Nearest Correlation Algorithms in the NAG Library

Craig Lucas and Mike Croucher - Numerical Algorithms Group

1. Introduction

A correlation matrix C, has elements c_{ij} representing the pair-wise correlation of entity i with entity j, that is, the strength and direction of a linear relationship between the two. The matrix:

(a) is real, square and symmetric
(b) has unit diagonal and $|c_{ij}| \leq 1$
(c) is positive semidefinite, its eigenvalues are positive or zero

An *approximate* correlation matrix is one that is not positive semidefinite. Consider an application from finance:

- correlations between stocks are used to construct portfolios
- some data may be missing, leading to constructed matrices *n*ot being positive semidefinite
- a mathematically *true* correlation matrix is required for analysis, and we seek one that is, in some sense, *near* to the original matrix

This poster discusses the algorithms, and their properties, that compute nearest correlation matrices in the NAG Library.

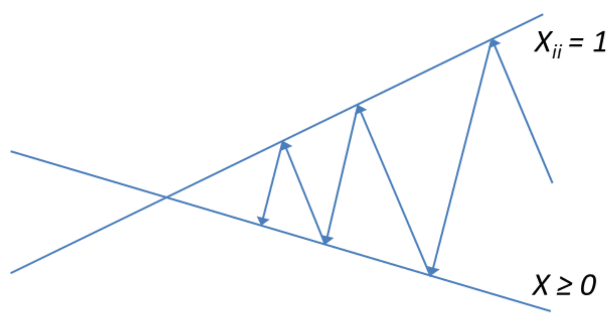
2. The Basic Problem and the Algorithms

Many of our algorithms find a true correlation matrix X that is closest to the approximate input matrix, G, in the Frobenius norm, which may be weighted. That is, we find the minimum of:

$$\|G - X\|_F.$$

A **Newton Method** described by Qi and Sun [4] and improved at the University of Manchester by Borsdorf and Higham [1] forms the basis of four NAG routines: corrmat_nearest, solving the basic problem, and corrmat_nearest_bounded, corrmat_h_weight & corrmat_nearest_rank which offer additional functionality.

Alternating Projections with Anderson Acceleration [2] is used in corrmat_fixed. The input is repeatedly projected on to the sets of semidefinite and unit diagonal matrices, alternatively. This algorithm is slower to converge than Newton, but allows the *fixing* of elements.



The **Shrinking Method** of Higham, Strabić and Šego [3] finds a true correlation matrix, using a matrix of weights, H, of the form:

$$\alpha T + (1 - \alpha)G, \qquad T = H \circ G.$$

The smallest $\alpha \in [0, 1]$ that gives a positive semidefinite result is found. This is used in corrmat_shrinking and corrmat_target.

A choice of algorithms
Trade off between speed and nearness
Fix or influence matrix entries
Specify eigenvalues or rank

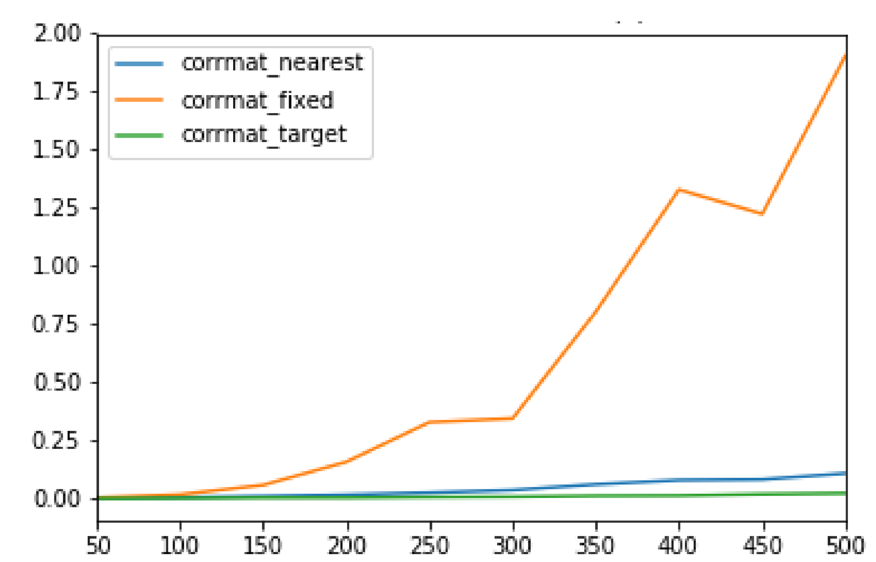
3. Basic Results

First we look at the three core algorithms solving the basic problem. Below we show the norm of the original matrix compared to the computed true correlation matrix and also the computation time.

n	_nearest	_fixed	_shrinking
100	0.058	0.058	1.265
200	0.198	0.198	4.974
300	0.3	0.3	8.392
400	0.462	0.462	13.615
500	0.603	0.603	18.901

$\|G - X\|_F$. for the basic problem for different n.

The shrinking algorithm gives a matrix far from the original, although it is the quickest. Alternating projections is the slowest by far, but it is not intended for this problem.



Times (secs) for the basic problem for different n.

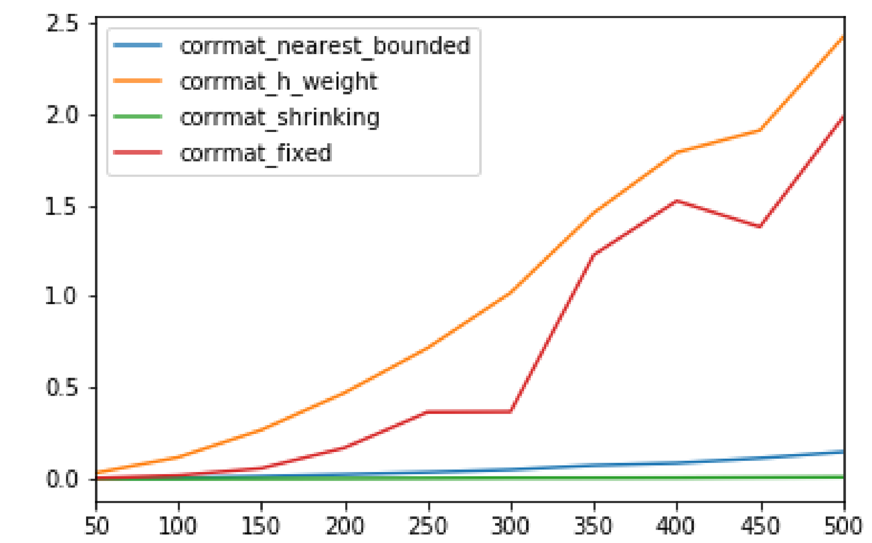
4. Forcing Positive Definiteness and Maximum Rank

Several routines in the NAG Library allow you specify a minimum eigenvalue or a maximum rank. The affect of this is, inevitably, to increase computation time and the distance from the orignal matrix.

5. Fixing a Block of Elements

Some correlations in an input matrix may be true. This leads to the need to *fix* or *weight* a block of entries. The shrinking algorithm handles this with the target matrix and alternating projections can be adapted easily. The Newton algorithm is extended for the following norms in corrmat_nearest_bounded and corrmat_h_weight respectively.

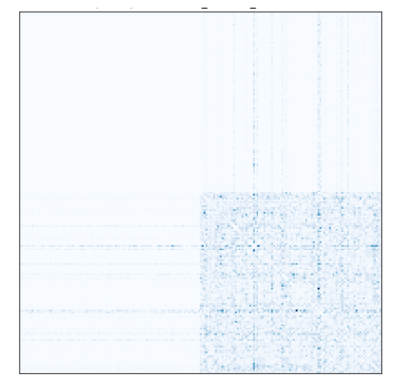
$$\min \|W^{1/2}(G - X)W^{1/2}\|_F \text{ and } \min \|H \circ (G - X)\|_F$$



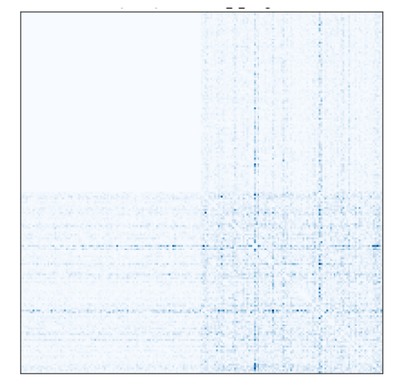
Times (secs) for fixed block problem for different n.

The routine corrmat_h_weight is the slowest and fails to fix the elements. However, corrmat_fixed does fix the elements as required. The other routines are much quicker, with corrmat_shrinking fixing the block, but the affect on the remaining elements may not be desired.

The plots on the right show the difference between the input and computed matrices for $n = 200$. The darker the blue, the greater the difference. We can see the effect of the weights on the off-block elements, which is severe for the shrinking algorithm.



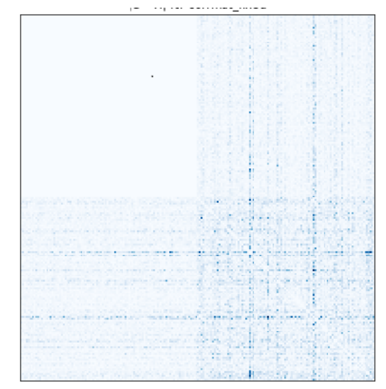
corrmat_nearest_bounded



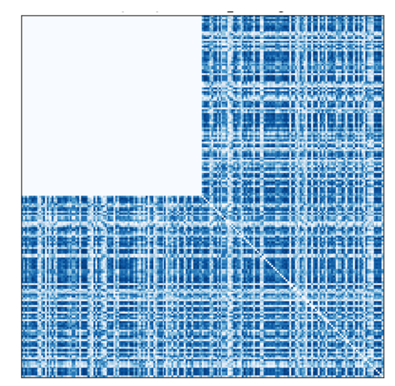
corrmat_h_weight

corrmat_nearest_bounded	0.274
corrmat_h_weight	0.203
corrmat_fixed	0.200
corrmat_shrinking	4.750

$\|G-X\|_F$. for the fixed block problem for different n.



corrmat_fixed



corrmat_shrinking

6. Summary of Routines

	Nearness in Norm	Shrinking Algorithm	Weights	Fixing	Min E'value	Max Rank
corrmat_nearest	✓					
corrmat_nearest_bounded	✓		✓		✓	
corrmat_h_weight	✓		✓		✓	
corrmat_nearest_rank	✓					✓
corrmat_nearest_shrinking		✓		✓		
corrmat_nearest_target		✓		✓	✓	
corrmat_nearest_fixed	✓			✓	✓	

References

To find out more visit the NAG website and see the following papers:

[1] R. Borsdorf and N. J. Higham. A preconditioned (Newton) algorithm for the nearest correlation matrix. IMA J. of Numer. Anal., 30(1):94-107, 2010.
[2] N. J. Higham N J and N. Strabić. Anderson acceleration of the alternating projections method for computing the nearest correlation matrix Numer. Algor. 72 1021–1042, 2016
[3] N. J. Higham, N. Strabić and V. Šego. 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, 2014.
[4] H. Qi and D. Sun D. A quadratically convergent Newton method for computing the nearest correlation matrix. SIAM J. Matrix Anal. Appl., 29(2):360-385, 2006.

 craig.lucas@nag.co.uk