A Non-Simulation Based Method for Inducing Pearson's Correlation Between Input Random Variables

Eric R. Druker, Richard L. Coleman, Peter J. Braxton, Joel B. Hughes
Northrop Grumman – Information Technology

Eric R. Druker
Technical/Research Lead – Northrop Grumman IT (TASC)
4584 Emerald View Ct.
Eureka, MO 63025
Office: (636) 587-2624
Mobile: (703) 408-0589
Email: Eric.Druker@ngc.com

Richard L. Coleman
Sector Director – Northrop Grumman Cost/Price Analysis Center of Excellence
15036 Conference Center Dr.
Chantilly, VA 20151
Office: (703) 449-3627
Mobile: (703) 615-4482
Email: Richard.Coleman@ngc.com

Peter J. Braxton
Technical Fellow – Northrop Grumman IT (TASC)
15036 Conference Center Dr.
Chantilly, VA 20151
Office: (703) 961-3411
Mobile: (703) 944-3114
Email: Peter.Braxton@ngc.com

Joel B. Hughes
Sr. Financial Specialist – Northrop Grumman Shipbuilding
P.O. Box 149 M/S 1012-28
Pascagoula, MS 39568-0149
Phone: 228-935-2609
Email: Bart.Hughes@ngc.com

Abstract

Several previously published papers have cited the need to include correlation in risk analysis models. In particular, a landmark paper published by Philip Lurie and Matthew Goldberg presented a methodology for inducing Pearson's correlation between input/independent random variables. The one subject absent from the paper was a methodology for finding the optimal applied correlation matrix given a desired outcome correlation. Since the publishing of the Lurie-Goldberg paper there has been continuing discussion on its implementation, however there has not been any presentation of an optimization algorithm that does not involve the use of computing-heavy simulations. This paper reviews the general methodology used by Lurie and Goldberg (along with its predecessor papers) and presents a non-simulation approach to finding the optimal input correlation matrix given a set of marginal distributions and a desired correlation matrix.

Biographies

Eric R. Druker graduated from the College of William and Mary with a B.S. in Applied Mathematics in 2005 concentrating in both Operations Research and Probability & Statistics with a minor in Economics. He is employed by Northrop Grumman as a Technical & Research lead. He performs cost and risk analysis on several programs within both the Intelligence and DoD communities. He was a recipient of the 2005 NGIT President's award for his work on Independent Cost Evaluations during which he helped to develop the risk process currently used by NGIT's ICE teams. As a member of Northrop Grumman's ICE working group, he has helped shape the cost and risk practices used on independent cost estimates and evaluations across the corporation. In addition to SCEA conferences, Eric has also presented papers at the Naval Postgraduate School's Acquisition Research Symposium, DoDCAS and the NASA PM Challenge. He has also performed decision tree analysis for NG Corporate law and built models for Hurricane Katrina Impact Studies and Schedule/Cost Growth determination.

Richard L. Coleman is a 1968 Naval Academy graduate, received an M. S. with Distinction from the U. S. Naval Postgraduate School and retired from active duty as a Captain, USN, in 1993. His service included tours as Commanding Officer of *USS Dewey (DDG 45)*, and as Director, Naval Center for Cost Analysis. He has worked extensively in cost, CAIV, and risk for the Missile Defence Agency (MDA), Navy ARO, the intelligence community, NAVAIR, and the DD(X) Design Agent team. He has supported numerous ship programs including DD(X), the DDG 51 class, Deepwater, LHD 8 and LHA 6, the LPD 17 class, *Virginia* class submarines, CNN 77, and CVN 21. He is the Director of the Cost and Price Analysis Center of Excellence and conducts Independent Cost Evaluations on Northrop Grumman programs. He has more than 65 professional papers to his credit, including five ISPA/SCEA and SCEA Best Paper Awards and two ADoDCAS Outstanding Contributed Papers. He was a senior reviewer for all the SCEA CostPROF modules and lead author of the Risk Module. He has served as Regional and National Vice President of SCEA and is currently a Board Member.

Peter J. Braxton holds an AB in Mathematics from Princeton University and an M. S. in Applied Science (Operations Research) from the College of William and Mary.
He has worked to advance the state of knowledge of cost estimating, Cost As an Independent Variable (CAIV), Target Costing, and risk analysis on behalf of the Navy Acquisition Reform Office (ARO), the DD(X) development program, and other ship and intelligence community programs. He has co-authored several professional papers, including ISPA/SCEA International Conference award-winners in CAIV (1999) and Management (2005). He served as managing editor for the original development of the acclaimed Cost Programmed Review Of Fundamentals (CostPROF) body of knowledge and training course materials and is currently undertaking to lead a large team of cost professional in a comprehensive update thereof. He serves as SCEA's Director of Training and was recently appointed a Northrop Grumman Technical Fellow.

Joel B. Hughes received a B.S. in Finance from the University of South Alabama, along with an M.B.A from Tennessee Technological University. He is also beginning work towards an M.S. in Operations Management at the University of Alabama. Currently working as a Senior Business

Analyst, he has performed cost and risk analysis on several programs at Northrop Grumman Ship Systems sector. He is the recipient of multiple Business Management Awards, and was instrumental in initializing the risk quantification process at Ship Systems.

Table of Contents

Table of Figures

A Non-Simulation Based Method for Inducing Pearson's Correlation Between Input Random Variables

Introduction

The Complete Correlation Algorithm (CCA) developed by Northrop Grumman and recently implemented in NG developed risk models is a product of more than two years of research and development. Several previously published papers have cited the need to include correlation in risk analysis models, however none present an optimization algorithm that does not involve the use of computing-heavy simulations. In particular, a landmark paper published by Philip Lurie and Matthew Goldberg[1] presented a methodology for inducing Pearson's correlation between input random variables. This paper reviews the general methodology used by Lurie and Goldberg (along with its predecessor papers) and presents the Druker Algorithm: a non-simulation approach to finding the optimal input correlation matrix given a set of marginal distributions and a desired correlation matrix.

The *CCA* was deliberately created bearing in mind identified environmental factors that prevent easy implementation of commercially available models. No one on the team had any experience implementing correlation into Monte Carlo simulations beyond the use of COTS programs such as @Risk™ and Crystal Ball™. To determine the best development method, the following factors were considered:

1. The Northrop Grumman risk models need to be of an easily transferable electronic size, as the models are often shared via email or network drives.
2. A diverse group of users must be able to run the software in a variety of work environments; Microsoft Office is the only platform that is transferable to all parties. Users include risk practitioners, program managers and members of pricing organizations; locations include unclassified and classified Northrop Grumman facilities, unclassified and classified customer facilities and home offices.
3. Custom implementations are frequent; much of NGIT-TASC risk work requires risk simulations to be built into pre-existing cost and price models. These models are generally limited to Microsoft Excel and Access; however web-based platforms are not unheard of.

The above concerns drove the decision to use Visual Basic source code to develop *the CCA.*

Initially, the development was focused on an algorithm that could induce Pearson's correlation between typical distributions in risk analysis: Bernoulli (discrete), Triangular, Normal and Log-Normal. By limiting the problem to the most common applications, in theory, the solution should have been easier to find. While attempting to ascertain the maximum correlation between any two Bernoulli distributions, however, the general solution was uncovered. The resulting algorithm induces Pearson's correlation between any set of random variables (while still preserving the marginal distributions) using the Lurie-Goldberg method and without the use of simulation to find the optimal applied correlation matrix.

The *CCA* is a compilation of multiple algorithms (each named for their main author(s)) from several sources: existing papers, public source code and internally developed code. Most of the algorithms used were taken from a variety of existing papers. Although these papers all

[1] Goldberg, Matthew S., Lurie, Philip M., "An Approximate Method for Sampling Correlated Random Variables From Partially-Specified Distributions", Management Science, Volume 44, Issue 2, published by INFORMS, February 1998.

provided complete algorithms, they were sometimes lacking details in how to accomplish key steps; in cases such as these, gaps were filled in with open-source code solutions. The optimization of the applied correlation matrix, the last step in the correlation algorithm, was developed entirely by the Northrop Grumman team.

Definitions and Assumptions

Matrix Definitions:

1. **Consistent Correlation Matrix:** Consistent Correlation matrices have diagonal entries equal to 1.0, all other entries between [-1, 1] and are symmetric and positive definite. Consistency is necessary for a viable correlation matrix, but a Consistent Correlation Matrix may not necessarily be viable given the **Parent Distributions.**
2. **Input Correlation Matrix (I)** – The user-inputted correlation matrix. This matrix may or may not be a consistent correlation matrix.
3. **Adjusted Correlation Matrix (L)** – The **Input Correlation Matrix** adjusted to be a **Consistent Correlation Matrix.** This matrix will, by definition, be positive definite. Additionally, the adjusted matrix will be viable as correlations between various distributions of random variables will be achievable. When **(L)** is generated, the differences between **(I)** and **(L)** are minimized.
4. **Applied Correlation Matrix (A)** – The correlation matrix used by the *grand algorithm* to generate correlated random number draws. This matrix may be the same, or very different from, the **Adjusted Correlation Matrix;** the extent of the differences will depend on the random variables to be correlated.
5. **Optimal Applied Correlation Matrix (A')** – The Applied Correlation Matrix optimized using the Lurie-Goldberg method.
6. **Outcome Correlation Matrix (O)** – The correlation matrix of the simulated variables following the simulation run. The goal of the *grant correlation algorithm* is for (O) to be identical to (L).

Other Definitions:

1. **Parent Distribution –** The distributions correlated for use in the simulation. The distributions are simulated using the Inverse CDF technique. The goal is to induce a desired correlation between these distributions.
2. **Pearson's Correlation –** A parametric statistic that measures the *strength and direction of a linear relationship between two random variables.*[2]
3. **Spearman's Rank Correlation** - A non-parametric statistic that measures the monotonicity of a function without making any assumptions as to the distribution of the variables.
4. **Eigenvalues** – A scalar (L) associated with a matrix such that if if (A) is a matrix and (X) is a vector, AX = LX. The vector (X) is known as the **Eigenvector** that corresponds to the **Eigenvalue** (L).

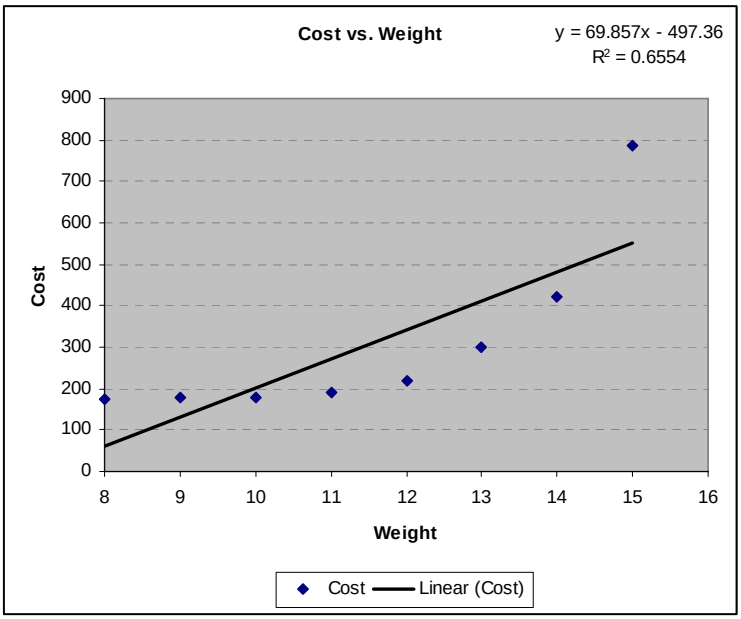
Assumptions:

1. **Normal Distributions –** Any reference to the normal distribution, whether in a univariate or bivariate case, is assumed to be the Standard Normal distribution (Mean of 0, Standard Deviation of 1).

[2] Definition taken from Wikipedia: http://en.wikipedia.org/wiki/Correlation

Pearson's vs. Rank Correlation

Most COTS risk tools use Spearman's rank correlation as a substitute for Pearson's correlation between parent distributions. Spearman's rank correlation (a non-parametric statistic) differs from Pearson's correlation (a parametric statistic) in that it measures the monotony of a function whereas Pearson's correlation measures the strength of the linear relationship between two functions (see figure 1). Though studies have shown that, using the most common risk distributions, models using rank correlation yield similar results to those using Pearson's[3], there is a distinct difference between the two. Although this paper will not detail all of the differences between the two measures, a quick (and exaggerated) example is presented below. The *grand algorithm* supersedes the need to substitute for Pearson's correlation with Spearman's rank correlation.



Pearson's Rho	**0.81**
Spearman's Rho	**1.00**

Figure 1- Pearson's vs. Spearman's Rank Correlation

Algorithm Overview

There are three main steps behind the *grand algorithm.* An outline of these steps follows; the upcoming sections of this paper will review each individual step in detail.

1. Correct the **User-Input Correlation Matrix (I)**
   a. Correct **I** so that it is consistent; both in terms of a general correlation matrix and the properties of the parent distributions being correlated.
   b. Through these corrections, the Adjusted Correlation Matrix **(L)** will be generated.
2. Optimize the **Applied Correlation Matrix**

[3] Robinson, M and Salls, W. More on Correlation Accuracy in Crystal Ball Simulations (or What We've Now Learned about Spearman's R in Cost Risk Analyses). Presented at the 2004 SCEA Conference, Manhattan Beach, CA, June 2004

   a. Find the **Optimal Applied Correlation Matrix (A')** such that when **A'** is run through the Lurie-Goldberg method, the **Outcome Correlation Matrix (O)** is identical to **L**.
3. Correlate the Input Random Variables
   a. Using **A'**, apply the Lurie-Goldberg method to correlate the parent distributions.

For purposes of presenting the methodology, it is necessary to show how the input random variables are to be correlated before discussing how to find **A'**.

Correcting the User-Input Correlation Matrix (Part I)

Giving a user the ability to input their own correlation matrix allows for the possibility that the **User-Input Correlation Matrix (I)** may not be a viable correlation matrix. Correlation matrices, by definition, have diagonal entries of 1.0, all other entries between [-1, 1], are symmetric and are positive definite. The first step in inducing correlation between input random variables is checking whether or not **I** is a consistent correlation matrix. If it is not, it must be corrected such that it is one.

The *Iman-Davenport Algorithm*, which is based on a paper[4] by Ronald Iman and James Davenport, is used to correct **I** to make it a consistent correlation matrix. While numerous other papers have been published describing methods to correct **I** such that it is altered as little as possible[5], the *Iman-Davenport Algorithm* is the most computationally efficient one the authors uncovered. Given that additional adjustment may be required based on the parent distributions being correlated; the resulting matrix is close enough to **I** to satisfy this requirement.

The algorithm corrects **I** in three main phases. First, the algorithm checks whether **I** is symmetric with diagonal entries of 1.0 and off-diagonal entries between [-1, 1]. If **it** is not, the user is prompted to re-input the matrix, correcting for the discrepancies.

Second, once the above conditions are satisfied, the algorithm checks whether **I** is positive-definite. One way to test for this is to find the eigenvalues for **I** (positive-definite matrices have all positive eigenvalues). The paper referenced did not describe an approach for finding the eigenvalues of the matrix; after further research, the Jacobi Eigenvalue Algorithm was determined to be a sufficiently efficient way to evaluate a matrix's eigenvalues. As a result of the algorithm, the eigenvalues are produced as the diagonals of an otherwise zero-matrix. The Jacobi Eigenvalue Algorithm is computationally inexpensive and pre-existing source code was used in its implementation.

If all eigenvalues for **I** are positive and the other conditions have been satisfied, then **I** is a consistent correlation matrix. Otherwise, in the third phase, negative eigenvalues are changed to small, positive values (.000001 for example). The diagonal matrix of adjusted eigenvalues is then multiplied by the associated matrix of eigenvectors (also produced using the Jacobi Eigenvalue Algorithm(. That product is in turn multiplied by the inverse of the matrix of eigenvectors to arrive at a new matrix that is similar, but not equal to, **I**. Lastly, the diagonals are re-set to 1.0 as they may have changed during the transformation. This third section of the algorithm is repeated until all eigenvectors of the adjusted matrix are positive. At this point, the user input matrix has been adjusted such that it is a consistent correlation matrix.

---

[4] Iman, Ronald L. Davenport, James M. *An Iterative Approach to Produce a Positive Definite Correlation Matrix from an "Approximate Correlation Matrix"*. Sandia National Laboratories, June 1982

[5] Higham, N. *Computing the Nearest Correlation Matrix – A Problem from Finance.* IMA Journal of Numerical Analysis. 2002

Though the **User-Input Correlation Matrix** is now a consistent correlation matrix, the transformation of **I** is not complete and the **Adjusted Correlation Matrix (L)** has not been determined. As will be shown later, depending on the parent distributions being correlated, there may be a maximum achievable correlation between any two of the variables. Determination of **L** will be covered later in the section: *Correcting the User-Input Correlation Matrix (Part II)*.

Correlating Input Random Variables

In order to understand how the **Applied Correlation Matrix (A)** is to be optimized such that the **Output Correlation Matrix (O)** is identical to the **Adjusted Correlation Matrix (L)**, the method for correlating the parent distributions must first be discussed. It is a well known fact that normal random variables can be correlated by multiplying a vector of uncorrelated normal random draws by the Cholesky decomposition[6] of the desired correlation matrix. The Lurie-Goldberg method takes this one step further using normal random variates to generate correlated uniform random variates. These uniform random variates are then transformed via the inverse-CDF technique to generate draws from the desired parent distributions. In this method, although the correlations between the normal random draws are known, as these draws are transformed into other distributions, the correlations change. Hence, the core problem: how can the **Optimal Applied Correlation Matrix (A')** be uncovered such that **O** matches **L**? Answering this question is the key to implementing the Lurie-Goldberg method. The authors have developed an algorithm that addresses this very question, without necessitating any runs of the simulation. Additionally, they have found began the process of optimizing this algorithm, finding heuristics that allow it to run with a minimal number of calculations.

Implementation and Application of the *CCA*

The *CCA's* chief advantage is that it is non-recurring and its implementation requires no simulation. Furthermore, because the algorithm only requires looking at pairs of parent distributions, once the applied matrix has been found for a set of parent distributions, the algorithm must only be run when distributions are added or changed, and even then, only for the new/altered distributions. The algorithm also uses Pearson's correlation while COTS risk tools substitute Spearman's rank correlation.

The applications of the *CCA* reach beyond the Cost and Risk analysis community; this algorithm is useful anywhere there is a need to induce Pearson's correlation between input variables. For example, this algorithm can applied to auto correlating stock market projections in the financial arena and to traditional modeling and simulation situations when correlation is needed. The first implementation of the algorithm is does not involve cost risk but rather the modeling of conditional probabilities between Bernoulli random variables. The algorithm was designed with a focus on portability; because algorithm is coded with Visual Basic, it can be easily integrated in existing tools and models.

[6] The Cholesky Decomposition Matrix of any matrix M is L such that $M = LL^T$

References

Iman, R and Davenport J. *An Intterative Algorithm to Produce a Positive Definite Matrix from an "Approximated Correlation Matrix" (With a Program User's Guide)* Sandia National Laboratories for the US DoE, June 1982

Higham, N. *Computing the Nearest Correlation Matrix – A Problem from Finance.* IMA Journal of Numerical Analysis. 2002

Robinson, M and Salls, W. *More on Correlation Accuracy in Crystal Ball Simulations (or What We've Now Learned about Spearman's R in Cost Risk Analyses).* Presented at the 2004 SCEA Conference, Manhattan Beach, CA, June 2004

Goldberg, Matthew S., Lurie, Philip M., "An Approximate Method for Sampling Correlated Random Variables From Partially-Specified Distributions", *Management Science*, Volume 44, Issue 2, published by INFORMS, February 1998.

Goldberg, Matthew S, Lurie, Phillip M. *Correlating Random Variables*, 32nd DoDCAS, Williamsburg, VA. February 1999

Open Source Code References:

The Foxes Team, Italy - http://digilander.libero.it/foxes
Axel Vogt, Germany - http://www.axelvogt.de/axalom/bivariateNormal_Series.zip