NAG Toolbox

nag_correg_corrmat_nearest_kfactor (g02ae)

1 Purpose

nag_correg_corrmat_nearest_kfactor (g02ae) computes the factor loading matrix associated with the nearest correlation matrix with k-factor structure, in the Frobenius norm, to a given square, input matrix.

2 Syntax

```
[g, x, iter, feval, nrmpgd, ifail] = nag_correg_corrmat_nearest_kfactor(g, k,
'n', n, 'errtol', errtol, 'maxit', maxit)
```

```
[g, x, iter, feval, nrmpgd, ifail] = g02ae(g, k, 'n', n, 'errtol', errtol,
'maxit', maxit)
```

3 Description

A correlation matrix C with k-factor structure may be characterised as a real square matrix that is symmetric, has a unit diagonal, is positive semidefinite and can be written as $C = XX^{\mathrm{T}} + \mathrm{diag}(I - XX^{\mathrm{T}})$, where I is the identity matrix and X has n rows and k columns. X is often referred to as the factor loading matrix.

nag_correg_corrmat_nearest_kfactor (g02ae) applies a spectral projected gradient method to the modified problem $\min \|G - XX^{\mathrm{T}} + \mathrm{diag}(XX^{\mathrm{T}} - I)\|_F$ such that $\left\|x_i^{\mathrm{T}}\right\|_2 \leq 1$, for $i = 1, 2, \ldots, n$, where x_i is the ith row of the factor loading matrix, X, which gives us the solution.

4 References

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

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

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: **k** – INTEGER

k, the number of factors and columns of X.

Constraint: $0 <$ **k** $\leq$ **n**.

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the order of the matrix G.

Constraint: $\mathbf{n} > 0$.

2: **errtol** – REAL (KIND=nag_wp)

Default: 0.0

The termination tolerance for the projected gradient norm. See references for further details. If **errtol** ≤ 0.0 then 0.01 is used. This is often a suitable default value.

3: **maxit** – INTEGER

Default: 0

Specifies the maximum number of iterations in the spectral projected gradient method.

If **maxit** ≤ 0, 40000 is used.

5.3 Output Parameters

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

A symmetric matrix $\frac{1}{2}(G + G^{\mathrm{T}})$ with the diagonal elements set to unity.

2: $\mathbf{x}(ldx, \mathbf{k})$ – REAL (KIND=nag_wp) array

Contains the matrix X.

3: **iter** – INTEGER

The number of steps taken in the spectral projected gradient method.

4: **feval** – INTEGER

The number of evaluations of $\|G - XX^{\mathrm{T}} + \mathrm{diag}(XX^{\mathrm{T}} - I)\|_F$.

5: **nrmpgd** – REAL (KIND=nag_wp)

The norm of the projected gradient at 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: $0 < \mathbf{k} \leq \mathbf{n}$.

Constraint: $ldg \geq \mathbf{n}$.

Constraint: $ldx \geq \mathbf{n}$.

Constraint: $\mathbf{n} > 0$.

ifail $= 2$

Spectral gradient method fails to converge in $\langle value \rangle$ iterations.

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***.

8 Further Comments

Arrays are internally allocated by nag_correg_corrmat_nearest_kfactor (g02ae). The total size of these arrays is $\mathbf{n} \times \mathbf{n} + 4 \times \mathbf{n} \times \mathbf{k} + (nb + 3) \times \mathbf{n} + \mathbf{n} + 50$ double elements and $6 \times \mathbf{n}$ integer elements. Here nb is the block size required for optimal performance by nag_lapack_dsytrd (f08fe) and nag_lapack_dormtr (f08fg) which are called internally. All allocated memory is freed before return of nag_correg_corrmat_nearest_kfactor (g02ae).

See nag_mv_factor (g03ca) for constructing the factor loading matrix from a known correlation matrix.

9 Example

This example finds the nearest correlation matrix with $k = 2$ factor structure to:

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

9.1 Program Text

```
     function g02ae_example

fprintf('g02ae example results\n\n');

g = [2, -1,  0,  0;
    -1,  2, -1,  0;
     0, -1,  2, -1;
     0,  0, -1,  2];
k = nag_int(2);

% Calculate nearest correlation matrix
[g, x, iter, feval, nrmpgd, ifail] = ...
  g02ae(g, k);

fprintf('\n Factor Loading Matrix x:\n');
disp(x);
fprintf('\n Number of Newton steps taken:   %d\n', iter);
fprintf(' Number of function evaluations: %d\n', feval);

% Generate Nearest k factor correlation matrix

fprintf('\n Nearest Correlation Matrix:\n');
disp(x*transpose(x) + diag(diag(eye(4)-x*transpose(x))));
```

9.2 Program Results

```
     g02ae example results

 Factor Loading Matrix x:
    0.7665   -0.6271
   -0.4250    0.9052
   -0.4250   -0.9052
```

```
   0.7665    0.6271

Number of Newton steps taken:   5
Number of function evaluations: 6

Nearest Correlation Matrix:
   1.0000   -0.8935    0.2419    0.1943
  -0.8935    1.0000   -0.6388    0.2419
   0.2419   -0.6388    1.0000   -0.8935
   0.1943    0.2419   -0.8935    1.0000
```