# Nearest Correlation Matrix

The NAG Library has a range of functionality in the area of computing the *nearest correlation matrix*. In this article we take a look at nearest correlation matrix problems, giving some background and introducing the routines that solve them.

## Introduction

A correlation matrix is a real, square matrix that

- is symmetric
- has 1's on the diagonal
- has non-negative eigenvalues, it is positive semidefinite.

If a matrix $C$ is a correlation matrix then its elements, $c_{ij}$, represent the pair-wise correlation of entity $i$ with entity $j$, that is, the strength and direction of a linear relationship between the two.

In the literature there are numerous examples illustrating the use of correlation matrices but the one we have encountered the most arises in finance where the correlation between various stocks is used to construct sensible portfolios. Unfortunately, for a variety of reasons, an input matrix which is supposed to be a correlation matrix may fail to be semidefinite. The correlations may be between stocks measured over a period of time and some data may be missing, for example. If individual correlations are computed using observation data the two variables have in common, and the particular observations vary over all the variables, it will give rise to a matrix that is not positive semidefinite. Still drawing from finance, a practitioner may wish to explore the effect on a portfolio of assigning correlations between certain assets different from those computed from historical values. This can also result in negative eigenvalues in the computed matrix.

In such situations, the user has a matrix which is an *approximates correlation matrix* and this must be fixed for subsequent analysis that relies upon having a *true correlation matrix* for the results to be valid. It is thus natural to seek a neighbouring true matrix which differs least, in some measure of nearness, from the input matrix to act in its stead. This is our *basic nearest correlation matrix problem*.

## The Basic Nearest Correlation Matrix Problem

The NAG routine g02aa implements a Newton algorithm to solve our basic problem. It finds a true correlation matrix $X$ that is closest to the approximate input matrix, $G$, in the Frobenius norm; that is, we find the minimum of

$$||G - X||_F$$

The algorithm, described in a paper by Qi and Sun [6], has a superior rate of convergence compared to previously suggested approaches. Borsdorf and Higham [2], at the University of Manchester, looked at this in greater detail and offered further improvements. These include a different iterative solver (MINRES was preferred to Conjugate Gradient) and a means of pre-conditioning the linear equations. We have incorporated this enhanced algorithm into our library releases.

## Weighted Problems and Forcing a Positive Definite Correlation Matrix

In NAG routine g02ab we extend the functionality provided by g02aa. If we have an approximate correlation matrix it is reasonable to suppose that part of it may actually be true. Similarly, we may trust some correlations more than others and wish for these to stay closer to their input value in the final matrix.

In this routine we apply the original work of Qi and Sun to now use a *weighted* norm. Thus, we find the minimum of

$$\left\| W^{\frac{1}{2}}(G - X)W^{\frac{1}{2}} \right\|_F$$

Here $W$ is a diagonal matrix of weights. This means that we are seeking to minimize the elements $\sqrt{w_{ii}}(g_{ij}-x_{ij})\sqrt{w_{jj}}$. Thus, by choosing elements in $W$ appropriately we can favour some elements in $G$, forcing the corresponding elements in $X$ to be closer to them.

This method means that whole rows and columns of $G$ are weighted. However, g02aj allows element-wise weighting and in this routine we find the minimum of

$$\|H \circ (G - X)\|_F$$

where $C = A \circ B$ denotes the matrix $C$ with elements $c_{ij} = a_{ij}b_{ij}$. Thus by choosing appropriate values in $H$ we can emphasize individual elements in $G$ and leave the others unweighted. The algorithm employed here is one by Jiang, Sun and Toh [5], and has the Newton algorithm at its core.

Both g02ab and g02aj allow the user to specify that the computed correlation matrix is positive definite; that is, that its eigenvalues are greater than zero. This is required for further analysis in some applications.

### Fixing Correlations with a Shrinking Method

We now turn our attention to *fixing* some of the elements that are known to be true correlations. Instead of using a Newton method like the previous three algorithms, here we use a *shrinking* method.

One common example where this is needed is where the correlations between a subset of our variables are trusted and on their own would form a valid correlation matrix. We could thus arrange these into the leading block of our input matrix and seek to fix them while we correct the remainder. We call this the *fixed block* problem. The routine g02an preserves such a leading block of correlations in our approximate matrix. Using the shrinking method of Higham, Strabić and Šego [4], the routine finds a true correlation matrix of the following form

$$\alpha \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} + (1-\alpha)G.$$

$G$ is again our input matrix and we find the smallest $\alpha$ in the interval [0,1] that gives a positive semidefinite result. The smaller the $\alpha$, the closer we stay to our original matrix, but any $\alpha$ preserves the leading submatrix $A$, which needs to be positive definite. The algorithm uses a bisection method which converges quickly in a finite number of steps.

The routine g02ap generalizes the shrinking idea and allows users to supply their own *targe*t matrix. The target matrix, $T$, is defined by specifying a matrix of weights, $H$, and $T = H \circ G$. We then find a solution of the form

$$\alpha T + (1-\alpha)G$$

computing $\alpha$ as before. A bound on the smallest eigenvalue can also be specified. Specifying a value of 1 in $H$ essentially fixes an element in $G$ so it is unchanged in $X$.

For example, it is sometimes required to fix two diagonal blocks, so we could choose *H* to be

$$H = \begin{pmatrix} \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} & 0 \\ 0 & \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \end{pmatrix}.$$

The algorithm then finds the smallest $\alpha$ that gives a positive semidefinite matrix of the following form

$$\alpha \begin{pmatrix} G_{11} & 0 \\ 0 & G_{22} \end{pmatrix} + (1-\alpha)G$$

and we perturb only the two off-diagonal blocks of the input.

## Choosing a Nearest Correlation Matrix Routine

When choosing a routine, the trade-off is between computation time and the distance of the solution from the original matrix. The Newton algorithm based routines (g02aa, g02ab and g02aj) will, in general, find a solution nearer to the input than the shrinking routines (g02an and g02ap). However, the shrinking routines will be much quicker.

For the basic problem g02aa will always find the nearest matrix. Using g02ap with an identity matrix as the target will produce a matrix further away than this, which is understandable given the form of the solution, but with a shorter computation time.

If you wish to solve the fixed block problem the specialist routine g02an will be the fastest. Of the Newton routines g02ab will find a solution in reasonable time but, as we weight whole rows and columns, some elements will be over-emphasized outside of the correct block. A more accurate weighting can be achieved with g02aj and a close solution will be found. However, the routine will take considerably more time.

For fixing two diagonal blocks, or for arbitrary fixing and weighting, the choice is between g02aj and g02ap with the same speed and nearness trade-off. Recall, though, that the shrinking algorithm is strictly fixing elements and the target matrix is required to be positive definite and form part of a valid correlation matrix. This means that g02aj may offer more flexibility here.

If we seek to fix the minimum eigenvalue, and no weighting is required, g02ab or g02ap can be used. In this case the latter should use an identity target, as for the basic problem. If weighting or fixing is also required, then similar results are found for the problems described above. However, in combination with weighting g02ap can return a large value of $\alpha$, which means that much of the input matrix has been lost and a result far from it is returned.

The tolerance used in all the algorithms, which defines convergence, can obviously affect the number of iterations undertaken and thus the speed and the nearness. We recommend some experimentation using data that represents your typical problem. The routine g02aj can be sensitive to the weights used, so different values should be tried to tune both the nearness and the computation time.

## The Nearest Correlation Matrix with Factor Structure

A correlation matrix with *factor structure* is one where the off-diagonal elements agree with some matrix of rank $k$. That is, a correlation matrix $C$ can be written as

$$C = \mathrm{diag}(I - XX^T) + XX^T$$

where $X$ is an $n \times k$ matrix, often referred to as the factor loading matrix, and $k$ is generally much smaller than $n$. These correlation matrices arise in factor models of asset returns, collateralized debit obligations and multivariate time series.

The routine g02ae computes the nearest factor loading matrix, $X$, that gives the nearest correlation matrix for an approximate one, $G$, by finding the minimum of

$$\|G - XX^T + \mathrm{diag}(XX^T - I)\|_F$$

We have implemented the spectral projected gradient method of Birgin, Martinez and Raydan [1] as suggested by Borsdorf, Higham and Raydan [3].

## Table of Functionality

This table lists all our nearest correlation matrix routines and indicates which use Frobenius norm as the measure of nearness and which use a shrinking algorithm. We also show what weighting and fixing can be used in each and whether a minimum eigenvalue can be requested.

| Routine | Nearness measured in the Frobenius Norm | Shrinking Algorithm | Nearest Matrix with Factor Structure | Elements can be weighted | Elements can be fixed | Minimum eigenvalue can be requested |
|---|---|---|---|---|---|---|
| g02aa | ✓ | | | | | |
| g02ab | ✓ | | | ✓ | | ✓ |
| g02ae | ✓ | | ✓ | | | |
| g02aj | ✓ | | | ✓ | | ✓ |
| g02an | | ✓ | | | ✓ | |
| g02ap | | ✓ | | ✓ | ✓ | ✓ |

## References


[1] Birgin E G, Martínez J M and Raydan M (2001) Algorithm 813: SPG–software for convex-constrained optimization *ACM Trans. Math Software* **27** 340–349

[2] Borsdorf R and Higham N J (2010) A preconditioned (Newton) algorithm for the nearest correlation matrix *IMA Journal of Numerical Analysis* **30(1)** 94–107

[3] Borsdorf R, Higham N J and Raydan M (2010) Computing a nearest correlation matrix with factor structure. *SIAM J. Matrix Anal. Appl.* **31(5)** 2603–2622

[4] Higham N J, Strabić N and Šego V (2016) Restoring definiteness via shrinking, with an application to correlation matrices with a fixed block. *SIAM Review* **58(2)** 245—263

[5] Jiang K, Sun D and Toh K-C (2012) An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP *SIAM J. Optim.* **22(3)** 1042–1064

[6] Qi H and Sun D (2006) A quadratically convergent Newton method for computing the nearest correlation matrix *SIAM J. Matrix Anal. Appl.* **29(2)** 360–385