
Simulation of High-Dimensional t-Student Copulas with a Given Block Correlation Matrix

Gerard Torrent-Gironella · Josep Fortiana

Abstract A computationally efficient method is presented for the simulation of high dimensional t-Student copulas given a block correlation matrix. For such matrices, Cholesky decomposition, product of the Cholesky matrix by a vector, and inverse of the Cholesky matrix have been adapted to reduce the number of operations and memory requirements. The condition number for this family of matrices has been found to determine the propagation of errors and the impact of dimensionality in the method stability. We show that the copula parameters can be estimated by the method of maximum likelihood using the adapted algorithms. An approximation of the t-Student Spearman's rank correlation is also presented that enables the copula construction directly from the given correlation matrix. The proposed simulation algorithm allows us to determine the loss distribution of a credit portfolio composed of thousands of obligors using Monte Carlo simulation of obligors default times.

Keywords t-Student Copula · Block Matrices · Correlated Random Numbers · Cholesky Decomposition · Spearman's Rank Correlation · Maximum Likelihood · Simulation of High-dimensional Multivariate Distributions

Gerard Torrent-Gironella
Tel.: +34-606.179.518
E-mail: gtorrent@ccruncher.net

Josep Fortiana
Dpt. de Probabilitat, Lògica i Estadística, Universitat de Barcelona
Gran Via de les Corts Catalanes, 585, 08007 Barcelona, Spain
Tel.: +34-934.021.654
E-mail: fortiana@ub.edu

1 Introduction

The aim of this work is to provide an algorithm for simulating a large number of correlated random variables satisfying a given block correlation matrix. An application is the simulation of the obligors default times in a credit portfolio to determine loan loss distribution by using the Monte Carlo method (Li 2000). In this case, the default times are modelled as a multivariate random variable where each marginal represents an obligor default time, and the correlation matrix measures the dependence between obligors due to economic sector membership.

Sklar's theorem (Sklar 1959) states that any multivariate distribution with continuous marginals can be decomposed into the marginals and a copula that reflects the structure of dependence between them. It also follows that the multivariate random variable can be simulated using the simulation of the copula and the marginal distributions. Consequently, the simulation of correlated random variables reduces to the simulation of copulas.

In general, correlations do not uniquely reflect the dependence structure of a joint distribution. Elliptical copulas, however, are determined by their correlation matrix (Embrechts et al 1999). This family has another property that makes it desirable for the simulation - symmetry (Demarta and McNeil 2005). We can use this feature to apply the antithetic method as a variance reduction technique in Monte Carlo simulations.

The t-Student copula is an elliptical copula and has a known algorithm for generating random values (Embrechts et al 2001). In the following sections, has been adapted this algorithm to operate in higher dimensions (e.g. 60000 marginals) provided that the correlation matrix is a block matrix.

There are several methods for estimating the parameters of a copula, being the maximum likelihood one of the most used. We see that this method can be used when the correlation matrix has the given block structure.

The correlation matrix used to define an elliptical copula differs from that of the resulting multivariate distribution. We have been determined the transformation to be applied to the correlation matrix such that the simulated copula has the given correlations. This transformation does not have an analytical expression, and it is therefore approximated by numerical methods.

The content is organized into two sections. The first one defines the block matrices, modifies the Cholesky algorithm to operate with them, establishes their eigenvalues to determine the stability of the decomposition, and gives an algorithm to compute the inverse of the Cholesky matrix. The second section provides a brief introduction to copulas, calculates a numerical approximation of the bivariate t-Student Spearman's rank correlation, check the possibility to do parameter estimation using the maximum likelihood method, and exposes the algorithm to simulate a high-dimensional t-Student copula with a given block correlation matrix.

2 Symmetric block matrices

The simulation of a multivariate t-Student distribution requires the Cholesky decomposition of the correlation matrix and, for each draw, the multiplication of this matrix by a vector. When the number of marginals is very large, the simulation becomes computationally intractable for two reasons. The first problem, is the amount of memory required to store the Cholesky matrix. The second issue is the number of operations involving both the Cholesky decomposition and the multiplication of the Cholesky matrix by a vector. For example, if you want to simulate 50000 correlated marginals using the standard method, the Cholesky matrix holds 9.3 GB of RAM, and the product of this matrix by a vector involves 1250025000 multiplications.

When the number of marginals increases, the number of correlations to estimate becomes intractable, being of the order of the square of the number of marginals. Another problem that arises is the lack of data to compute them. In these cases, it is acceptable to assume that there are k groups of marginals such that the elements inherit the correlations of their group. This assumption dramatically reduces the number of values to be estimated and makes the correlation matrix have a structure composed of blocks as defined below.

Definition 1 We say that a matrix A is a matrix of type $T_k(n, d, M)$ where

- $n = (n_1, \dots, n_k)^\top$ with $n_i \in \mathbb{N}$ and $1 \leq n_i$ (number of elements per block),
- $d = (d_1, \dots, d_k)^\top$ with $d_i \in \mathbb{R}$ (diagonal block values), and
- M is a $k \times k$ symmetric matrix with values $m_{ij} \in \mathbb{R}$ (block values).

when

- $a_{ii} = d_r$ with $i \in I_r$, and
- $a_{ij} = m_{rs}$ with $i \in I_r$ and $j \in I_s$.

To simplify the exposition, we use the following notations:

- $n = \sum_{t=1}^k n_t$ (size).
- $I_r = [\sum_{t=1}^{r-1} n_t + 1, \sum_{t=1}^r n_t]$ (block interval).
- $\sum_{n_r} x_t = \sum_{t=n_1+\dots+n_{r-1}+1}^{n_1+\dots+n_r} x_t$ (block sum).

The above definition includes the block correlation matrices and allows block matrices with absolute values above 1 and diagonal values that are distinct from 1. This is because the results stated below are valid for this wider class of matrices.

Example 1 We simulate the default times of 6 obligors. The first three belong to the banking sector, the next two belong to the energy sector and the last one belongs to the services sector. The correlations are determined by sectors, where m_{ij} is the correlation between two obligors belonging to sectors i and j .

$$T_3 \left(\left(\begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0.5 & 0.2 & 0.1 \\ 0.2 & 0.4 & 0.15 \\ 0.1 & 0.15 & 0.3 \end{pmatrix} \right) = \left(\begin{array}{ccc|cc|c} 1 & 0.5 & 0.5 & 0.2 & 0.2 & 0.1 \\ 0.5 & 1 & 0.5 & 0.2 & 0.2 & 0.1 \\ 0.5 & 0.5 & 1 & 0.2 & 0.2 & 0.1 \\ \hline 0.2 & 0.2 & 0.2 & 1 & 0.4 & 0.15 \\ 0.2 & 0.2 & 0.2 & 0.4 & 1 & 0.15 \\ \hline 0.1 & 0.1 & 0.1 & 0.15 & 0.15 & 1 \end{array} \right)$$

2.1 Eigenvalues and eigenvectors

The next statement gives the eigenvalues of a matrix of type T_k . This result is important because it allows us know when the block matrix is positive-definite, and it allows us to determine the stability of the Cholesky decomposition.

Proposition 1 *Let $A = T_k(n, d, M)$ a non-singular matrix, and let G be the $k \times k$ deflated matrix*

$$G = \begin{pmatrix} d_1 + (n_1 - 1) \cdot m_{11} & n_2 \cdot m_{12} & \dots & n_k \cdot m_{1k} \\ n_1 \cdot m_{21} & d_2 + (n_2 - 1) \cdot m_{22} & \dots & n_k \cdot m_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ n_1 \cdot m_{k1} & n_2 \cdot m_{k2} & \dots & d_k + (n_k - 1) \cdot m_{kk} \end{pmatrix}$$

Then, the eigenvalues of A are as follows:

- $d_r - m_{rr}$ with multiplicity $n_r - 1$ for $r = 1, \dots, k$.
- λ_r , the eigenvalues of G with multiplicity 1.

Their corresponding eigenvectors are as follows:

- $v^{rj} = \left(\boxed{0, \dots, 0}_1, \dots, \boxed{0, \dots, 0, +1, 0, \dots, 0, -1}_r, \dots, \boxed{0, \dots, 0}_k \right)^\top$,
where -1 is located at position $\sum_{t=1}^r n_t$, and $+1$ is located at position $(\sum_{t=1}^r n_t) - j$
with $j = 1, \dots, n_r - 1$.
- $w^r = \left(\boxed{x_1^r, \dots, x_1^r}_1, \dots, \boxed{x_k^r, \dots, x_k^r}_k \right)^\top$ where $(x_1^r, \dots, x_k^r)^\top$ is the eigenvector of G with eigenvalue λ_r .

Proof We will verify that the vector v^{rj} is an eigenvector of A with eigenvalue $d_r - m_{rr}$ where $r = 1, \dots, k$ and $j = 1, \dots, n_r - 1$. First, we see that the eigenvalue is different from 0 and then see that the vectors v^{rj} satisfy the conditions of an eigenvector.

For every $r = 1, \dots, k$ with $n_r > 1$, it holds that $m_{rr} \neq d_r$. If there is an r such that $m_{rr} = d_r$, then the columns of A with index $\sum_{t=1}^r n_t$ and index $\sum_{t=1}^r n_t - 1$ would be equal, leading to a singular matrix, a fact that contradicts the statement.

Decompose the matrix A into the sum of a block matrix B with values m_{ij} and a diagonal matrix D with values $d_i - m_{ii}$ such that $A = B + D$. Then, verify that the vectors v^{rj} satisfy the eigenvector condition

$$A \cdot v^{rj} = (B + D) \cdot v^{rj} = B \cdot v^{rj} + D \cdot v^{rj} = 0 + (d_r - m_{rr}) \cdot v^{rj}$$

We note that the eigenvectors v^{rj} are linearly independent and that the subspace generated by them are the vectors $x = (x_1, \dots, x_n)^\top$, where $\sum_{n_r} x_t = 0$ for all $r = 1, \dots, k$. We will prove that w^r are eigenvectors of A with eigenvalues λ_r . First, we see that the matrix G is non-singular, and then, we see that the eigenvalues λ_r are the eigenvalues of G and A .

Eigenvalues and eigenvectors satisfy the following $n_1 + \dots + n_k$ equations:

$$\left. \begin{array}{l} [m_{11} \cdot \sum_{n_1} v_t + \cdots + m_{1k} \cdot \sum_{n_k} v_t] + (d_1 - m_{11}) \cdot v_1 = \lambda \cdot v_1 \\ \vdots \\ [m_{k1} \cdot \sum_{n_1} v_t + \cdots + m_{kk} \cdot \sum_{n_k} v_t] + (d_k - m_{kk}) \cdot v_{n_1+\dots+n_k} = \lambda \cdot v_{n_1+\dots+n_k} \end{array} \right\}$$

For each block, add up their equations to obtain the following k equations:

$$\left. \begin{array}{l} n_1 \cdot [m_{11} \cdot \sum_{n_1} v_t + \cdots + m_{1k} \cdot \sum_{n_k} v_t] + (d_1 - m_{11}) \cdot \sum_{n_1} v_t = \lambda \cdot \sum_{n_1} v_t \\ \vdots \\ n_k \cdot [m_{k1} \cdot \sum_{n_1} v_t + \cdots + m_{kk} \cdot \sum_{n_k} v_t] + (d_k - m_{kk}) \cdot \sum_{n_k} v_t = \lambda \cdot \sum_{n_k} v_t \end{array} \right\}$$

This set of equations can be written in matrix form as $G^\top x = \lambda x$, where $x = (\sum_{n_1} v_t, \dots, \sum_{n_k} v_t)^\top$. Observe that the $n_1 + \dots + n_k - k$ eigenvectors of the form v^r are trivial solutions of this linear system ($G^\top \mathbf{0} = \mathbf{0}$). As matrix A is non-singular, the remaining eigenvectors are nontrivial solutions of this linear system. Thus, $\det(G^\top) \neq 0$.

We assume that the remaining eigenvectors have the form w^r where the components of each block are identical. We write the linear system that must be met in this case, which consists of k equations:

$$\left. \begin{array}{l} [m_{11} \cdot n_1 \cdot x_1 + \cdots + m_{1k} \cdot n_k \cdot x_k] + (d_1 - m_{11}) \cdot x_1 = \lambda \cdot x_1 \\ \vdots \\ [m_{k1} \cdot n_1 \cdot x_1 + \cdots + m_{kk} \cdot n_k \cdot x_k] + (d_k - m_{kk}) \cdot x_k = \lambda \cdot x_k \end{array} \right\}$$

This set of equations can be written in matrix form as $Gw = \lambda w$, and it has a unique solution because $\det(G) = \det(G^\top) \neq 0$. Therefore, the assumption about the shape of the eigenvectors being w^r is correct, and its eigenvectors and eigenvalues are determined by the deflated matrix G . □

Corollary 1 A matrix $A = T_k(n, d, M)$ is definite-positive if

- $d_r > m_{rr}$ for all $r = 1, \dots, k$, and
- the deflated matrix G is definite-positive.

Corollary 2 The determinant of $A = T_k(n, d, M)$ is

$$\det(A) = \prod_{i=1}^k \lambda_i \cdot (n_i - 1) \cdot (d_i - m_{ii})$$

2.2 The Cholesky decomposition

The Cholesky matrix of a T_k matrix is not a block matrix; rather, it has regularities. Its columns are composed of blocks, except on the diagonal. This observation is important because it justifies the adaptation of the Cholesky algorithm to reduce memory size and the number of operations involved in its computation.

Proposition 2 *The Cholesky matrix, L , of a positive-definite symmetric matrix $A = T_k(n, d, M)$ has the following property:*

$$l_{pj} = l_{qj} \text{ if } j < p \text{ and } j < q \text{ and } p, q \in I_r$$

Proof We apply the Cholesky algorithm (Burden and Faires 1998) to matrix A and we prove the proposition for each step.

1. Set $l_{11} = \sqrt{m_{11}}$.
The property does not apply because it is an element of the diagonal.
2. For $j = 2, \dots, n_1 + \dots + n_k$, set $l_{j1} = a_{j1}/l_{11}$.
Consider that $i, j \in I_r$; then,
 $l_{i1} = a_{i1}/l_{11} = m_{r1}/l_{11} = a_{j1}/l_{11} = l_{j1}$.
3. For $i = 2, \dots, n-1$, do steps 4 and 5.
In this step, there is nothing to prove.
4. Set $l_{ii} = \sqrt{a_{ii} - \sum_{t=1}^{i-1} l_{it}^2}$.
The property does not apply because it is an element of the diagonal.
5. For $j = i+1, \dots, n$, set $l_{ji} = \frac{1}{l_{ii}} \cdot [a_{ji} - \sum_{t=1}^{i-1} l_{jt} \cdot l_{it}]$.
We prove that the statement is true if this is true for the previous columns. Consider that $p, q \in I_r$ where $i < p$ and $i < q$, then $l_{pi} = \frac{1}{l_{ii}} \cdot [a_{pi} - \sum_{t=1}^{i-1} l_{pt} \cdot l_{it}] = \frac{1}{l_{ii}} \cdot [m_r - \sum_{t=1}^{i-1} l_{pt} \cdot l_{it}] = \frac{1}{l_{ii}} \cdot [a_{qi} - \sum_{t=1}^{i-1} l_{qt} \cdot l_{it}] = l_{qi}$. As the statement is true for column 1 (see step 2), then it is true for column i .
6. Set $l_{nn} = \sqrt{a_{nn} - \sum_{t=1}^{n-1} l_{nt}^2}$, where $n = n_1 + \dots, n_k$.
The property does not apply because it is an element of the diagonal. □

We use Proposition 2 to adapt the Cholesky algorithm to operate with the block matrices to calculate only the non-repeated elements and to save memory. The adapted algorithm is detailed in Algorithm 1. The computed values that are distinct from the diagonal values are stored in a matrix of size $k \times n$, and the diagonal elements are stored in a vector of size n .

Memory requirements. The adapted algorithm reduces the memory needed to store L from an order of n^2 to an order of $(k+1) \times n$. In the standard algorithm, the matrix L takes $\dim(L) = n \times (n+1)/2$, while in the adapted algorithm, the matrix L takes $\dim(H) + \dim(v) = k \times n + n$.

Number of operations. The adapted algorithm reduces the number of operations required to perform the Cholesky decomposition. Compared with the standard algorithm (Burden and Faires 1998), it reduces the number of operations from order n^3 to order n^2 , as showed in Table 1. The number of operations has been determined by placing a counter for additions/subtractions and a counter for multiplications/divisions for each element of the matrix H and for each diagonal element of v . Recurrence has been inferred from an analysis of a case.

Algorithm 1 Cholesky decomposition to factorize $A = T_k(n, d, M)$ definite-positive as LL^\top , where L is a lower triangular matrix.

Input:

- Numer of blocks, k .
- Number of elements per block, n_i where $i = 1, \dots, k$.
- Diagonal block values, d_i where $i = 1, \dots, k$.
- Block values, m_{rs} where $r, s = 1, \dots, k$.
- Matrix H with size $k \times (n_1 + \dots + n_k)$.
- Vector v with size $n_1 + \dots + n_k$.

```

 $v_1 = \sqrt{d_1}$ 
for  $r = 1$  to  $k$  do
     $H_{r1} = m_{r1}/v_1$ 
end for
 $i = 0$ 
for  $r = 1$  to  $k$  do
    for  $q = 1$  to  $n_r$  do
         $i = i + 1$ 
        if  $i \neq 1$  then
            if  $q = 1$  then
                 $\text{sum} = \sum_{t=1}^{i-1} H_{rt}^2$ 
            else
                 $\text{sum} = \text{sum} + H_{r,i-1}^2$ 
            end if
             $v_i = \sqrt{d_r - \text{sum}}$ 
            for  $s = r$  to  $k$  do
                if  $s \neq r$  OR  $q \neq n_r$  then
                     $H_{si} = \frac{1}{v_i} \cdot (m_{sr} - \sum_{t=1}^{i-1} H_{st} \cdot H_{rt})$ 
                end if
            end for
        end if
    end for
end if
end for
end for

```

Output:

- Matrix H
 - Vector v
 - $L_{ij} = \begin{cases} 0 & \text{if } i < j \\ v_j & \text{if } i = j \\ H_{rj} & \text{if } i > j \text{ where } i \in I_r \end{cases}$
-

Stability. If the matrix A is ill-conditioned, the estimated Cholesky matrix may differ from the exact Cholesky matrix due to representation errors and rounding errors. As indicated in (Higham 2002), the order of magnitude of the error is

$$\frac{|L - \tilde{L}|}{|L|} = O(\kappa(A)\varepsilon)$$

The matrix A is normal ($A^*A = AA^*$) because it is symmetrical; hence, its condition number is $\kappa(A) = |\lambda_{\max}(A)/\lambda_{\min}(A)|$, where λ_{\max} and λ_{\min} are the eigenvalues of A with maximum and minimum module. According to the definition of the 2-norm, $|L|_2 = \sqrt{|\lambda_{\max}(L^*L)|} = \sqrt{|\lambda_{\max}(A)|}$. So,

$$\frac{|L - \tilde{L}|}{|L|} = \frac{|L - \tilde{L}|}{\sqrt{|\lambda_{\max}(A)|}} = O\left(\left|\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}\right| \cdot \varepsilon\right)$$

We can determine when the Cholesky decomposition is stable numerically using the eigenvalues of A , which can be calculated easily using the Proposition 1.

Example 2 We compute the Cholesky decomposition for the matrix of example 1 using the standard algorithm (L) and the adapted algorithm (H, v).

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.8660 & 0 & 0 & 0 & 0 \\ 0.5 & 0.2887 & 0.8165 & 0 & 0 & 0 \\ 0.2 & 0.1155 & 0.0817 & 0.9695 & 0 & 0 \\ 0.2 & 0.1155 & 0.0817 & 0.3507 & 0.9039 & 0 \\ 0.1 & 0.0577 & 0.0408 & 0.1238 & 0.0847 & 0.9811 \end{pmatrix}$$

$$H = \begin{pmatrix} 0.5 & 0.2887 & 0 & 0 & 0 & 0 \\ 0.2 & 0.1155 & 0.0817 & 0.3507 & 0 & 0 \\ 0.1 & 0.0577 & 0.0408 & 0.1238 & 0.0847 & 0 \end{pmatrix}$$

$$v = (1, 0.8660, 0.8165, 0.9695, 0.9039, 0.9811)$$

Example 3 We consider example 1, and we increase the number of marginals such that $n = (30000, 20000, 10000)^\top$. In this case, the memory required to store the matrix L using the adapted algorithm is 4×60000 compared with the 60000×60000 required by the standard algorithm. We observe that in spite of increasing the matrix size, the numerical stability of the decomposition is maintained.

$$\text{eig}(A) = \{0.5_{(29999)}, 0.6_{(19999)}, 0.7_{(9999)}, 17946, 5834.4, 2221.5\}$$

These eigenvalues give a condition number $\kappa(A) = 17946/0.5 = 35892$. This condition number is low enough to use values of type double and gives an accuracy of 10 decimals or more. For the same reason, it is not advisable to use values of type float.

2.3 Product of the Cholesky matrix by a vector

Algorithm 2 present an algorithm for multiplying the Cholesky matrix of an array of type T_k by a vector that takes advantage of repeated values to reduce the number of operations.

Number of operations. This algorithm compared with the standard algorithm reduces the number of operations from order n^2 to order n , as showed in Table 2. The number of operations has been determined by placing a counter for additions/subtractions and a counter for multiplications/divisions for each element of the vector x . Recurrence has been inferred from an analysis of a case.

Algorithm 2 Multiplication of the Cholesky matrix of $A = T_k(n, d, M)$ by a vector x , $Lx = y$

Input:

- Number of blocks, k .
- Number of elements per block, n_i where $i = 1, \dots, k$.
- Cholesky matrix H with size $k \times (n_1 + \dots + n_k)$.
- Cholesky vector v with size $n_1 + \dots + n_k$.
- Input vector x with size $n_1 + \dots + n_k$.
- Output vector y with size $n_1 + \dots + n_k$.

```

i = 0
for r = 1 to k do
  i = i + 1
   $y_i = \sum_{t=1}^{i-1} H_{rt} \cdot x_t$ 
  for q = 1 to  $n_r - 1$  do
    i = i + 1
     $y_i = y_{(i-1)} + H_{r(i-1)} \cdot x_{(i-1)}$ 
  end for
end for
for j = 1 to i do
   $y_j = y_j + v_j \cdot x_j$ 
end for

```

Output: Vector y

Stability. By multiplying the Cholesky matrix by a vector, Lx , the errors of x can be magnified up to $|\lambda_{\max}(L)| = \sqrt{|\lambda_{\max}(A)|}$. We can calculate this value using the Proposition 1 regarding the eigenvalues of the matrix A .

2.4 Inverse of the Cholesky matrix

The inverse of a lower triangular matrix is another lower triangular matrix that can be easily calculated due to the triangular structure. If the lower triangular matrix is the Cholesky matrix of A , then we can obtain the inverse of A doing $A^{-1} = (LL')^{-1} = (L^{-1})'L^{-1}$. When the matrix A is of type T_k , the rows of L^{-1} are composed of blocks, except on the diagonal.

Algorithm 3 adapts the inversion algorithm of a Cholesky matrix (Press et al 1992) to operate with the adapted Cholesky matrix to calculate only the non-repeated elements and to save memory. The computed values that are distinct from the diagonal values are stored in a matrix of size $n \times k$, and the diagonal elements are stored in a vector of size n .

Algorithm 4 present an algorithm for multiplying the inverse of the Cholesky matrix of an array of type T_k by a vector that takes advantage of repeated values to reduce the number of operations.

Memory requirements. The adapted algorithm reduces the memory needed to store L^{-1} from an order of n^2 to an order of $n \times (k + 1)$. In the standard algorithm, the matrix

Algorithm 3 Inverse of the Cholesky matrix of $A = T_k(n, d, M), L^{-1}$ **Input:**

- Number of blocks, k .
- Number of elements per block, n_i where $i = 1, \dots, k$.
- Cholesky matrix H with size $k \times (n_1 + \dots + n_k)$.
- Cholesky vector v with size $n_1 + \dots + n_k$.
- Inverse matrix J with size $(n_1 + \dots + n_k) \times k$.
- Inverse vector w with size $n_1 + \dots + n_k$.

```

for  $i = 1$  to  $n_1 + \dots + n_k$  do
   $w_i = \frac{1}{v_i}$ 
end for
 $i = 1$ 
for  $r = 1$  to  $k$  do
   $j = i - 1$ 
  for  $s = r$  to  $k$  do
     $\text{sum} = -H_{si} \cdot w_i - \sum_{t=i+1}^{j-1} H_{st} \cdot J_{tr}$ 
    for  $q = 1$  to  $n_s$  do
       $j = j + 1$ 
      if  $j \neq i$  OR  $s \neq r$  then
         $\text{sum} = \text{sum} - H_{s,j-1} \cdot J_{j-1,r}$ 
         $J_{ir} = \frac{\text{sum}}{v_j}$ 
      end if
    end for
  end for
   $i = i + n_r$ 
end for

```

Output:

- Matrix J
- Vector w
- $L_{ij}^{-1} = \begin{cases} 0 & \text{if } i < j \\ w_j & \text{if } i = j \\ J_{ir} & \text{if } i > j \text{ where } j \in I_r \end{cases}$

L^{-1} takes $\dim(L^{-1}) = n \times (n + 1)/2$, while in the adapted algorithm, the matrix L^{-1} takes $\dim(J) + \dim(w) = n \times k + n$.

Number of operations. Table 3 and Table 4 shows the number of operations required to compute the inverse of the Cholesky matrix and the number of operations required to multiply this matrix by a vector using the standard algorithms and the adapted algorithms. In both cases, the number of operations has been determined by placing a counter for additions/subtractions and a counter for multiplications/divisions for each element of the matrix J , for each diagonal element of w , and for each element of the vector x . Recurrence has been inferred from an analysis of several case.

Example 4 We compute the inverse of the Cholesky matrix for the matrix of example 1 using the standard algorithm (L^{-1}) and the adapted algorithm (J, w).

$$L^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -0.5774 & 1.1547 & 0 & 0 & 0 & 0 \\ -0.4083 & -0.4083 & 1.2247 & 0 & 0 & 0 \\ -0.1031 & -0.1031 & -0.1031 & 1.0314 & 0 & 0 \\ -0.0706 & -0.0706 & -0.0706 & -0.4002 & 1.1063 & 0 \\ -0.0319 & -0.0319 & -0.0319 & -0.0956 & -0.0956 & 1.0193 \end{pmatrix}$$

$$J = \begin{pmatrix} 0 & 0 & 0 \\ -0.5774 & 0 & 0 \\ -0.4083 & 0 & 0 \\ -0.1031 & 0 & 0 \\ -0.0706 & -0.4002 & 0 \\ -0.0319 & -0.0956 & 0 \end{pmatrix} \quad w = \begin{pmatrix} 1 \\ 1.1547 \\ 1.2247 \\ 1.0314 \\ 1.1063 \\ 1.0193 \end{pmatrix}$$

Algorithm 4 Multiplication of the inverse of the Cholesky matrix of $A = T_k(n, d, M)$ by a vector x , $L^{-1}x = y$

Input:

- Number of blocks, k .
- Number of elements per block, n_i where $i = 1, \dots, k$.
- Inverse Cholesky matrix J with size $(n_1 + \dots + n_k) \times k$.
- Inverse Cholesky vector w with size $n_1 + \dots + n_k$.
- Input vector x with size $n_1 + \dots + n_k$.
- Output vector y with size $n_1 + \dots + n_k$.

```

for  $i = 1$  to  $n$  do
   $y_i = w_i \cdot x_i$ 
end for
 $i = 0$ 
for  $r = 1$  to  $k$  do
   $sum = 0$ 
  for  $q = 1$  to  $n_r$  do
     $i = i + 1$ 
    if  $q > 1$  then
       $y_i = y_i + sum \cdot J_{ir}$ 
    end if
     $sum = sum + x_i$ 
  end for
  for  $j = i + 1$  to  $n$  do
     $y_j = y_j + sum \cdot J_{jr}$ 
  end for
end for

```

Output: Vector y

3 The t-Student copula

It is common to use correlation as a measure of dependency between random variables. In most cases, this measure does not fully reflect the structure of dependence between them. The mathematical concept that does reflect the structure of dependence between random variables, however, is the copula, which we define below (Embrechts et al 1999).

Definition 2 A copula is the distribution function of a random vector in \mathbb{R}^n with uniform-[0,1] marginals. Alternatively, a copula is any function $C : [0, 1]^n \rightarrow [0, 1]$ that has the following three properties:

- $C(x_1, \dots, x_n)$ is increasing in each component x_i ,
- $C(1, \dots, 1, x_i, 1, \dots, 1) = x_i$ for all $i \in \{1, \dots, n\}$, $x_i \in [0, 1]$, and
- For all $(a_1, \dots, a_n), (b_1, \dots, b_n) \in [0, 1]^n$ with $a_i \leq b_i$ we have:

$$\sum_{i_1=1}^2 \dots \sum_{i_n=1}^2 (-1)^{i_1+\dots+i_n} C(x_{1i_1}, \dots, x_{ni_n}) \geq 0$$

where $x_{j1} = a_j$ and $x_{j2} = b_j$ for all $j \in \{1, \dots, n\}$.

Sklar's theorem (Sklar 1959) (Embrechts et al 2001) states that any multivariate distribution with continuous marginals can be decomposed into the marginals and a copula that reflects the structure of dependence between them. Later, we will use this statement to define and simulate the t-Student copula.

Theorem 1 (Sklar) Let H be an n -dimensional distribution function with margins F_1, \dots, F_n . Then there exists an n -copula C such that for all $x \in \mathbb{R}^n$,

$$H(x_1, \dots, x_n) = C(F_1(x_1), \dots, F_n(x_n))$$

If F_1, \dots, F_n are all continuous, the C is unique; otherwise C is uniquely determined on $\text{Ran}F_1 \times \dots \times \text{Ran}F_n$. Conversely, if C is an n -copula and F_1, \dots, F_n are distribution functions, then the function H defined above is an n -dimensional distribution function with margins F_1, \dots, F_n .

We want to simulate random values $U[0,1]$ correlated according to a given correlation matrix. This involves simulating a copula that has this correlation matrix between its marginals. The elliptical copulas are determined by the correlation matrix. The t-Student copula is a copula of this type with a known simulation algorithm (Embrechts et al 2001) and that allows a wide range of dependence structures through parameter ν . Next, we define the multivariate random variable t-Student (Demarta and McNeil 2005) that will allow us to define the t-Student copula later.

Definition 3 The n -dimensional random vector $X = (X_1, \dots, X_n)^\top$ is said to have a (non-singular) multivariate t distribution with ν degrees of freedom, mean vector μ and positive-definite dispersion or scatter matrix Σ , denoted $t \sim t_n(\nu, \mu, \Sigma)$, if its density is given by

$$f(x) = \frac{\Gamma\left(\frac{\nu+n}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right) \sqrt{(\pi\nu)^n |\Sigma|}} \left(1 + \frac{(x-\mu)^\top \Sigma^{-1} (x-\mu)}{\nu}\right)^{-\frac{\nu+n}{2}}$$

where $|\Sigma|$ represents the absolute value of the determinant of the matrix. The multivariate t has the representation

$$X \stackrel{d}{=} \mu + \sqrt{\frac{\mathbf{v}}{S}} \cdot Z$$

where $Z \sim N(0, \Sigma)$ and $S \sim \chi_{\mathbf{v}}^2$.

We define the t-Student copula as the copula of the multivariate t-Student. Sklar's theorem provides a method for calculating the distribution function (Demarta and McNeil 2005) (Fantazzini 2004).

Definition 4 We call the t-Student copula that with distribution function

$$C_{\mathbf{v}, \Sigma}^n = \int_{-\infty}^{t_{\mathbf{v}}^{-1}(u_1)} \dots \int_{-\infty}^{t_{\mathbf{v}}^{-1}(u_n)} f(x) dx$$

where $f(x)$ is the density function of $t_n(\mathbf{v}, 0, \Sigma)$, and $t_{\mathbf{v}}^{-1}$ denotes the quantile function of the univariate distribution $t_1(\mathbf{v}, 0, 1)$. The copula density is

$$c_{\mathbf{v}, \Sigma}^n(t_{\mathbf{v}}(x_1), \dots, t_{\mathbf{v}}(x_n)) = |\Sigma|^{-\frac{1}{2}} \frac{\Gamma(\frac{\mathbf{v}+n}{2})}{\Gamma(\frac{\mathbf{v}}{2})} \left[\frac{\Gamma(\frac{\mathbf{v}}{2})}{\Gamma(\frac{\mathbf{v}+1}{2})} \right]^n \frac{\left(1 + \frac{\zeta' \Sigma^{-1} \zeta}{\mathbf{v}}\right)^{-\frac{\mathbf{v}+n}{2}}}{\prod_{i=1}^n \left(1 + \frac{\zeta_i^2}{2}\right)^{-\frac{\mathbf{v}+1}{2}}} \quad (1)$$

where $\zeta = (t_{\mathbf{v}}^{-1}(u_1), \dots, t_{\mathbf{v}}^{-1}(u_n))$ is the vector of the t-student univariate inverse distribution functions.

3.1 Spearman's rank correlation

We note that the elliptical copulas are determined by a matrix of correlations, but their correlations do not agree with this matrix. To demonstrate this, we need to introduce the concept of Spearman's rank correlation (Embrechts et al 1999).

Definition 5 Let X and Y be random variables with distribution functions F_1 and F_2 and joint distribution function F . Spearman's rank correlation, also named Spearman's rho, is given by

$$\rho_S(X, Y) = \rho(F_1(X), F_2(Y))$$

where ρ is the usual linear correlation.

We have a bivariate t-Student $X = (X_1, X_2)^\top$, and we want to know the correlation of the copula, $U = (U_1, U_2)^\top$. By Sklar's theorem, the copula marginals are $f(X_1)$ and $f(X_2)$, where f is the univariate $t_{\mathbf{v}}$ distribution. Then, the correlation between U_1 and U_2 is the Spearman's rank correlation between X_1 and X_2 .

$$\rho(U_1, U_2) = \rho(f(X_1), f(X_2)) = \rho_S(X_1, X_2)$$

Here, we present a theorem (Embrechts et al 2001) that allows us to calculate the Spearman's rank correlation.

Theorem 2 Let $(X, Y)^\top$ be a vector of continuous random variables with copula C . Then Spearman's rank correlation for $(X, Y)^\top$ is given by

$$\rho_S(X, Y) = 12 \int \int_{[0,1]^2} C(u, v) du dv - 3$$

Given a bivariate $t_\nu(\rho)$, we call g_ν the function that determines the Spearman's rank correlation from the linear correlation, $\rho_S = g_\nu(\rho)$. This function exists and is continuous and differentiable by the previous theorem and the definition of the t-Student copula. Suppose we can compute g_ν^{-1} . In this case, given any correlation ρ , we can simulate a bivariate t-Student with correlation $g_\nu^{-1}(\rho)$. Its copula will have a correlation $g_\nu(g_\nu^{-1}(\rho)) = \rho$, which is our intention.

ρ_S and, consequently, g_ν have no analytical formulae (Faivre 2003). In the following paragraphs, we will seek an analytic formula that approximates g_ν . The function g_ν has an analytic formula for the Normal copula (Embrechts et al 1999), which is equal to the t-Student copula considering $\nu = \infty$ (Demarta and McNeil 2005). If $X = (X_1, X_2)^\top$ is a bivariate Normal with $\rho(X_1, X_2) = \rho$, then

$$\rho(U_1, U_2) = \rho(\phi(X_1), \phi(X_2)) = \frac{6}{\pi} \arcsin\left(\frac{\rho}{2}\right) = g_\infty(\rho)$$

where ϕ is the standard normal distribution function. We use this result to propose the following approximation to g_ν :

$$g_\nu(\rho) \approx \tilde{g}_\nu(\rho) = \frac{\arcsin(\rho \cdot \sin(h(\nu)))}{h(\nu)}$$

Function h estimation. Fixing ν and ρ , we can simulate a bivariate t-Student copula 1000000 times using the standard algorithm (Embrechts et al 2001) and approximate $g_\nu(\rho)$ using the usual correlation estimator. The existing error prevents us from obtaining $h(\nu)$ directly from a single value. For this reason, to compute $h(\nu)$, we have fixed ν and varied ρ by taking 101 equally spaced points along the interval $[0,1]$ to approximate the value $h(\nu)$ by least squares. Figure 1 shows the case for $\nu = 4$.

With the above procedure, we can calculate the value $h(\nu)$ by using least squares. Now we vary ν along the range $(2, \infty)$, and for each value, we compute the numerical approximation of $h(\nu)$. We adjust this set of points to a function of type

$$h(\nu) = \frac{\pi}{6} + \frac{1}{a + b\nu} \quad \text{if } \nu \in (2, \infty)$$

The coefficients of h determined by least squares are $a = 0.44593$ and $b = 1.3089$. A ν step size of 0.1 in the range $(2, 10)$, a step size of 0.25 in the range $[10, 100)$, and a step size of 1 in the range $[100, 1000]$ have been taken. Figure 2 shows the set of points and the adjusted function.

We estimate the error of \tilde{g} by comparing the value given by \tilde{g} with the estimated correlations used to approximate g . Figure 3 shows the error histogram of the 135340 correlations used to estimate the function g . Note that the error comes from two sources: the approximation of g and the correlation estimation. The observed error was always less than 0.005.

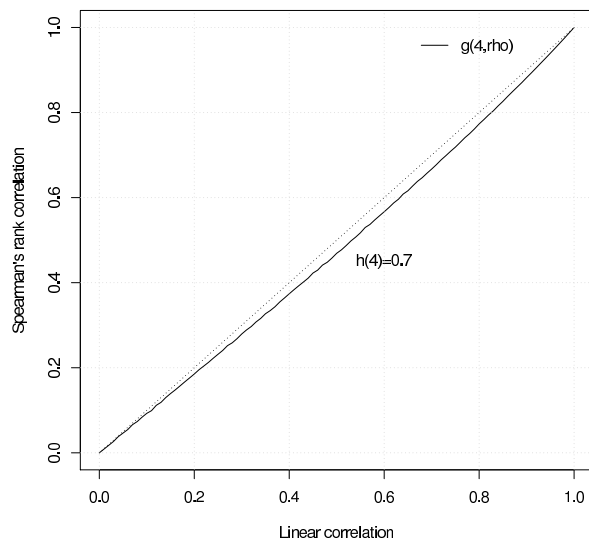


Figure 1 t_4 -Student Spearman's rank correlation fitted by least squares

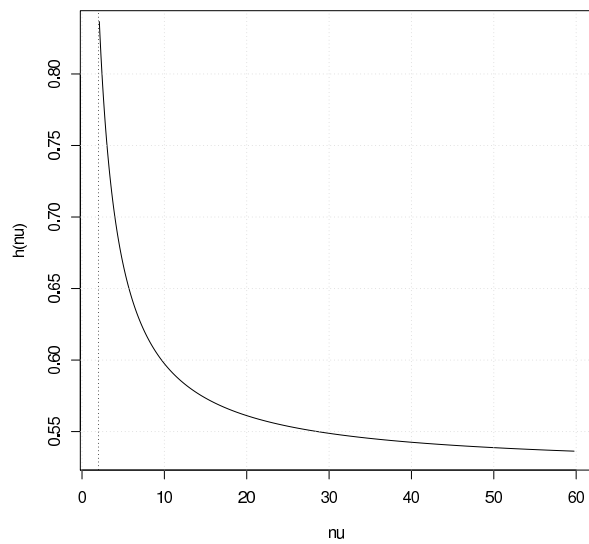


Figure 2 Function h fitted by least squares

3.2 Parameter estimation

The parameter estimation methods based on maximum likelihood need to evaluate the density function of the copula. Here we see that we have all the elements for this evaluation using the Canonical Maximum Likelihood method (Fantazzini 2004), but not limited to it.

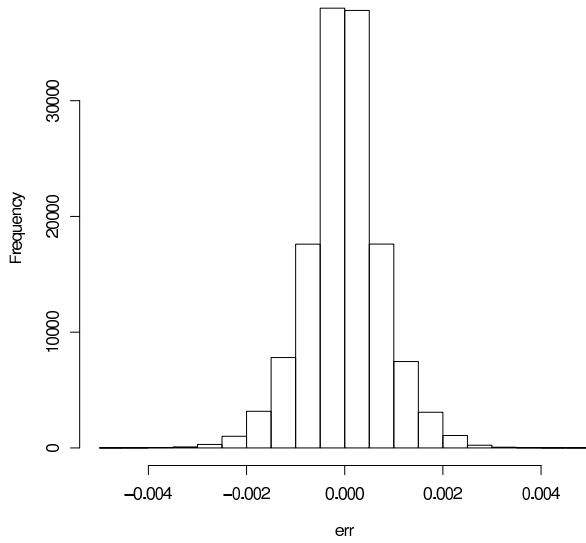


Figure 3 Observed error of $\hat{g}(v, \rho)$

Let $X = (x_{1t}, \dots, x_{nt})$ where n is the copula dimension and $t = 1, \dots, T$ represents the number of observations available. Then, the Canonical Maximum Likelihood states that

$$\hat{\theta} = \arg \max \sum_{t=1}^T \log(c(\hat{u}_{1,t}, \dots, \hat{u}_{n,t}); \theta)$$

where $c(u_1, \dots, u_n)$ is the copula density exposed in (1) and θ is the set of parameters to be estimated.

The determinant of the correlation matrix involved in the evaluation of the density function can be calculated using the Corollary 2. Furthermore, the values $x' \Sigma^{-1} x$ involved in the evaluation of the density function can be calculated using the inverse of the Cholesky matrix presented in section 2.4 doing

$$x' \Sigma^{-1} x = x' (LL')^{-1} x = x' (L'^{-1} L^{-1}) x = (L^{-1} x)' \cdot (L^{-1} x)$$

where L^{-1} can be computed using the Algorithm 3 and $L^{-1} x$ can be computed using the Algorithm 4.

Finally, the powers to the copula dimension involved in the evaluation of the density are converted to products under the sign of logarithm avoiding accuracy problems.

3.3 Simulation

Algorithm 5 is the standard algorithm for the simulation of a t-Student copula (Embrechts et al 2001) adapted to work with blocks matrices. Similarly, we can adapt the simulation standard algorithm of any other elliptical copula based on the Normal, to operate with block matrices.

Algorithm 5 t-Student copula simulation with ν degrees of freedom given a correlation matrix Σ .

Input:

- Numer of blocks, k .
- Number of elements per block, n_i where $i = 1, \dots, k$.
- Block values, ρ_{rs} where $r, s = 1, \dots, k$.
- Degrees of freedom, ν .

(1) Map Σ to A applying the inverse of g_ν :

$$a_{ij} = \frac{\sin(h(\nu) \cdot \rho_{ij})}{\sin(h(\nu))} \quad \text{where} \quad h(\nu) = \frac{\pi}{6} + \frac{1}{0.44593 + 1.3089\nu}$$

- (2) Find the Cholesky decomposition L of A using Algorithm 1.
 (3) Simulate n independent random variates z_1, \dots, z_n from $N(0, 1)$.
 (4) Simulate a random variate s from χ_ν^2 independent of z_1, \dots, z_n .
 (5) Set $y = Lz$ using Algorithm 2.
 (6) Set $x = \frac{\sqrt{\nu}}{\sqrt{s}}y$.
 (7) Set $u_i = t_\nu(x_i)$, $i = 1, \dots, n$

Output:

- Simulated values, u_i .

where $n = n_1 + \dots + n_k$ and $t_\nu(x)$ is the cumulative distribution function for the univariate t-Student with ν degrees of freedom.

4 Conclusions

We have presented a fast algorithm for the simulation of a high-dimensional t-Student copula given a correlation block matrix with low memory requirements. This is the standard algorithm using adapted versions for the Cholesky decomposition and for the product of the Cholesky matrix by a vector to operate with block matrices. The adaptation of the Cholesky decomposition to this class of matrices reduces the memory requirements from order n^2 to $n \times (k + 1)$ and the number of operations from order n^3 to n^2 , where n is the copula dimension, and k is the number of blocks. Additionally, the product of the Cholesky matrix by a vector reduces the number of operations from order n^2 to n .

We have identified the eigenvalues of the block matrices and presented an algorithm for calculating the inverse of the Cholesky matrix for these matrices that allow the copula parameter estimation using the maximum likelihood method.

We have determined an analytical formula that approximates the Spearman's rank correlation for the bivariate t-Student with an error lower than 0.005. This has allowed us to add an initial step to the simulation algorithm that transforms the correlation matrix so that the simulated copula has the given correlation matrix.

References

- Burden RL, Faires JD (1998) *Análisis Numérico*. International Thomson Publishing
- Demarta S, McNeil AJ (2005) The t copula and related copulas. *International Statistical Review* 73:111–129
- Embrechts P, McNeil A, Straumann D (1999) Correlation and dependence in risk management: properties and pitfalls. RiskLab
- Embrechts P, Lindskog F, McNeil A (2001) Modelling dependence with copulas and applications to risk management. RiskLab
- Faivre F (2003) Copula: A new vision for economic capital and application to a four line of business company. Paper presented at ASTIN Colloquium, Berlin
- Fantazzini D (2004) Copula's conditional dependence measures for portfolio management and value at risk. Working paper, Department of Economics, University of Konstanz (Germany)
- Higham NJ (2002) Accuracy and stability of numerical algorithms. SIAM
- Li DX (2000) On default correlation: A copula function approach. Working Paper Number 99-07
- Press W, Teukolsky S, Vetterling W, Flannery B (1992) *Numerical Recipes in C*, 2nd edn. Cambridge University Press, Cambridge, UK
- Sklar A (1959) Fonctions de repartition a n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris* pp 8:229–231

Tables

	Standard algorithm	Adapted algorithm
Addition/ Subtraction	$\frac{n^3}{6} - \frac{n}{6}$	$\frac{(n_1-1) \cdot (n_1)}{2} + \frac{(n_1+n_2-1) \cdot (n_1+n_2)}{2} + \dots + \frac{(n_1+\dots+n_{k-1}-1) \cdot (n_1+\dots+n_k)}{2}$ $+n_1 + (n_1+n_2) + \dots + (n_1+\dots+n_{k-1})$ $-2k + 2 \cdot (n_1 + \dots + n_k) - 1$
Multiplications/ Divisions	$\frac{n^3}{6} + \frac{n^2}{2} - \frac{2n}{3}$	$\frac{(n_1-1) \cdot (n_1)}{2} + \frac{(n_1+n_2-1) \cdot (n_1+n_2)}{2} + \dots + \frac{(n_1+\dots+n_{k-1}-1) \cdot (n_1+\dots+n_k)}{2}$ $+n_1 + (n_1+n_2) + \dots + (n_1+\dots+n_{k-1})$ $-k + (n_1 + \dots + n_k)$
Square Roots	n	n

Table 1 Number of operations in the Cholesky decomposition

	Standard algorithm	Adapted algorithm
Addition/ Subtraction	$\frac{(n-1) \cdot n}{2}$	$2 \cdot (n_1 + \dots + n_k - k)$ $+ (n_1) + (n_1 + n_2) + \dots + (n_1 + \dots + n_{k-1}) + k$
Multiplications/ Divisions	$\frac{n \cdot (n+1)}{2}$	$2 \cdot (n_1 + \dots + n_k - k)$ $+ (n_1) + (n_1 + n_2) + \dots + (n_1 + \dots + n_{k-1}) + k$

Table 2 Number of operations in the multiplication of the Cholesky matrix by a vector

	Standard algorithm	Adapted algorithm
Addition/ Subtraction	$2n + \sum_{i=1}^{n-1} i \cdot (n-i)$	$(n_1 + \dots + n_k) + \dots + (n_{k-1} + n_k) + (n_k) - k$ $+ \sum_{i=1}^{k-1} \left[-(i-1) + \sum_{j=1}^i j \cdot n_{k-j} \right] + 1$
Multiplications/ Divisions	$\frac{(n-1) \cdot n}{2} + n + \sum_{i=1}^{n-1} i \cdot (n-i)$	$2 \cdot \left[(n_1 + \dots + n_k) + \dots + (n_{k-1} + n_k) + (n_k) - k \right]$ $+ \sum_{i=1}^{k-1} \left[-(i-1) + \sum_{j=1}^i j \cdot n_{k-j} \right] + 1$

Table 3 Number of operations in the inverse of the Cholesky matrix

	Standard algorithm	Adapted algorithm
Addition/ Subtraction	$\frac{(n-1) \cdot n}{2}$	$\sum_{i=1}^k (i+1)n_i$
Multiplications/ Divisions	$\frac{n \cdot (n+1)}{2}$	$\sum_{i=1}^k (i+1)n_i$

Table 4 Number of operations in the multiplication of the inverse of the Cholesky matrix by a vector