(stcopmle);
$rho
[1] 0.4331891 0.3846318 0.4862471

$delta
[1] -0.3292710 -0.2045169 -0.4915625

$nu
[1] 4.741675

$logL
[1] 686.4662

>
```