Simulating Historical Ratings Transition Matrices for Credit Risk Analysis in *Mathematica*

Mark S. Coleman  
Chatham Research Alliance  
65 Hillside Avenue, Arlington, MA., 02476, USA  
mark@markcoleman.com

Quantitative methods for evaluating credit risk have gained increasing importance the past several years. Many of the most common analytical credit risk procedures use historical ratings transition matrices published by the three major U.S. rating agencies. In many cases, however, the specific approaches used are too limiting in that the observed historical variation in the Markov transition probabilities is not fully utilized to evaluate the risk of portfolios of credit instruments. We argue that this variation, which in part is caused by changes in the underlying economic and financial factors that determine credit market behavior, is an important element of true underlying risk of these financial instruments. This paper uses *Mathematica* to present an approach for simulating ratings transitions that accounts for the underlying variation in credit transition behavior. We use a copula-based Monte Carlo approach to help account for the historical cross-correlation of ratings probability changes over a 20-year sample of data spanning 1981 - 2000.


1. Introduction

Over the past decade, quantitative methods for evaluating, predicting, and managing the risk of portfolios of credit-based securities have become increasingly important for financial institutions. This need for improved risk assessment is a direct consequence of several factors. First, financial institutions are subject to increasingly stringent formal regulatory requirements, such as The New Basel Capital Accord[1]. Second, there has been an enormous increase in the use of credit derivatives and other credit-related instruments for financing and hedging[2]. Third, investors in credit-sensitive instruments are demanding greater liquidity, lower trading costs, and more "transparency" for their portfolios — requirements that can only be met by the use of sophisticated valuation models. The end result has been a proliferation of credit risk analytics[3] [4].

This paper uses *Mathematica* to develop a generalization of a widely used credit risk estimation procedure. Our method addresses an important limitation to the existing procedure in that we more fully account for the implicit variation seen in published historical ratings transitions. We argue such variations flow from fundamental economic factors and are an important source of underlying credit volatility that are not adequately captured by existing methods. Using published annual transitions data spanning the period 1981 - 2000, we employ copula functions to build a multivariate distribution of ratings probabilities based upon the assumption that each element of the mean transition matrix has a unique (marginal) beta distribution. This allows for correlations across the transition probabilities — a feature consistent with the historical data. We also present a basic Monte Carlo procedure for drawing correlated random vectors from this multivariate distribution.
2. Modeling Credit Risk Using Ratings Transition Matrices

Some of the most widely used credit risk methodologies employ ratings transition matrices [5], published regularly by the major U.S. credit rating agencies [6]. In essence, these matrices depict the average probability of a rated credit security moving from one rating category to another over a pre-determined time interval. The probabilities are tabulated using an historical database of ratings issuance conducted by each agency over a specific period of time. For each period of interest, e.g., one-year, a cohort of existing credits is first defined and then tracked through the end of the period. Any changes in ratings issued by the agency over the period is noted and then used to update the proportion of credits in each cell of the matrix at the end of period. This method of tabulation leads to a rather obvious Markov interpretation; hence formal Markov methods are typically used to model transition behavior and credit risk.

For instance, consider the following simple hypothetical example [5]. This example calculates the expected value for a single bond one-year ahead. For the purposes of illustrating our method, we adopt the conventional Standard & Poor's ratings categories, AAA, AA, A, BBB, BB, B, CCC, D, where D denotes default and is an absorbing state. We begin with a $100 par value bond, having a 6% annual coupon and a five year maturity with a beginning-of-year rating of BBB. Nominally, an investor holding this bond would receive $6 at the end of each of the first four years. When the bond matures at the end of year 5, the investor receives $106 (the $100 face value plus the final $6 coupon). The value of such a bond today is simply the present discounted value of this five year earnings stream, assuming some appropriate market discount rate. This simple calculation, however, ignores the credit risk of the bond. To determine the expected value of a risky bond, we need to calculate a discounted value for each possible coupon stream that the bond would pay should it migrate to another rating category (for simplicity we assume that only one transition can occur). This requires a vector of one-year transition probabilities and a vector of discount rates corresponding to each possible rating category for each year of the bond's life (such rates are referred to as forward zero curves). The following table presents these inputs:

<table>
<thead>
<tr>
<th>Year-End Rating</th>
<th>Transition Probability</th>
<th>Forward Zero Year 1</th>
<th>Forward Zero Year 2</th>
<th>Forward Zero Year 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>0.02 %</td>
<td>3.6 %</td>
<td>4.17 %</td>
<td>4.73 %</td>
</tr>
<tr>
<td>AA</td>
<td>0.33 %</td>
<td>3.65 %</td>
<td>4.22 %</td>
<td>4.78 %</td>
</tr>
<tr>
<td>A</td>
<td>5.95 %</td>
<td>3.72 %</td>
<td>4.32 %</td>
<td>4.93 %</td>
</tr>
<tr>
<td>BBB</td>
<td>86.93 %</td>
<td>4.10 %</td>
<td>4.67 %</td>
<td>5.25 %</td>
</tr>
<tr>
<td>BB</td>
<td>5.30 %</td>
<td>5.55 %</td>
<td>6.02 %</td>
<td>6.78 %</td>
</tr>
<tr>
<td>B</td>
<td>1.17 %</td>
<td>6.05 %</td>
<td>7.02 %</td>
<td>8.03 %</td>
</tr>
<tr>
<td>CCC</td>
<td>0.12 %</td>
<td>15.05 %</td>
<td>15.02 %</td>
<td>14.03 %</td>
</tr>
<tr>
<td>Default</td>
<td>0.18 %</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

(1) Assumes a post-default recovery value of $51.13

For instance, if the bond is upgraded from BBB to A status, then the value of the bond at the end of one year is simply

\[ V(BBB \rightarrow A) = \frac{6}{1 + 0.0372} + \frac{6}{(1 + 0.0432)^2} + \frac{6}{(1 + 0.0493)^3} + \frac{6}{(1 + 0.0532)^4} = 108.66 \]  

(1)

Table 1 repeats this calculation for each possible rating migration, yielding a set of future values for the bond (note: we use an industry-standard assumption about the recovery value of the bond should it migrate to the default state). Each of
these values is then weighted by the corresponding migration probability, giving an expected value of the bond at the end of one year. The volatility of the one-year ahead value is then calculated as a standard deviation in the usual manner. In this case

\[ E(V) = \$107.09, \quad \sigma(V) = \$2.99 \]  

(2)

3. Economic Variation and Correlations in Ratings Transitions

Though the example shown above is admittedly simplified, it is not far from procedures used in practice. In particular, most transition-based credit risk models rely upon a single point estimate of transition probabilities. This has the obvious defect of ignoring the time series behavior of transition probabilities, as well as any cross-category correlations. This is all the more curious in that, within the context of broader macroeconomic business cycles, credit cycles are quite pronounced and exhibit strong empirical regularity and predictability over time [6]. Credit quality is strongly determined by borrower earnings, which in turn are tied to broader macroeconomic factors such as sales, wages, productivity, interest rates, exports, etc. The historical correlation between key credit market factors — including the values and volatilities of the assets underlying the credit obligation — and general economic indicators is well-documented [6] [7]. In addition, credit market spreads — differences between yields on comparable maturity bonds of different credit quality — and spread volatilities also have strong statistical ties to broader capital market factors [8]. Another more subtle reason to expect economic correlation is that ratings analysts are also monitoring economic conditions and will periodically adjust rating criteria to reflect their perceptions of future economic risk. As a result, a reasonable estimate of expected credit risk should reflect prospective changes in underlying economic factors.

The historical volatility of selected transition probabilities can be seen in Figure 1. The graph depicts the time series of the "no-migration" (the main diagonal) probabilities taken from the series of one-year transition matrices published by Standard & Poor’s over the period 1981 - 2000. There is obvious cyclical variation over this period, which includes a full business cycle. The statistical evidence indicates significant first-order autocorrelation for several ratings levels, as well as significant correlation across several categories. This is the case for any combination each of the rating classes (a summary of the sample statistics and correlations are available from the author upon request)

![Figure 1: Diagonal transition probabilities by rating category](image)
4. A Note on Calculating Markov Generators

As noted above, transition matrices are often modeled using Markov processes. Published transition matrices, however, are not always well-suited for valuing many types of credit securities. For example, transition matrices are typically available only for annual frequencies, with the shortest period being one year. Many credit instruments have maturities under one year, making accurate valuation problematic. In addition, since many published matrices are at best only approximations to true Markov processes[8], complicating the valuation of longer-term securities as well since multi-year transition probabilities cannot always be accurately calculated by the product of the one-year transitions. In a recent article, Isreal, Rosenthal, and Wei [9] (IRW) identify a set of conditions under which a valid Markov generator exists and present an algorithm for estimating an approximate generator when a true generator does not exist. If \( P \) is a time-homogenous \( N \times N \) Markov transition matrix, then a generator \( Q \) has the property that \( P(t) = \exp(tQ) \). Using a Markov generator implies that proper transition probabilities can be estimated for any \( t \geq 0 \), enabling the evaluation of credit securities of any maturity. We have implemented the IRW procedure in *Mathematica* (available upon request) and applied it to each of the one-year transition matrices in our sample. The results that follow reflect this adjustment.

5. Simulating Transitions

A. Generating Uncorrelated Transition Probabilities

Markov models of credit risk treat default rates as a random variable. Default rates are often modeled as gamma distributed random variates, which follows from the idea that the underlying number of defaults in any transition cell can be modeled as a Poisson arrival process. We generalize this approach by employing a Monte Carlo procedure, where we draw from a unique beta distribution for each transition probability. The beta distribution is well-suited to modeling set of proportions, and can be shown to generalize the gamma distribution is certain circumstances [10]. In the discussion that follows, we begin by assuming no correlation between probabilities; we then relax this assumption using a procedure shown in Clemen and Reilly [11] that employs a multivariate normal copula function to define a joint distribution of the operabilities using only marginal distributions and pairwise correlations.

The beta distribution is defined as

\[
F(x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha) \Gamma(\beta)} x^{\alpha-1} (1 - x)^{\beta-1}
\]

for \( \alpha > 0, \beta > 0, \text{and} \ 0 < x < 1 \), where \( \Gamma(*) \) is the gamma function. The mean and variance of (3) can be written in terms of \( \alpha \) and \( \beta \) as

\[
\mu = \frac{\alpha}{\alpha + \beta} \quad \text{and} \quad \sigma^2 = \frac{\alpha \beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)}
\]

By solving (4), we can parameterize the specific beta distributions that we need for the Monte Carlo simulation. Solving this system yields

\[
\text{betaParams} = \text{Solve}\{\{\mu = \alpha / (\alpha + \beta), \ \sigma^2 = (\alpha \beta) / ((\alpha + \beta)^2 (\alpha + \beta + 1))\}, \{\alpha, \beta\}\}
\]

\[
\{\{\beta \to (\mu - 2 \mu^2 + \mu^2 - \sigma^2 + \mu \sigma^2) / \sigma^2, \ \alpha \to -((\mu - \mu^2 + \sigma^2) / \sigma^2)\}
\]
As an example, consider the mean of the mean and standard deviation of the AAA→AAA transition probability. The corresponding beta parameters are

\[
\beta = \text{Reverse[Flatten[betaParams /. \{\mu \to 0.92, \sigma^2 \to 0.00222\}]]}
\]

\[
\{\beta \to 2.57225, \alpha \to 29.5809\}
\]

So to generate a random sample,

\[
\text{RandomArray[BetaDistribution[bp[[1]][[2]], bp[[2]][[2]]], \{1000, 7\}]; // Timing}
\]

\[
0.6 \text{ Second, Null}
\]

To implement this on our data sample, we first calculated the mean one-year transition probabilities by averaging across each of the cells in our historical sample. We then calculated the variance of each cell in a similar manner. The resulting matrices are

\[
\begin{bmatrix}
0.93 & 0.062 & 0.0053 & 0.0011 & 0.000034 & 0.000048 & 0.000012 & 0.00001
0.0075 & 0.92 & 0.068 & 0.006 & 0.0011 & 0.000029 & 0.000018
0.001 & 0.024 & 0.91 & 0.054 & 0.0061 & 0.0025 & 0.00024 & 0.00058
0.00047 & 0.0032 & 0.055 & 0.87 & 0.05 & 0.011 & 0.00019 & 0.00028
0.00021 & 0.0011 & 0.0055 & 0.07 & 0.81 & 0.0033 & 0.01 & 0.013
0.000064 & 0.0011 & 0.0042 & 0.0055 & 0.06 & 0.0135 & 0.055
0.0011 & 0.00013 & 0.0036 & 0.0063 & 0.018 & 0.11 & 0.65 & 0.22
0.0.0.0.0.0.0.0.0.1.
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.00241 & 0.00235 & 0.0000297 & 3.74 \times 10^{-6} & 1.51 \times 10^{-6} & 7.33 \times 10^{-6} & 5.48 \times 10^{-10} & 2.01 \times 10^{-10}
0.0000431 & 0.000572 & 0.000439 & 0.0000219 & 5.13 \times 10^{-6} & 5.71 \times 10^{-6} & 5.88 \times 10^{-7} & 1.44 \times 10^{-7}
2.63 \times 10^{-6} & 0.00018 & 0.00117 & 0.000379 & 0.000041 & 0.000018 & 1.33 \times 10^{-7} & 1.01 \times 10^{-7}
1.2 \times 10^{-6} & 7.51 \times 10^{-6} & 0.000688 & 0.000238 & 0.000407 & 0.000118 & 5.16 \times 10^{-6} & 4.86 \times 10^{-6}
4.89 \times 10^{-7} & 4.26 \times 10^{-6} & 0.0000147 & 0.000891 & 0.00332 & 0.00013 & 0.000145
5.19 \times 10^{-8} & 5.49 \times 10^{-6} & 0.000010 & 8.29 \times 10^{-6} & 0.00104 & 0.00265 & 0.000642 & 0.00112
5.0000256 & 3.75 \times 10^{-8} & 0.0000839 & 0.000124 & 0.00048 & 0.0103 & 0.0196 & 0.0151
\end{bmatrix}
\]

Now calculate the beta parameters for each element of the mean transition matrices (note: warnings suppressed)

\[
\text{betaParamsMat = betaParams \/. \{\mu \to \text{meanfinalTrans}, \sigma^2 \to \text{varfinalTrans}\}}
\]

\[
\{\beta \to \{3.9197, 23.0882, 116.71, 208.958, 184.085, \text{Indeterminate}, \text{Indeterminate}, \text{Indeterminate}, 122.573\},
\{333.291, 127.419, 10.346, 127.904, 145.088, 136.46, 832.233, 422.391, 97.2141\},
\{388.355, 312.005, 74.9291, 8.85981, 121.497, 82.3819, 317.473, 418.891, 115.009\},
\{332.233, 224.652, 293.319, 79.5213, 14.1701, 18.3348, 88.4819, 89.0936, 58.325\},
\{\text{Indeterminate}, \text{Indeterminate}, \text{Indeterminate}, \text{Indeterminate}, \text{Indeterminate}, \text{Indeterminate}, \text{Indeterminate}, \text{Indeterminate}, 12.2034\},
\{96.478, \text{Indeterminate}, 38.3763, 43.4116, 34.2216, 9.1537, 4.10454, 9.74711, 12.2034\},
\alpha \to \{35.3657, 1.48783, 0.458755, 0.165207, 0.049165, \text{Indeterminate}, \text{Indeterminate}, \text{Indeterminate}, 4.32007\},
\{1.24606, 129.379, 9.88445, 0.953702, 0.134375, 0.18654, 0.0992414, 0.049165, 3.1325\},
\{0.290216, 2.98495, 73.2246, 7.06425, 0.859632, 0.318897, 0.049937, 0.185934, 3.95574\},
\{0.147001, 0.82901, 4.18807, 42.4046, 6.11987, 0.82356, 0.510918, 0.978295, 6.68273\},
\{0.0498425, 0.189991, 1.38215, 5.47458, 41.3392, 1.54666, 0.815733, 1.01006, 5.56061\},
\{\text{Indeterminate}, 0.196713, 7.55279, 1.87412, 3.57773, 29.1936, 1.79579, 2.42991, 5.58522\},
\{0.046988, \text{Indeterminate}, 0.0952169, 0.204122, 0.457416, 1.00917, 5.48954, 2.25095, 1.65694\}}
\]

Now extract the \(\alpha\) and \(\beta\) parameters into separate lists:
\[
\alpha_{\text{Mat}} = \text{betaParamsMat[[1]][[2]][[2]]};
\]
\[
\beta_{\text{Mat}} = \text{betaParamsMat[[1]][[1]][[2]]};
\]

At this point, we need to handle the indeterminate cases. We assign the indeterminate entries values \( \alpha \rightarrow 0.0000001 \) and \( \beta \rightarrow 0.15 \). This will force the beta probabilities to be effectively zero.

\[
\alpha_{\text{Mat}} = \alpha_{\text{Mat}} \cdot \text{Indeterminate} \rightarrow 0.00000001;
\]
\[
\beta_{\text{Mat}} = \beta_{\text{Mat}} / \cdot \text{Indeterminate} \rightarrow 0.15;
\]

Recall from the example above that to value a specific bond, we first consider its initial credit rating. This corresponds to selecting a specific row of \( \alpha_{\text{Mat}} \) and \( \beta_{\text{Mat}} \). For this row, we calculate two beta parameters for each element (corresponding to one of the eight possible ratings to which the bond could migrate). Thus we need to generate \( N \) trials for each of the elements of

\[
\text{ratingsList} = \{"AAA", "AA", "A", "BBB", "BB", "B", "CCC", "D"\};
\]

The following module generates the Monte Carlo draws from the set of beta distributions

\[
\text{transMonteCarlo}[\text{rating}_\text{, n}] :=
\]
\[
\text{Module}[
\{r, \alpha_{\text{Vec}}, \beta_{\text{Vec}}\},
\]
\[
r = \text{Flatten}[\text{Position}[\text{ratingsList}, \text{rating}]] ;
\]
\[
\alpha_{\text{Vec}} = \text{Flatten}[\alpha_{\text{Mat}}[[r]]]; \quad \beta_{\text{Vec}} = \text{Flatten}[\beta_{\text{Mat}}[[r]]];
\]
\[
\text{Map}[\text{RandomArray}[\#, \text{n}] \&, \text{MapThread}[\text{BetaDistribution}, \{\alpha_{\text{Vec}}, \beta_{\text{Vec}}\}]]
\]

For instance, to generate 10,000 trials of the vector of migration probabilities for a bond initially rated "A",

\[
y = \text{transMonteCarlo}["A", 10000]; \quad \text{// Timing}
\]
\[
5.71 \text{ Second, Null}
\]

B. Generating Correlated Transitions

As noted above, transition probabilities show evidence of correlation across rating categories. In general, to simulate a vector of correlated variates requires specification of the underlying joint density. This can be an unwieldy procedure if the univariate marginal distributions are complex, or if the number of variates is large. An alternative procedure is suggested by Clemen and Reilly [11], in which an arbitrarily complex multivariate density function can be constructed using only marginal distributions and a standard measure of pairwise dependence among the random variables. Recently, copulas have emerged as an important tool to model correlations and dependence in quantitative finance [12] [13] [14].

The core of the Clemen and Reilly approach makes use of copula functions. Let \( F(x_1, ..., x_n) \) be a multivariate distribution function for random variables \( X_1, ..., X_n \), with corresponding marginal distributions \( F_1(x_1), ..., F_n(x_n) \). Then

\[
F(x_1, ..., x_n) = \text{C}(F_1(x_1), ..., F_n(x_n)) \tag{5}
\]

where \( \text{C}(u_1, ..., u_n) \) is the joint distribution with uniform marginals. Under the assumption that each \( F_i \) and \( C \) are differentiable, then the corresponding joint probability density function \( f(x_1, ..., x_n) \) can we written as

\[
f(x_1, ..., x_n) = f_1(x_1) \times \cdots \times f_n(x_n) \times c[F_1(x_1), ..., F_n(x_n)] \tag{6}
\]

where \( f_i(x_i) \) is the density corresponding to the distribution \( F_i(x_i) \), and \( c = \frac{\partial^2 C}{\partial F_1 \partial F_2} \) is called the copula function. The copula function \( c \) essentially "couples" the marginals to a joint distribution and incorporates information about the
dependence among the underlying random variates.

The copula approach has several important advantages over the conventional approach of expressing a joint density function as the product of conditional densities and marginal densities. First, arbitrarily complex marginal distributions can be accommodated. Second, dependencies across the variates can often be expressed as pairwise correlations, which are often simple for analysts to assess, even for large problems; Third, the copula-based approach is often computationally more efficient to sample from than drawing directly from the complete joint density, making it useful for Monte Carlo procedures.

To make this procedure operational, we need to define a copula that embeds the dependence among the transition probabilities. One of the simplest and most common measures of statistical dependence is the Pearson product-moment correlation. Following Clemen and Reilly, we make use of the multivariate normal copula, \( c_N \), which makes use of the correlation matrix in the same was as the familiar multivariate normal distribution. Let \( \mathbf{R} \) be an \( n \times n \) positive definite real matrix giving the pairwise correlations of the \( X_i \)'s, and let \( \Phi \) denote the univariate standard normal distribution. If \( y = (y_1, ..., y_n) \) is an \( n \)-dimensional vector, the multivariate normal copula can be written

\[
c_N(\Phi(y_1), ..., \Phi(y_n) | \mathbf{R}) = \exp\left[ -\frac{1}{2} y^T (\mathbf{R}^{-1} - \mathbf{I}) y \right] / |\mathbf{R}|^{1/2}
\]  

(7)

where \( \mathbf{I} \) is the \( n \times n \) identity matrix. To develop the multivariate density, we use the normal inverse transformation \( \Phi^{-1} \) and define \( y_i = \Phi^{-1}(F_i(x_i)) \) for \( i = 1, ..., n \). Substituting into (6) and (7) gives

\[
f(x_1, ..., x_n | \mathbf{R}) = f_1(x_1) \times \cdots \times f_n(x_n) \times \exp \left\{ - (\Phi^{-1}(F_1(x_1)), ..., \Phi^{-1}(F_n(x_n)))^T \times (\mathbf{R}^{-1} - \mathbf{I}) \times (\Phi^{-1}(F_1(x_1)), ..., \Phi^{-1}(F_n(x_n)))^T / 2 \right\} / |\mathbf{R}|^{1/2}
\]  

(8)

To implement this method in Mathematica we first define a the marginal density function and CDF for each of the cells. Since each is a beta distribution, we can write

\[
\text{betaDensity}[^\alpha, ^\beta, x_] := \text{PDF}[	ext{BetaDistribution}[^\alpha, ^\beta], x];
\text{betaCDF}[^\alpha, ^\beta, x_] := \text{CDF}[	ext{BetaDistribution}[^\alpha, ^\beta], x];
\]

The first term in (8) is a product of the underlying beta marginals. Using the convention discussed above for a specific initial rating category, the following module calculates the marginal density product

\[
\text{densityProduct}[^\text{rating}, x_] := \text{Module}[\{r, ^\text{aVec}, ^\text{bVec}\},
\begin{align*}
  r &= \text{Flatten}[^\text{Position}[\text{ratingsList}, ^\text{rating}]]; \\
  ^\text{aVec} &= \text{Flatten}[\text{aMat}[[r]]]; \\
  ^\text{bVec} &= \text{Flatten}[\text{bMat}[[r]]]; \\
  \text{Apply}[\text{Times}, \text{MapThread}[\text{betaDensity}, \{^\text{aVec}, ^\text{bVec}, x\}]]
\end{align*}
\]

Next we define the input vector for the normal copula, using the inverse normal CDF

\[
\text{copulaInput}[^\text{rating}, x_] := \text{Module}[\{r, ^\text{aVec}, ^\text{bVec}\},
\begin{align*}
  r &= \text{Flatten}[^\text{Position}[\text{ratingsList}, ^\text{rating}]]; \\
  ^\text{aVec} &= \text{Flatten}[\text{aMat}[[r]]]; \\
  ^\text{bVec} &= \text{Flatten}[\text{bMat}[[r]]]; \\
  \text{Print}[\text{MapThread}[\text{betaCDF}, \{^\text{aVec}, ^\text{bVec}, x\}]]; \\
  \text{Map}[\text{Quantile}[\text{NormalDistribution}[0, 1], ^\#] &, \text{MapThread}[\text{betaCDF}, \{^\text{aVec}, ^\text{bVec}, x\}]]
\end{align*}
\]

Now set-up the multivariate normal copula function, incorporating the correlation matrix \( \mathbf{R} \),

\[
\text{mvnCopula}[y_, \mathbf{R}_] :=
\text{Exp}[\left(-y . (\text{Inverse}[^\mathbf{R}] - \text{IdentityMatrix}[\text{Dimensions}[^\mathbf{R}][[1]]]) . y \right) / 2.] / \text{Sqrt}[\text{Det}[^\mathbf{R}]]
\]

Finally, we define the complete multivariate joint density for the initial rating category,
mvBeta[rating_, x_, R_] :=
  densityProduct[rating, x] . mvnCopula[copulaInput[rating, x], R];

Since our convention applies to a prespecified initial rating, we also need a way of calculating R corresponding to each initial rating state. Assuming the historical data is kept in finalTransList, then

  corrCopula[rating_] := Module[{r},
    r = Flatten[Position[ratingsList, rating]];
    CorrelationMatrix[Map[Extract[#, r] & , finalTransList]]
  ];

For instance, for the AAA rating, the historical correlation matrix is

  corrCopula["AAA"] // MatrixForm

  1.  -0.992454  -0.209104  0.0933114  0.113246  0.200519  0.198666  0.163038
  -0.992454  1.  -0.0982446 -0.106344  -0.144779 -0.203931 -0.193374 -0.210668
 -0.209104 -0.0982446  1.  -0.221863  0.0230232 -0.171409 -0.190661 -0.219436
  0.0933114 -0.106344 -0.221863  1.  -0.0272803 0.0173925 -0.180626 0.264366
  0.113246 -0.144779  0.0230232 -0.0272803  1.  0.790768 0.549566 0.39899
  0.200519 -0.203931 -0.171409  0.0173925  0.790768  1.  0.692964 0.440961
  0.198666 -0.193374 -0.139061 -0.180626  0.649566 0.692964  1.  0.418955
  0.163038 -0.210668  0.219436  0.264166  0.39899  0.440961  0.418955  1.

Using mvBeta[ ], it is straightforward to evaluate any function of the multivariate distribution, such as conditional densities, expected values, etc. For purposes of generating correlated beta variates for a Monte Carlo simulation of transition probabilities, we generate a vector of multivariate standard normal variates with correlation matrix R, apply the standard normal distribution function \( \Phi(*) \) to each element of the vector, and then transform these elements to the appropriate beta distribution using \( F_\gamma^{-1}[\Phi(*)] \) (see the module copulaInput[ ]).

**References and Notes**

1. See http://www.bis.org/publ/bcbsca.htm for details.


