OPEN ACCESS Freely available online

PLOS ONE

Generating Correlation Matrices Based on the Boundaries of Their Coefficients

Kawee Numpacharoen[1,2]*, Amporn Atsawarungruangkit[3]

1 Financial Product Development, Kasikorn Securities, Bangkok, Thailand, **2** Department of Mathematics, Faculty of Science, Mahidol University, Bangkok, Thailand, **3** College of Medicine, Rangsit University, Bangkok, Thailand

Abstract

Correlation coefficients among multiple variables are commonly described in the form of matrices. Applications of such correlation matrices can be found in many fields, such as finance, engineering, statistics, and medicine. This article proposes an efficient way to sequentially obtain the theoretical bounds of correlation coefficients together with an algorithm to generate n × n correlation matrices using any bounded random variables. Interestingly, the correlation matrices generated by this method using uniform random variables as an example produce more extreme relationships among the variables than other methods, which might be useful for modeling complex biological systems where rare cases are very important.

Citation: Numpacharoen K, Atsawarungruangkit A (2012) Generating Correlation Matrices Based on the Boundaries of Their Coefficients. PLoS ONE 7(11): e48902. doi:10.1371/journal.pone.0048902

Editor: Enrico Scalas, Università del Piemonte Orientale, Italy

Received August 27, 2012; **Accepted** October 1, 2012; **Published** November 12, 2012

Funding: The authors have no support or funding to report.

Competing Interests: One of the authors (KN) is a paid employee with Kasikorn Securities. This does not alter the authors' adherence to the PLOS ONE policies on sharing data and materials.

* E-mail: kawee.num@student.mahidol.ac.th

Introduction

Many important properties of financial models, engineering problems, and biological systems can be represented as correlation matrices, which describe the linear relationships among variables. It is not always the case that these correlation matrices are known; therefore, correlation matrices are an integral part of simulation techniques for solving or analyzing problems in, for example, signal processing [1], portfolio selection [2], factor analytic research [3], genetic modeling [4], and neuroscience [5].

To create a correlation matrix, it is important to ensure that it is valid, meaning that the matrix must be symmetric and positive semi-definite, with the unit diagonal and other elements in the closed interval $[-1, 1]$. On the contrary, an invalid correlation matrix is one in which assets or variables cannot be correlated according to the specified relationship. The simplest method for constructing a correlation matrix is to use the rejection sampling method, which generates correlation coefficients using uniform random variables in the closed interval $[-1, 1]$. Subsequently, we check whether the matrix is semi-definite and, if not, another correlation matrix is generated. This procedure is repeated until a valid matrix is obtained. Further details of rejection sampling will be described later in this article. For a low-dimensional matrix, it is relatively easy to use rejection sampling, but when the dimension is greater than or equal to four, the chance of finding a valid correlation matrix becomes very low. However, the number of variables in physical or economic systems is normally considerably greater than four, and so the rejection sampling method is considered inefficient for the large-scale construction of correlation matrices.

Instead, for large-dimensional problems, there are several techniques for generating a correlation matrix. These can be classified, based on the relevant objectives or constraints, as follows:

1. Generating of a correlation matrix with predetermined eigenvalues and spectrum [6,7,8];
2. Generating of a correlation matrix with a given mean value [9];
3. Generating of a correlation matrix based on a random Gram matrix [10]; and
4. Generating of a correlation matrix in which each correlation coefficient is distributed within its boundaries [11].

This article focuses on the fourth method presenting an efficient algorithm to calculate the theoretical boundaries of correlation coefficients without the use of optimization techniques. Instead, the theoretical boundaries of each correlation coefficient are calculated from the mathematical structure of the correlation matrix constructed by hypersphere decomposition [12]. Although the theoretical work conducted in [11] is similar to the methodology presented here, its primary technique is the optimization approach, whereas our work uses a non-optimization technique. In addition, the sequence for computing the boundaries of each correlation coefficient is heavily reliant on the concept of adjusting the correlation matrix [13] and its boundaries [14]. After finding the theoretical bounds, we present the techniques for generating a correlation matrix.

Methods

Valid correlation matrix

It is important to have a common understanding of the definition of a valid correlation matrix. Such a matrix conforms to the following properties:

1. All diagonal entries must be equal to one;
2. Non-diagonal elements consist entirely of real numbers in the closed interval [−1, 1];
3. The matrix is symmetric; and
4. The matrix is positive semi-definite.

The first three requirements are relatively easy to satisfy. However, the final property of being positive semi-definite requires all eigenvalues to be greater than or equal to zero.

Interestingly, a valid correlation matrix (C) can be constructed using a method proposed in [12] in terms of trigonometric functions. The correlation matrix then becomes a function of angles ($\theta(i,j)$), which finally gives an efficient way of computing the correlation matrix boundaries without using an optimization method. According to [12], the valid correlation matrix can be described as:

$$C = BB^T, \tag{1}$$

$$b_{i,j} = \begin{cases} \cos\theta_{i,j} & \text{for } j = 1 \\ \cos\theta_{i,j} \cdot \prod_{k=1}^{j-1} \sin\theta_{i,k} & \text{for } j = 2 \text{ to } n-1 \\ \prod_{k=1}^{j-1} \sin\theta_{i,k} & \text{for } j = \text{n} \end{cases}. \tag{2}$$

Generally, B is a square matrix with n dimensions whose elements are represented by the $b_{i,j}$ in (2). As explained in [15], (2) can be simplified by setting $\theta_{i,i}$ to zero for all i. B then reduces to a lower triangular matrix, and:

$$b_{i,j} = \begin{cases} 1 & \text{for } i = 1, j = 1 \\ \cos\theta_{i,j} & \text{for } i \geq 2, j = 1 \\ \prod_{k=1}^{j-1} \sin\theta_{i,k} & \text{for } i = j, 2 \leq i, j \leq n \\ \cos\theta_{i,j} \cdot \prod_{k=1}^{j-1} \sin\theta_{i,k} & \text{for } 2 \leq j \leq i-1 \\ 0 & \text{for } i+1 \leq j \leq n \end{cases} \tag{3}$$

As a result, B can be expressed as

$$B = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \cos\theta_{2,1} & \sin\theta_{2,1} & 0 & \cdots & 0 \\ \cos\theta_{3,1} & \cos\theta_{3,2}\sin\theta_{3,1} & \sin\theta_{3,2}\sin\theta_{3,1} & \cdots & 0 \\ \cos\theta_{4,1} & \cos\theta_{4,2}\sin\theta_{4,1} & \cos\theta_{4,3}\sin\theta_{4,2}\sin\theta_{4,1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \cos\theta_{n,1} & \cos\theta_{n,2}\sin\theta_{n,1} & \cos\theta_{n,3}\sin\theta_{n,2}\sin\theta_{n,1} & \cdots & \prod_{k=1}^{n-1}\sin\theta_{n,k} \end{bmatrix}. \tag{4}$$

It is evident from (4) that matrix B depends solely on $\theta_{i,j}$, which is called the correlative angle. The square matrix of correlative angles (θ) is defined as:

$$\theta = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \theta_{2,1} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \theta_{3,1} & \theta_{3,2} & 0 & \cdots & 0 & 0 & 0 \\ \theta_{4,1} & \theta_{4,2} & \theta_{4,3} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \theta_{n-1,1} & \theta_{n-1,2} & \theta_{n-1,3} & \cdots & \theta_{n-1,n-2} & 0 & 0 \\ \theta_{n,1} & \theta_{n,2} & \theta_{n,3} & \cdots & \theta_{n,n-2} & \theta_{n,n-1} & 0 \end{bmatrix}. \tag{5}$$

Thus, a valid correlation matrix can be calculated if the correlative angle matrix (θ) in (2.5)is known.

Example 1. Let us assume that the four-dimensional correlative angle matrix is:

$$\theta = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \theta_{2,1} & 0 & 0 & 0 \\ \theta_{3,1} & \theta_{3,2} & 0 & 0 \\ \theta_{4,1} & \theta_{4,2} & \theta_{4,3} & 0 \end{bmatrix}. \tag{6}$$

The matrix B can then be expressed as:

$$\begin{aligned} B &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ b_{2,1} & b_{2,2} & 0 & 0 \\ b_{3,1} & b_{3,2} & b_{3,3} & 0 \\ b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ \cos\theta_{2,1} & \sin\theta_{2,1} & 0 & 0 \\ \cos\theta_{3,1} & \cos\theta_{3,2}\sin\theta_{3,1} & \sin\theta_{3,2}\sin\theta_{3,1} & 0 \\ \cos\theta_{4,1} & \cos\theta_{4,2}\sin\theta_{4,1} & \cos\theta_{4,3}\sin\theta_{4,2}\sin\theta_{4,1} & \sin\theta_{4,3}\sin\theta_{4,2}\sin\theta_{4,1} \end{bmatrix} \end{aligned}. \tag{7}$$

Finally, the correlation matrix is:

$$C = BB^T = \begin{bmatrix} 1 & c_{2,1} & c_{3,1} & c_{4,1} \\ c_{2,1} & 1 & c_{3,2} & c_{4,2} \\ c_{3,1} & c_{3,2} & 1 & c_{4,3} \\ c_{4,1} & c_{4,2} & c_{4,3} & 1 \end{bmatrix}, \tag{8}$$

where

$$\begin{aligned} c_{2,1} &= b_{2,1} \\ c_{3,1} &= b_{3,1} \\ c_{3,2} &= b_{4,1} \\ c_{4,1} &= b_{2,1}b_{3,1} + b_{2,2}b_{3,2} \\ c_{4,2} &= b_{2,1}b_{4,1} + b_{2,2}b_{4,2} \\ c_{4,3} &= b_{3,1}b_{4,1} + b_{3,2}b_{4,2} + b_{3,3}b_{4,3} \end{aligned} \tag{9}$$

which can be written in terms of the correlative angles as

$$\begin{aligned}
c_{2,1} &= \cos\theta_{2,1} \\
c_{3,1} &= \cos\theta_{3,1} \\
c_{3,2} &= \cos\theta_{4,1} \\
c_{4,1} &= \cos\theta_{2,1}\cos\theta_{3,1} + \sin\theta_{2,1}\cos\theta_{3,2}\sin\theta_{3,1} \\
c_{4,2} &= \cos\theta_{2,1}\cos\theta_{4,1} + \sin\theta_{2,1}\cos\theta_{4,2}\sin\theta_{4,1} \\
c_{4,3} &= \cos\theta_{3,1}\cos\theta_{4,1} + \cos\theta_{3,2}\sin\theta_{3,1}\cos\theta_{4,2}\sin\theta_{4,1} \\
&\quad + \sin\theta_{3,2}\sin\theta_{3,1}\cos\theta_{4,3}\sin\theta_{4,2}\sin\theta_{4,1}
\end{aligned} \quad . \quad (10)$$

Boundaries of the correlation coefficients

As shown in (6) to (10), a valid correlation matrix can be constructed from the matrix B, and the elements in B are determined by the correlative angles. Consequently, we can determine which elements of B are impacted by changes to the correlative angle in a four-dimensional correlation matrix, from which two important aspects can be inferred:

1. Correlation coefficients in the first column ($c_{i,1}$) depend solely on $\theta_{i,1}$.
2. Other correlation coefficients ($c_{i,j}$) for $j \geq 2$) can be calculated if $\theta_{p,q}$ are given, where $p \leq i$ and $q \leq j < i$.

Because all $\theta_{i,j}$ are in the closed interval [0, π], the sine functions will produce non-negative values, whereas the cosine functions will output values in the range [-1, 1]. Using the correlation coefficients in (10) as an example, it is straight forward to conclude that the boundaries of each correlation coefficient ($c_{i,j}$) can be calculated by setting $\cos\theta_{i,j}$ to -1 or 1. Moreover, the boundaries require only $\theta_{p,q}$ where $p \leq i$ and $q \leq j < i$, except for $p = i$ and $q = j$ (although not every $\theta_{p,q}$ is required), as shown in Table 1. As a result, if $c_{i,j}$ lies within its boundaries and the required $\theta_{p,q}$ are given, $\theta_{i,j}$ can be calculated by (11).

$$\begin{aligned}
\theta_{2,1} &= \arccos(c_{2,1}) \\
\theta_{3,1} &= \arccos(c_{3,1}) \\
\theta_{4,1} &= \arccos(c_{4,1}) \\
\theta_{3,2} &= \arccos\left(\frac{c_{3,2} - b_{2,1}b_{3,1}}{b_{2,2}\sin\theta_{3,1}}\right) \\
\theta_{4,2} &= \arccos\left(\frac{c_{4,2} - b_{2,1}b_{4,1}}{b_{2,2}\sin\theta_{4,1}}\right) \\
\theta_{4,3} &= \arccos\left(\frac{c_{4,3} - (b_{3,1}b_{4,1} + b_{3,2}b_{4,2})}{b_{3,3}\sin\theta_{4,2}\sin\theta_{4,1}}\right)
\end{aligned} \quad . \quad (11)$$

The same logic can easily be applied to higher-dimensional correlation matrices, albeit that longer formulas and computational procedures are obtained.

Algorithm for constructing a random correlation matrix

This section describes an algorithm to obtain a correlation matrix by sequentially computing the boundaries of each correlation coefficient, as described in earlier section, and generating uniform random variables (other bounded distributions can always be substituted) within these boundaries. Nevertheless, it is important to note that no optimization is needed to calculate the boundaries of each correlation coefficient. This non-optimization approach is the major difference between our work and that from presented in [11]. Let [0, 1] be the strictly lower triangular matrix of uniform random variables in the closed interval [0, 1], θ be the strictly lower triangular matrix of correlative angles, and Y and Z be the strictly lower triangular matrix of lower and upper bounds of the correlation coefficients, respectively. The four-step algorithm for constructing an $n \times n$ correlation matrix is then:

Step 1: Calculate correlation coefficients in the first column

- For $i = 1, \ldots, n$, set $c_{i,1} = -1 + 2 \times u_{i,1}$, $b_{i,1} = c_{i,1}$, and extract $\theta_{i,1}$.
- For $i = 2, \ldots, n$, set $b_{i,j} = \sin\theta_{i,1}$ for $j = 1, \ldots, i$.

Step 2: Calculate the remaining correlation coefficients from the third row to the last row and from the second column to the last column of each row.

For $i = 3, \ldots, n$
For $j = 2, \ldots, i-1$

- Calculate the lower bound ($y_{i,j}$) and upper bound ($z_{i,j}$) of each correlation coefficient.
- The method for calculating these boundaries is explained in the earlier section. Please see Table 1 for an example of the upper and lower bounds using a four-dimensional correlation matrix.
- If $z_{i,j} - y_{i,j} < K$, then $c_{i,j} = y_{i,j} + (z_{i,j} - y_{i,j})/2$. Otherwise, using $c_{i,j} = y_{i,j} + (z_{i,j} - y_{i,j}) \times u_{i,j}$.
- During our large numerical experiment, numerical instability occurs when the boundary gap ($z_{i,j} - y_{i,j}$) becomes very small. As a result a threshold factor (K) is introduced. This reduces instability by forcing every correlation coefficient with a boundary gap of less than K to be centered within its boundaries. Larger value of K will produce a more stable system, but imply less randomization in the $c_{i,j}$.
- Extract $\theta_{i,j}$ using similar formulas to those shown in (11).

End
End

- Create a symmetric correlation matrix with unit diagonal elements based on all generated correlation coefficients.

Step 3: Randomly reorder the correlation matrix. The underlying concept of this step is to ensure that every correlation coefficient is equally distributed. Without this step, the cumulative distribution function (CDF) of correlation coefficients will not be the same (see Figure 1). After applying random reordering, the CDF of the same correlation coefficients will be almost identical, as displayed in Figure 2.

Step 4: Check the validity of the correlation matrix. Even though the above steps should theoretically generate a valid correlation matrix, in some cases numerical instability can still occur. We can detect two major causes of instability: Firstly, K is too low relative to the dimension of matrix; Secondly, generated correlation coefficients are very close to the boundaries. Based on our experiments, in which 1 million 100×100 correlation matrices were generated with $K = 0.01$, there is only 0.0167% (or 167 matrices) probability that an invalid correlation matrix will occur. Although the probability of an invalid matrix is very small, it is non-zero. That is why this step is necessary, to ensure that invalid correlation matrices will be rejected. The two basic procedures of this step are:

1. Check the minimum eigenvalue. If it is negative, the correlation matrix is invalid. Otherwise, the correlation matrix is valid.
2. Reject the invalid correlation matrix, and regenerate the correlation matrix by returning to step 1

Table 1. Boundaries of each correlation coefficient in a 4×4 matrix.

$c_{i,j}$	Lower bound	Upper Bound	Required $\theta_{p,q}$
$c_{2,1}$	-1	1	No
$c_{3,1}$	-1	1	No
$c_{4,1}$	-1	1	No
$c_{3,2}$	$\cos\theta_{2,1}\cos\theta_{3,1}-\sin\theta_{2,1}\sin\theta_{3,1}$	$\cos\theta_{2,1}\cos\theta_{3,1}+\sin\theta_{2,1}\sin\theta_{3,1}$	$\theta_{2,1},\theta_{3,1}$
$c_{4,2}$	$\cos\theta_{2,1}\cos\theta_{4,1}-\sin\theta_{2,1}\sin\theta_{4,1}$	$\cos\theta_{2,1}\cos\theta_{4,1}+\sin\theta_{2,1}\sin\theta_{4,1}$	$\theta_{2,1},\theta_{4,1}$
$c_{4,3}$	$\cos\theta_{2,1}\cos\theta_{4,1}+\cos\theta_{3,2}$ $\sin\theta_{3,1}\cos\theta_{4,2}\sin\theta_{4,1}-\sin\theta_{3,2}\sin\theta_{3,1}$ $\sin\theta_{4,2}\sin\theta_{4,1}$	$\cos\theta_{2,1}\cos\theta_{4,1}+\cos\theta_{3,2}\sin\theta_{3,1}$ $\cos\theta_{4,2}\sin\theta_{4,1}+\sin\theta_{3,2}\sin\theta_{3,1}$ $\sin\theta_{4,2}\sin\theta_{4,1}$	$\theta_{3,1}, \theta_{4,1}, \theta_{3,2}, \theta_{4,2}$

doi:10.1371/journal.pone.0048902.t001

In addition, from (1) to (4), we can generate a valid correlation matrix directly from random sample of correlative angles. Unfortunately, based on our experiment, this direct method is not numerically stable. As a result, one may not be able to use the matrix generated from this method in some applications. Thus, we believe that our new algorithm is superior in terms of numerical stability.

Example 2. For a five-dimensional correlation matrix, let us assume that the uniform random matrix U described in step 1 of the algorithm is:

$$U=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.6220 & 0 & 0 & 0 & 0 \\ 0.0751 & 0.8576 & 0 & 0 & 0 \\ 0.9668 & 0.6035 & 0.4107 & 0 & 0 \\ 0.6100 & 0.8478 & 0.7324 & 0.7571 & 0 \end{bmatrix}. \quad (12)$$

The lower-bound matrix Y, upper-bound matrix Z, and correlation matrix C (before being randomly reordered) can then be generated as follows:

$$Y=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1.0000 & 0 & 0 & 0 & 0 \\ -1.0000 & -0.7185 & 0 & 0 & 0 \\ -1.0000 & -0.1197 & -0.8946 & 0 & 0 \\ -1.0000 & -0.8923 & -0.1893 & 0.0213 & 0 \end{bmatrix}, \quad (13)$$

$$Z=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 \\ 1.0000 & 0.3038 & 0 & 0 & 0 \\ 1.0000 & 0.5753 & -0.6363 & 0 & 0 \\ 0.6100 & 0.8478 & 0.7324 & 0.7571 & 0 \end{bmatrix}, \quad (14)$$

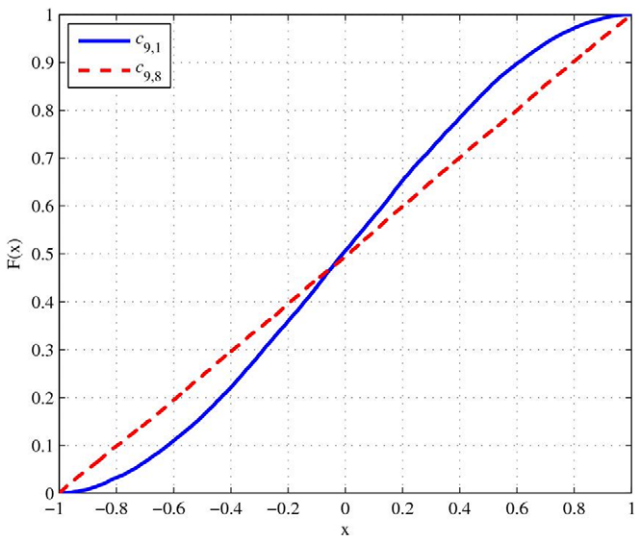

Figure 1. CDF from the proposed algorithm without random reordering.
doi:10.1371/journal.pone.0048902.g001

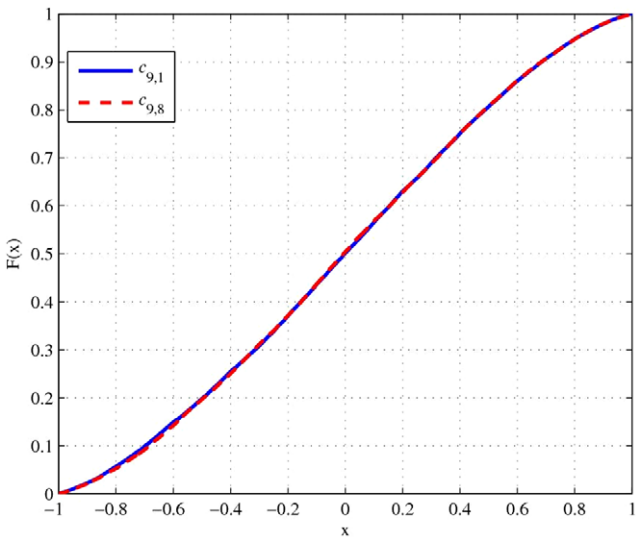

Figure 2. CDF from the proposed algorithm with random reordering.
doi:10.1371/journal.pone.0048902.g002

Table 2. Comparison of computational performance.

	$P_{valid}(\%)$			T_{avg} (ms)			T_{exp} (ms)		
n	NA	RS	RC	NA	RS	RC	NA	RS	RC
2	100	100	100	0.0492	0.0149	0.3819	0.0492	0.0149	0.3819
3	100	61.678	100	0.0710	0.0185	0.4720	0.0710	0.0300	0.4720
4	100	18.2341	100	0.0900	0.0204	0.5688	0.0900	0.1121	0.5688
5	100	2.1723	100	0.1164	0.0229	0.6521	0.1164	1.0532	0.6521
6	100	0.1009	100	0.1501	0.0254	0.7472	0.1501	25.19	0.7472
7	100	0.001	100	0.1827	0.0385	0.8567	0.1827	3,849.7	0.8567
8	100	0	100	0.2306	0.0321	0.9669	0.2306	Inf.	0.9669
9	100	0	100	0.2804	0.0355	1.1653	0.2804	Inf.	1.1653
10	100	0	100	0.3304	0.0404	1.2686	0.3304	Inf.	1.2686
11	100	0	100	0.4039	0.0449	1.2318	0.4039	Inf.	1.2318
12	100	0	100	0.4586	0.0485	1.3230	0.4586	Inf.	1.3230
13	100	0	100	0.5513	0.0546	1.4448	0.5513	Inf.	1.4448
14	100	0	100	0.6138	0.0589	1.5067	0.6138	Inf.	1.5067
15	100	0	100	0.6987	0.0647	1.6531	0.6987	Inf.	1.6531
16	100	0	100	0.7788	0.0785	1.7076	0.7788	Inf.	1.7076
17	100	0	100	0.8957	0.0811	1.8294	0.8957	Inf.	1.8294
18	100	0	100	1.0106	0.0873	1.9429	1.0106	Inf.	1.9429
19	100	0	100	1.0990	0.0907	2.0996	1.0990	Inf.	2.0996
20	100	0	100	1.2094	0.0974	2.2008	1.2094	Inf.	2.2008
21	100	0	100	1.3406	0.1051	2.2840	1.3406	Inf.	2.2840
22	100	0	100	1.4722	0.1132	2.3952	1.4722	Inf.	2.3952
23	100	0	100	1.6269	0.1217	2.5304	1.6269	Inf.	2.5304
24	100	0	100	1.7746	0.1296	2.6631	1.7746	Inf.	2.6631
25	100	0	100	1.9446	0.1393	2.7386	1.9446	Inf.	2.7386
26	100	0	100	2.1356	0.1492	2.8582	2.1356	Inf.	2.8582
27	100	0	100	2.2533	0.1585	2.9899	2.2533	Inf.	2.9899
28	100	0	100	2.4576	0.1689	3.0942	2.4576	Inf.	3.0942
29	100	0	100	2.6411	0.1806	3.2981	2.6411	Inf.	3.2981
30	100	0	100	2.8306	0.1904	3.4048	2.8306	Inf.	3.4048
35	100	0	100	3.9381	0.3315	4.0185	3.9381	Inf.	4.0185
40	100	0	100	5.3749	0.3971	4.7135	5.3749	Inf.	4.7135
45	100	0	100	6.8185	0.5067	5.7925	6.8185	Inf.	5.7925
50	100	0	100	8.5822	0.6172	8.5822	8.5822	Inf.	6.9464

Note: Inf. denotes infinity.

doi:10.1371/journal.pone.0048902.t002

$$C=\begin{bmatrix} 1.0000 & 0.2440 & -0.8498 & 0.9336 & 0.2200 \\ 0.2440 & 1.0000 & 0.1582 & 0.2997 & 0.7117 \\ -0.8498 & 0.1582 & 1.0000 & -0.7885 & 0.1889 \\ 0.9336 & 0.2997 & -0.7885 & 1.0000 & 0.3454 \\ 0.2200 & 0.7117 & 0.1889 & 0.3454 & 1.0000 \end{bmatrix}. \quad (15)$$

As the minimum eigenvalue of C in (15) is 0.00510, the correlation matrix is positive semi-definite. This confirms that C is a valid correlation matrix.

Results

All numerical tests in this study were conducted with MATLAB 7.8.0 (R2009a) on an Intel(R) Core™ 2 Duo CPU T6600 at 220 GHz with 3.50 GB of RAM. The computational performance and probability distribution function (PDF) of the proposed algorithm (NA) with $K = 0.01$ was evaluated and compared with the following two algorithms:

1. Rejection sampling method (RS)

   The rejection sampling method uses uniform random variables in the closed interval $[-1, 1]$ to represent each correlation coefficient in the symmetric correlation matrix. The correlation matrix will be rejected if it is invalid.

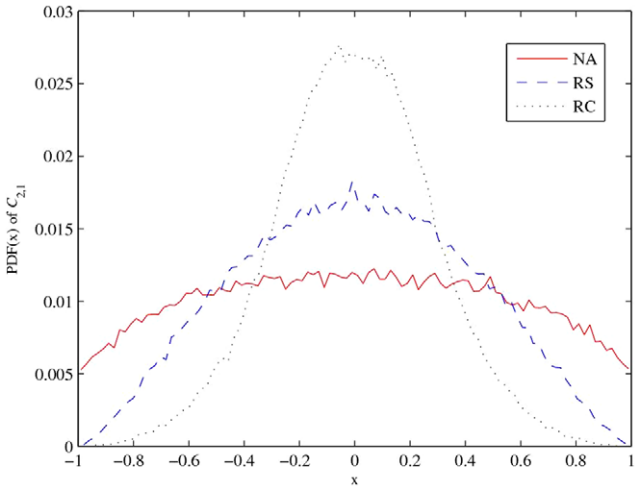

Figure 3. PDF of correlation coefficient ($C_{2,1}$).
doi:10.1371/journal.pone.0048902.g003

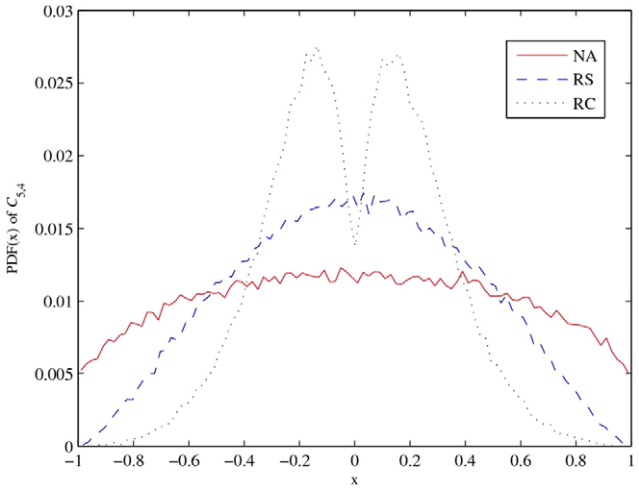

Figure 4. PDF of correlation coefficient ($C_{5,4}$).
doi:10.1371/journal.pone.0048902.g004

2. Randcorr function of MATLAB (RC)
   This algorithm is implemented as a MATLAB function, and is based on the work in [6] and [7].

The MATLAB code for the NA algorithm (denoted as RandomCorr) is available at http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId = 37804. The following MATLAB code was used to generate the correlation matrices (C) in the RS algorithm and to check their validity: C = tril(−1+2*round(rand(n,n)*10^8)/(10^8),−1);C = C+C′+ eye(-n);p = min(eig(C));. And the following MATLAB code was used to generate the correlation matrices (C) in the RC algorithm and to check their validity: C = gallery('randcorr',n);p = min(eig(C));.

Computational performance

The computational performance of each algorithm is primarily measured by the expected run time (T_{exp}), which can be calculated from the average run time (T_{avg}) divided by the probability of the generated correlation matrix being valid (P_{valid}). T_{avg} includes the time taken to construct the correlation matrix and calculate the minimum eigenvalue. The performance summary of the three algorithms over 1 million simulations is illustrated in Table 2.

With a P_{valid} score of 100% in all cases, both NA and RC algorithms are evidently stable. Moreover, the RC algorithm has the fastest expected run time when the dimension exceeds 35, although the RS algorithm is the fastest for dimensions of two and three. However, the RS method then becomes slower than the NA algorithm when $n \geq 4$, and slower than RC for $n \geq 5$. Even worse, the RS method cannot generate a valid correlation matrix for dimensions larger than seven, mainly due to the significant drop in Pvalid. Hence, the RS method is not very useful in practice. For dimensions from 4–35, the NA algorithm outperforms RS and RC in terms of expected run time.

Probability distribution function

To compare the PDF of the coefficients of correlation matrices, $c_{2,1}$ and $c_{5,4}$ are drawn from 100,000 valid 5×5 correlation matrices constructed by the above algorithms. Comparing Figures 3 and 4, we can clearly see that the correlation coefficients generated by the RC algorithm have significant differences.This fact is verified by the kurtosis and standard deviation of the RC algorithm, which are given in Table 3. In general, correlation coefficients from the NA and RC algorithms are equally distributed, but the NA algorithm produces a higher standard deviation and lower kurtosis, which implies more extreme correlation coefficients than the other algorithms.

Discussion

In this paper, we have presented an efficient method to calculate the boundaries of correlation coefficients. We also demonstrated a

Table 3. Statistical summary of random correlation coefficients ($c_{2,1}$ and $c_{5,4}$).

	$c_{2,1}$			$c_{5,4}$		
Statistical measure	**NA**	**RS**	**RC**	**NA**	**RS**	**RC**
Mean	−0.001	−0.0001	−0.0009	0.0004	−0.0004	0.001
Median	−0.0024	−0.0001	−0.0015	−0.0013	−0.0006	0.0011
Standard Deviation	0.5289	0.4079	0.2779	0.5281	0.4086	0.2901
10^{th} Percentile	−0.7288	−0.5515	−0.3536	−0.7268	−0.5528	−0.3667
90^{th} Percentile	0.7301	0.5503	0.3517	0.7297	0.5516	0.3697
Skewness	0.0062	0.0012	−0.0015	0.0027	−0.0018	0.009
Kurtosis	1.9421	2.2551	3.0207	1.9444	2.2496	2.6727

doi:10.1371/journal.pone.0048902.t003

technique for generating correlation matrices using any bounded random variable distribution within the boundaries of each correlation coefficient. However, this method causes the correlation coefficients to be unevenly distributed. Thus, we incorporated a technique for random reordering to ensure the even distribution of all correlation coefficients. The performance of the proposed algorithm was compared to that of other algorithms. It was shown that the new algorithm could efficiently construct correlation matrices, particularly when the dimension of the matrix was in the range 4–35. In theory, our algorithm should always return valid correlation matrices. However, without setting a threshold factor and using rejection sampling logic, the algorithm exhibited some numerical instability when the dimension became large. It is possible to adjust invalid matrices to form valid ones; this method has been developed in many studies [16,17,18]. Therefore, we strongly believe that our new algorithm is useful in the many applications where extreme cases are very important. More importantly, the uniform distribution can be replaced with any bounded distribution.

Author Contributions

Conceived and designed the experiments: KN AA. Performed the experiments: KN. Analyzed the data: KN AA. Contributed reagents/materials/analysis tools: KN AA. Wrote the paper: KN AA.

References

1. Mittelbach M, Matthiesen B, Jorswieck E (2012) Sampling Uniformly From the Set of Positive Definite Matrices With Trace Constraint. IEEE Trans Signal Process 60: 2167–2179.
2. Hirschberger M, Qi Y, Steuer RE (2007) Randomly generating portfolio-selection covariance matrices with specified distributional characteristics. European J Oper Res 177: 1610–1625.
3. Tucker L, Koopman R, Linn R (1969) Evaluation of factor analytic research procedures bymeans of simulated correlation matrices. Psychometrika 34: 421–459.
4. Chu JH, Weiss S, Carey V, Raby B (2009) Gene Expression Network Reconstruction by Convex Feature Selection when Incorporating Genetic Perturbations. BMC Syst Biol 3: 55.
5. Zalesky A, Fornito A, Bullmore E (2012) On the use of correlation as a measure of network connectivity, NeuroImage 60: 2096–2016.
6. Bendel RB, Mickey MR (1978) Population correlation matrices for sampling experiments, Commun Statist Simul Comp B7: 163–182.
7. Davies PI, Higham NJ (2000), Numerically stable generation of correlation matrices and their factors. BIT 40: 640–651.
8. Dhillon I, Heath R, Sustik M, Tropp J (2005) Generalized Finite Algorithms for Constructing Hermitian Matrices with Prescribed Diagonal and Spectrums. SIAM J Matrix Anal Appl 27: 67–71.
9. Marsaglia G, Olkin I (1984) Generating correlation-matrices. SIAM J Sci Statist Comput 5: 470–475.
10. Holmes RB (1991), On Random Correlation Matrices. SIAM J Matrix Anal Appl 12: 239–272.
11. Budden M, Hadavas P, Hoffman L (2008) On The Generation of Correlation Matrices. Appl Math E-Notes 8: 279–282.
12. Rebonat R, Jäckel P (2000) The most general methodology for creating a valid correlationmatrix for risk management and option pricing purposes. J Risk 2: 17–27.
13. Numpachaoren K, Bunwong K (2012) An intuitively valid algorithm for adjusting the correlation matrix in risk management and option pricing. SSRN website. Available: http://ssrn.com/abstract = 1980761. Accessed 2012 Oct 15.
14. Numpachaoren K, Bunwong K (2012) Boundaries of Correlation Adjustment with Applications to Financial Risk Management. Appl Math Finance: In Press.
15. Rapisarda F, Brigo D, Mercurio F (2007) Parameterizing correlations: a geometric interpretation, IMA J Manag Math 18: 55–73.
16. Higham NJ (2002) Computing the nearest correlation matrix-A problem from finance. IMA J Numer Anal 22: 329–343.
17. Li Q, Li D, Qi H (2010) Newton's Method for Computing the Nearest Correlation Matrix witha Simple Upper Bound. J Optim Theory Appl, 147: 546–568.
18. Simonian J (2010) The most simple methodology to create a valid correlation matrix for risk management and option pricing purposes. Appl Econ Lett 17: 1767–1768.