0.1 Other sampling algorithms

0.1.1 Conditional sampling

Algorithm 1. Conditional sampling

Require: Conditional copula distribution $C_{V|U}(v|u)$

1: Generate two independent variates $u, w \sim \mathcal{U}(0,1)$

2: Keep the first variate u as is

3: Transform the second variate w into v using the inverse of the conditional distribution:

$$v = C_{V|U}^{-1}(w|u)$$

Ensure: The resulting pair (u, v) will follow the desired copula distribution

We applied this method to sample from the Clayton copula, whose conditional distribution is given by:

$$C_{V|U}(v|u) = \frac{\partial}{\partial u}(u^{-\theta} + v^{-\theta} - 1)^{-1/\theta} = u^{-(1+\theta)}(u^{-\theta} + v^{-\theta} - 1)^{-(1+\theta)/\theta}, \tag{1}$$

and thus, setting (1) equal to the fixed probability q and solving for v delivers

$$v = C_{V|U}^{-1}(q|u) = \left[1 + u^{-\theta}(q^{-\theta/(1+\theta)} - 1)\right]^{-1/\theta}.$$

For more details, check p.275 in **?**.

0.1.2 Theorem 4.3.7 in ?

From Theorem 4.3.7 in **?** (p.129), it can be shown that Algorithm 2. generates random variates (u, v) whose joint distribution function is an Archimedean copula C with generator φ:

Algorithm 2. Sampling from Archimedean Copulas using Theorem 4.3.7 in **?**

Require: Generator function φ of the desired Archimedean C-copula

1: Generate two independent variates $s, t \sim \mathcal{U}(0,1)$

2: Set $w \leftarrow K_C^{(-1)}(t)$, where $K_C(w) = t - \dfrac{\varphi(w)}{\varphi'(w^+)}$

3: Set $u \leftarrow \varphi^{[-1]}(s\varphi(w))$ and $v \leftarrow \varphi^{[-1]}((1-s)\varphi(w))$

Ensure: The pair (u, v) will follow the desired copula distribution

Note that K_C is given by Theorem 4.3.4. (p.127) and it denotes the C-measure of the set $\{(u,v) \in [0,1]^2 \mid C(u,v) \leq w\}$

$$K_C(w) := t - \frac{\varphi(w)}{\varphi'(w^+)},$$

and $\varphi'(w^+)$ denotes the right-sided derivative of the generator.

In our case, we use this algorithm to sample from Gumbel, Joe and N14. Their C-measures are given below. Note that there is no analytical solution for their inverse, so we have to resort to numerical inversion.

$$\textit{Gumbel} \qquad K_C(w) = w \cdot \left[1 - \frac{\log(w)}{\theta}\right]$$

$$\textit{Joe} \qquad K_C(w) = w - \frac{1}{\theta} \cdot \frac{\log[1-(1-w)^\theta] \cdot [1-(1-w)^\theta]}{(1-w)^{\theta-1}}$$

$$\textit{N14} \qquad K_C(w) = -w \cdot (-2 + w^{1/\theta})$$

0.1.3 Gaussian Copula

Algorithm 3. Calibrating Gaussian Copula

Require: Pseudo-observations from training data: $\mathbf{u}, \mathbf{v} \in [0,1]^{T^{tr}}$

1: Transform $\mathbf{u}$ and $\mathbf{v}$ into standard normal variates $\mathbf{x} = \Phi^{-1}(\mathbf{u})$ and $\mathbf{y} = \Phi^{-1}(\mathbf{v})$

2: Obtain the empirical covariance matrix

$$\widehat{\boldsymbol{\Sigma}} := \begin{bmatrix} \hat{\sigma}_x^2 & \hat{\sigma}_{yx} \\ \hat{\sigma}_{xy} & \hat{\sigma}_y^2 \end{bmatrix} = \frac{1}{T_{tr}-1} \begin{bmatrix} \mathbf{x}^\top \mathbf{x} & \mathbf{x}^\top \mathbf{y} \\ \mathbf{y}^\top \mathbf{x} & \mathbf{y}^\top \mathbf{y} \end{bmatrix}$$

3: Set $\hat{\rho} \leftarrow \dfrac{\hat{\sigma}_{xy}}{\hat{\sigma}_x \hat{\sigma}_y}$

Ensure: $\hat{\rho}$

Algorithm 4. Sampling from Gaussian Copula

Require: $\hat{\boldsymbol{\Sigma}}$ from section 0.1.3

1: Generate correlated Gaussian pairs $(x, y) \sim \mathcal{N}(\mathbf{0}, \widehat{\boldsymbol{\Sigma}})$

2: Transform the Gaussian pairs into uniform variates $u = \Phi(x)$ and $v = \Phi(y)$.

Ensure: The pair (u, v) follows the Gaussian copula distribution.

0.1.4 Student-t Copula

Algorithm 5. Calibrating the Student-t Copula

Require: Pseudo-observations from training data: $\mathbf{u}, \mathbf{v} \in [0,1]^{T^{tr}}$

1: **for** each $\nu \in \mathcal{V} := [1, 15]$ **do**

2: Transform $\mathbf{u}$ and $\mathbf{v}$ into Student-t variates: $\mathbf{x}_\nu = t_\nu^{-1}(\mathbf{u})$; $\mathbf{y}_\nu = t_\nu^{-1}(\mathbf{v})$

3: Obtain the empirical covariance matrix

$$\widehat{\mathbf{\Sigma}}(\nu) := \begin{bmatrix} \hat{\sigma}_x^2(\nu) & \hat{\sigma}_{yx}(\nu) \\ \hat{\sigma}_{xy}(\nu) & \hat{\sigma}_y^2(\nu) \end{bmatrix} = \frac{1}{T_{tr}-1} \begin{bmatrix} \mathbf{x}_\nu^\top \mathbf{x}_\nu & \mathbf{x}_\nu^\top \mathbf{y}_\nu \\ \mathbf{y}_\nu^\top \mathbf{x}_\nu & \mathbf{y}_\nu^\top \mathbf{y}_\nu \end{bmatrix}$$

4: Evalutate the log-likelihood of the t-copula

$$\ell(\nu; \mathbf{u}, \mathbf{v}) := \sum_{t \in \mathcal{T}_{tr}} \log c(u_t, v_t; \nu, \hat{\rho}(\nu)) \quad \text{where} \quad \hat{\rho}(\nu) = \frac{\hat{\sigma}_{xy}(\nu)}{\hat{\sigma}_x(\nu)\hat{\sigma}_y(\nu)}$$

5: **end for**

6: Set $\nu^\star \leftarrow \arg\max_{\nu \in \mathcal{V}} \ell(\nu; \mathbf{u}, \mathbf{v})$, and $\hat{\rho}^\star \leftarrow \hat{\rho}(\nu^\star)$

Ensure: $(\nu^\star, \hat{\rho}^\star)$

Algorithm 6. Sampling from the calibrated Student-t Copula

Require: $\nu^\star$ and $\widehat{\mathbf{\Sigma}}(\nu^\star)$ from section 0.1.4

1: Sample from a bivariate normal: $(x_1, x_2) \sim \mathcal{N}(\mathbf{0}, \widehat{\mathbf{\Sigma}}(\nu^\star))$

2: Sample from a chi-square distribution with $\nu^\star$ degrees of freedom: $\chi \sim \chi^2_{\nu^\star}$

3: Compute the Student-t variates: $y_1 = x_1 / \sqrt{\chi/\nu^\star}$ and $y_2 = x_2 / \sqrt{\chi/\nu^\star}$

4: Transform the Student-t variates into uniform variates $u = t_\nu(y_1)$ and $v = t_\nu(y_2)$

Ensure: The pair (u, v) follows the Student-t copula distribution