Nearest Correlation Matrix

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

Introduction

A correlation matrix is characterized as being a real, square matrix that

- is symmetric;
- has ones on the diagonal;
- has non-negative eigenvalues.

A matrix with non-negative eigenvalues is called positive semidefinite. If a matrix C is a correlation matrix then the elements of C, 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 (See References) 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. For example, the correlations may be between stocks measured over a period of time and some data may be missing. If individual correlations are computed using observation data they have in common, and this varies over all the variables, it will give rise to an indefinite matrix. 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 result is a matrix which is an *approximate correlation matrix* and this must be fixed for subsequent analysis that relies upon having a *true correlation matrix* to be valid. Ideally, we wish to find the 'nearest' true correlation matrix to our approximate one for some sensible definition of 'near'. This is our *basic nearest correlation matrix problem.*

The Basic Nearest Correlation Matrix Problem

The NAG Library routine g02aa implements a Newton algorithm to solve the *basic problem* we outlined in the introduction. 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 [8], has superior convergence properties over 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. It is this enhanced algorithm that has been incorporated into our Library.

Weighted Problems and Forcing a Positive Definite Correlation Matrix

In NAG Library routine g02ab we extend the functionality provided by g02aa. If we have an approximate correlation matrix it is reasonable to suppose that not all of the matrix is actually approximate, perhaps only part of it is. 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 algorithm 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 $\sqrt{w_{ij}}\ (g_{ij}\text{-}x_{ij})\ \sqrt{w_{jj}}$. 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 $c_{i,i} = a_{i,j}\ b_{i,j}$. Thus by choosing appropriate values in *H*, we can emphasize individual elements in *G* and leave the others unweighted. The algorithm employed here is by Jiang, Sun and Toh [7], and has the Newton algorithm at its core.

Both g02ab and g02aj allows us to specify that the computed correlation matrix is positive definite, that is its eigenvalues are greater than zero. This is required in some applications to improve the condition of the matrix and to increase stability.

Constraining the Rank of the Correlation Matrix

If a low-rank correlation matrix is required, for example to constrain the number of independent random variables, g02ak can be used. It finds the nearest correlation matrix, in the Frobenius norm, of maximum prescribed rank. The routine is based on the Majorized Penalty Approach proposed by Gao and Sun [4].

Fixing Correlations with Shrinking and Alternating Projections

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 four 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 [6], 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 α in the interval [0,1] that gives a positive semidefinite result. The smaller α is, the closer we stay to our original matrix, and any α 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 usto supply our own *targe*t matrix. The target matrix, *T*, is defined by specifying a matrix of weights, *H*, with $T = H \circ G$. We then find a solution of the form

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

computing α 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 α 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.

The shrinking algorithms are characterized by their speed and the potentially large distance between the input and the output. Alternating projections with Anderson acceleration is another algorithm we employ to compute fixed block problems, and it's speed and nearness characteristics are the reverse of that of shrinking.

The input matrix is repeatedly, and alternately, projected on to the nearest matrix in the sets of semidefinite matrices and matrices with entries we wish to preserve, including the unit diagonal. Whilst there is no guarantee of convergence theoretically, in practice the algorithm will find the nearest correlation matrix in the intersection of these two sets. In the routine g02as we employ the method of Higham and Strabić [5], which computes the nearest correlation matrix in the Frobenius norm while fixing arbitrary elements, and optionally setting a minimum eigenvalue.

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 algorithms (g02aa, g02ab, g02aj and g02ak) and the alternating projection algorithm (g02as) will always find the nearest solution to the problem they are solving, recalling that weighted algorithms are only influencing the input. The shrinking routines (g02an and g02ap) will find a result further away but 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 algorithms 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. The closest solution will be found with g02as. Whilst much slower than the Newton algorithms it can outperform g02aj for speed.

For fixing two diagonal blocks, or for arbitrary fixing and weighting, the choice is between g02aj, g02ap and g02as with the same speed and nearness trade-off. The alternating projections of g02as will fix elements and find the nearest solution. Although the shrinking algorithm is fixing elements and is quick, the target matrix is required to be positive definite and form part of a valid correlation matrix, which can be a limitation. Since g02aj only weights elements in the input it may offer some flexibility here, if the elements you are interested in preserving are only close to ones in a valid correlation matrix.

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

To constrain the rank of the output correlation matrix use g02ak.

The tolerance used in all of the algorithms, which defines convergence, can obviously affect the number of iterations undertaken and the thus the speed and 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, the correlation matrix *C* can be written as

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

where X here is an $n\ \times k$ matrix, often referred to as the factor loading matrix, and *k* is generally much small 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 of all our nearest correlation matrix routines and indicates the measure of nearness and what weighting and fixing can be used in each.

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**	**Maximum rank can be requested**
g02aa	X						
g02ab	X			X		X	
g02ae	X		X				
g02aj	X			X		X	
g02aj	X						X
g02an		X			X		
g02ap		X		X	X	X	
g02as	X				X	X	

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] Gao Y and Sun D (2010) A majorized penalty approach for calibrating rank constrained correlation matrix problems *Technical report* Department of Mathematics, National University of Singapore

[5] Higham N J and Strabić N (2016) Anderson acceleration of the alternating projections method for computing the nearest correlation matrix *Numer. Algor. 72 1021–1042*

[6] Higham N J, Strabić N and Šego V (2014) Restoring definiteness via shrinking, with an application to correlation matrices with a fixed block *MIMS EPrint 2014.54* Manchester Institute for Mathematical Sciences, The University of Manchester, UK

[7] 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

[8] 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