NAG Toolbox

nag_nearest_correlation_h_weight (g02aj)

1 Purpose

nag_nearest_correlation_h_weight (g02aj) computes the nearest correlation matrix, using element-wise weighting in the Frobenius norm and optionally with bounds on the eigenvalues, to a given square, input matrix.

2 Syntax

```
[g, h, x, iter, norm_p, ifail] = nag_nearest_correlation_h_weight(g, alpha, h, 'n', n, 'errtol', errtol, 'maxit', maxit)
[g, h, x, iter, norm_p, ifail] = g02aj(g, alpha, h, 'n', n, 'errtol', errtol, 'maxit', maxit)
```

3 Description

nag_nearest_correlation_h_weight (g02aj) finds the nearest correlation matrix, X, to an approximate correlation matrix, G, using element-wise weighting, this minimizes $\|H \circ (G - X)\|_F$, where $C = A \circ B$ denotes the matrix C with elements $C_{ij} = A_{ij} \times B_{ij}$.

You can optionally specify a lower bound on the eigenvalues, α, of the computed correlation matrix, forcing the matrix to be strictly positive definite, if $0 < \alpha < 1$.

Zero elements in H should be used when you wish to put no emphasis on the corresponding element of G. The algorithm scales H so that the maximum element is 1. It is this scaled matrix that is used in computing the norm above and for the stopping criteria described in Section 7.

Note that if the elements in H vary by several orders of magnitude from one another the algorithm may fail to converge.

4 References

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

Jiang K, Sun D and Toh K-C (To appear) An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP

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

5 Parameters

5.1 Compulsory Input Parameters

1: **g**(ldg, **n**) – REAL (KIND=nag_wp) array

ldg, the first dimension of the array, must satisfy the constraint $ldg \geq$ **n**.

G, the initial matrix.

2: **alpha** – REAL (KIND=nag_wp)

The value of α.

If **alpha** < 0.0, 0.0 is used.

Constraint: **alpha** < 1.0.

3: **h**$(ldh, \mathbf{n})$ – REAL (KIND=nag_wp) array

ldh, the first dimension of the array, must satisfy the constraint $ldh \geq \mathbf{n}$.

The matrix of weights H.

Constraint: $h(i,j) \geq 0.0$, for all i and $j = 1, 2, \ldots, n$, $i \neq j$.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the arrays **h**, **g** and the second dimension of the arrays **h**, **g**. (An error is raised if these dimensions are not equal.)

The order of the matrix G.

Constraint: **n** > 0.

2: **errtol** – REAL (KIND=nag_wp)

Default: 0.0

The termination tolerance for the iteration. If **errtol** ≤ 0.0 then **n** $\times \sqrt{\textbf{\textit{machine precision}}}$ is used. See Section 7 for further details.

3: **maxit** – INTEGER

Default: 0

Specifies the maximum number of iterations to be used.

If **maxit** ≤ 0, 200 is used.

5.3 Output Parameters

1: **g**$(ldg, \mathbf{n})$ – REAL (KIND=nag_wp) array

2: **h**$(ldh, \mathbf{n})$ – REAL (KIND=nag_wp) array

A symmetric matrix $\frac{1}{2}(H + H^{\mathrm{T}})$ with its diagonal elements set to zero and the remaining elements scaled so that the maximum element is 1.0.

3: **x**$(ldx, \mathbf{n})$ – REAL (KIND=nag_wp) array

Contains the nearest correlation matrix.

4: **iter** – INTEGER

The number of iterations taken.

5: **norm_p** – REAL (KIND=nag_wp)

The value of $\|H \circ (G - X)\|_F$ after the final iteration.

6: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

Constraint: **n** > 0.

ifail $= 2$

Constraint: $ldg \geq$ **n**.

ifail $= 3$

Constraint: $ldh \geq$ **n**.

ifail $= 4$

Constraint: $ldx \geq$ **n**.

ifail $= 5$

Constraint: **alpha** < 1.0.

ifail $= 6$

On entry, one or more of the off-diagonal elements of H were negative.

ifail $= 7$

Function fails to converge in ⟨*value*⟩ iterations.
Increase **maxit** or check the call to the function.

ifail $= 8$ (*warning*)

Failure to solve intermediate eigenproblem. This should not occur. Please contact NAG with details of your call.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

The returned accuracy is controlled by **errtol** and limited by ***machine precision***. If e_i is the value of **norm_p** at the ith iteration, that is

$$e_i = \|H \circ (G - X)\|_F,$$

where H has been scaled as described above, then the algorithm terminates when:

$$\frac{|e_i - e_{i-1}|}{1 + \max(e_i, e_{i-1})} \leq \textbf{errtol}.$$

8 Further Comments

Arrays are internally allocated by nag_nearest_correlation_h_weight (g02aj). The total size of these arrays is $15 \times \mathbf{n} + 5 \times \mathbf{n} \times \mathbf{n} + \max(2 \times \mathbf{n} \times \mathbf{n} + 6 \times \mathbf{n} + 1, 120 + 9 \times \mathbf{n})$ double elements and $5 \times \mathbf{n} + 3$ integer elements. All allocated memory is freed before return of nag_nearest_correlation_h_weight (g02aj).

9 Example

This example finds the nearest correlation matrix to:

$$G = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

weighted by:

$$H = \begin{pmatrix} 0.0 & 10.0 & 0.0 & 0.0 \\ 10.0 & 0.0 & 1.5 & 1.5 \\ 0.0 & 1.5 & 0.0 & 0.0 \\ 0.0 & 1.5 & 0.0 & 0.0 \end{pmatrix}$$

with minimum eigenvalue 0.04.

9.1 Program Text

```
function g02aj_example

fprintf('g02aj example results\n\n');

g = [ 2, -1,  0,  0;
     -1,  2, -1,  0;
      0, -1,  2, -1;
      0,  0, -1,  2];
h = [ 0.0, 10.0,  0.0,  0.0;
     10.0,  0.0,  1.5,  1.5;
      0.0,  1.5,  0.0,  0.0;
      0.0,  1.5,  0.0,  0.0];
alpha = 0.04;
[g, h, x, iter, norm_p, ifail] = ...
  g02aj(g, alpha, h);

fprintf('\nReturned H Matrix\n');
disp(h);
fprintf('Nearest Correlation Matrix\n');
disp(x);
fprintf('Number of iterations taken:     %d\n', iter);
fprintf('Norm value: %26.4f\n', norm_p);
fprintf('Alpha:      %26.4f\n', alpha);

[~, w, info] = f08fa('n', 'u', x);

fprintf('\nEigenvalues of X\n');
disp(w');
```

9.2 Program Results

```
     g02aj example results

Returned H Matrix
         0    1.0000         0         0
    1.0000         0    0.1500    0.1500
         0    0.1500         0         0
         0    0.1500         0         0

Nearest Correlation Matrix
```

```
    1.0000   -0.9229    0.7734    0.0026
   -0.9229    1.0000   -0.7843   -0.0000
    0.7734   -0.7843    1.0000   -0.0615
    0.0026   -0.0000   -0.0615    1.0000

Number of iterations taken:     66
Norm value:                     0.1183
Alpha:                          0.0400

Eigenvalues of X
    0.0769    0.2637    1.0031    2.6563
```