Simulating Normal and t Copulas

There are standard routines for simulating the multivariate normal and t distributions starting with the correlation matrix. To simulate the corresponding copulas, you first simulate the multivariate distributions, then apply the normal or t distribution to get the probabilities. A simulated copula is a matrix of simulated probabilities that have been correlated according to the copula structure. Once you have those, you can take the inverse probabilities with any desired distributions to get the final simulated values.

The starting point is the Cholesky decomposition of the correlation matrix of a multivariate standard normal distribution. It is a transform of the correlation matrix to a matrix with zeros above the diagonal and gives the regressions (conditional distributions) of each standard normal variable on all the previous ones implied by the correlation matrix. That is, if you multiply it on the right by a column vector of standard normal observations, it gives the mean of each variable conditional on all the previous observations. As a result, multiplying it on the right by column vector of independent standard normals gives a vector of correlated standard normals, with the target correlation matrix. Thus if you can simulate independent standard normals, and you have the Cholesky decomposition, you can produce correlated standard normals that have any desired correlation matrix.

Later the calculation of the Cholesky decomposition is discussed, but there are a number of statistical routines that do this, so for now, how do you simulate the normal copula assuming you have the Cholesky decomposition?

Random number generation and the inverse normal distribution are quite commonly available, so it is easy to generate independent standard normal deviates. In Excel, for instance, if you want k of these, do normsinv(rand()) k times. If **L** is the Cholesky decomposition of the correlation matrix and **z** is a column vector of k independent standard normal simulations, then **x** = **Lz** is the resulting vector of correlated standard normals. If you take the standard normal probabilities of these, as in normsdist(**x**), that gives a simulation of the normal copula.

The t copula is not much more complicated. It takes as parameters a correlation matrix and a single number called degrees of freedom, denoted as n. For the t copula, start by simulating a correlated standard normal variable **x** from the t-copula correlation matrix. Then simulate y from a chi-squared distribution with n degrees of freedom. That is usually easy to do. For instance in Excel you could simulate y by y = 2*gammainv(rand(), n/2, 1). Then $\mathbf{w} = \mathbf{x}(n/y)^{½}$ is a correlated t-distributed vector. Finally, the t distribution with n degrees of freedom is applied to **w** to give the simulated copula. In Excel this can be implemented by the beta distribution , even if n is not an integer, by $T_n(\mathbf{w})$ = ½ + ½ sign(**w**)betadist[$\mathbf{w}^2/(n+\mathbf{w}^2)$, ½, n/2].

If a Cholesky routine is not readily available, you can use this recursive algorithm:

For a starting matrix with elements A_{ij} calculate the Cholesky decomposition using L_{ij} with i > j for the lower triangle, and D_i where the diagonal will be $D_i^{½}$:

$D_1 = A_{11}$, $L_{21} = A_{21}/D_1$, and for i > j ≥ 1,

$$D_i = A_{ii} - \sum_{k=1}^{i-1} L_{ik}^2 D_k$$

$$L_{ij} = \frac{1}{D_j}\left(A_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} D_k\right)$$

The sum is interpreted as zero if the upper index is less than the lower index.