NAG Library Routine Document

G03FAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G03FAF performs a principal coordinate analysis also known as classical metric scaling.

2 Specification

```
SUBROUTINE G03FAF (ROOTS, N, D, NDIM, X, LDX, EVAL, WK, IWK, IFAIL)

INTEGER             N, NDIM, LDX, IWK(5*N), IFAIL
REAL (KIND=nag_wp) D(N*(N-1)/2), X(LDX,NDIM), EVAL(N), WK(N*(N+17)/2-1)
CHARACTER(1)        ROOTS
```

3 Description

For a set of n objects a distance matrix D can be calculated such that d_{ij} is a measure of how 'far apart' are objects i and j (see G03EAF for example). Principal coordinate analysis or metric scaling starts with a distance matrix and finds points X in Euclidean space such that those points have the same distance matrix. The aim is to find a small number of dimensions, $k \ll (n-1)$, that provide an adequate representation of the distances.

The principal coordinates of the points are computed from the eigenvectors of the matrix E where $e_{ij} = -1/2\left(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2\right)$ with $d_{i.}^2$ denoting the average of d_{ij}^2 over the suffix j, etc.. The eigenvectors are then scaled by multiplying by the square root of the value of the corresponding eigenvalue.

Provided that the ordered eigenvalues, λ_i, of the matrix E are all positive, $\sum_{i=1}^{k}\lambda_i / \sum_{i=1}^{n-1}\lambda_i$ shows how well the data is represented in k dimensions. The eigenvalues will be non-negative if E is positive semidefinite. This will be true provided d_{ij} satisfies the inequality: $d_{ij} \leq d_{ik} + d_{jk}$ for all i, j, k. If this is not the case the size of the negative eigenvalue reflects the amount of deviation from this condition and the results should be treated cautiously in the presence of large negative eigenvalues. See Krzanowski (1990) for further discussion. G03FAF provides the option for all eigenvalues to be computed so that the smallest eigenvalues can be checked.

4 References

Chatfield C and Collins A J (1980) *Introduction to Multivariate Analysis* Chapman and Hall

Gower J C (1966) Some distance properties of latent root and vector methods used in multivariate analysis *Biometrika* **53** 325–338

Krzanowski W J (1990) *Principles of Multivariate Analysis* Oxford University Press

5 Parameters

1: ROOTS – CHARACTER(1) *Input*

On entry: indicates if all the eigenvalues are to be computed or just the NDIM largest.

ROOTS = 'A'
All the eigenvalues are computed.

ROOTS = 'L'
Only the largest NDIM eigenvalues are computed.

Constraint: ROOTS = 'A' or 'L'.

2: N – INTEGER *Input*

On entry: n, the number of objects in the distance matrix.

Constraint: N $>$ NDIM.

3: D(N $\times$ (N $-$ 1)/2) – REAL (KIND=nag_wp) array *Input*

On entry: the lower triangle of the distance matrix D stored packed by rows. That is D$((i-1)\times(i-2)/2+j)$ must contain d_{ij} for $i = 2, 3, \ldots, n; j = 1, 2, \ldots, i-1$.

Constraint: D$(i) \geq 0.0$, for $i = 1, 2, \ldots, n(n-1)/2$.

4: NDIM – INTEGER *Input*

On entry: k, the number of dimensions used to represent the data.

Constraint: NDIM ≥ 1.

5: X(LDX,NDIM) – REAL (KIND=nag_wp) array *Output*

On exit: the ith row contains k coordinates for the ith point, $i = 1, 2, \ldots, n$.

6: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G03FAF is called.

Constraint: LDX $\geq$ N.

7: EVAL(N) – REAL (KIND=nag_wp) array *Output*

On exit: if ROOTS = 'A', EVAL contains the n scaled eigenvalues of the matrix E.

If ROOTS = 'L', EVAL contains the largest k scaled eigenvalues of the matrix E.

In both cases the eigenvalues are divided by the sum of the eigenvalues (that is, the trace of E).

8: WK(N $\times$ (N + 17)/2 $-$ 1) – REAL (KIND=nag_wp) array *Workspace*

9: IWK(5 $\times$ N) – INTEGER array *Workspace*

10: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, NDIM < 1,
or N $<$ NDIM,
or ROOTS $\neq$ 'A' or 'L',
or LDX $<$ N.

IFAIL $= 2$

On entry, D$(i) < 0.0$ for some i, $i = 1, 2, \ldots, n(n-1)/2$,
or all elements of D $= 0.0$.

IFAIL $= 3$

There are less than NDIM eigenvalues greater than zero. Try a smaller number of dimensions (NDIM) or use non-metric scaling (G03FCF).

IFAIL $= 4$

The computation of the eigenvalues or eigenvectors has failed. Seek expert help.

7 Accuracy

G03FAF uses F08JFF (DSTERF) or F08JJF (DSTEBZ) to compute the eigenvalues and F08JKF (DSTEIN) to compute the eigenvectors. These routines should be consulted for a discussion of the accuracy of the computations involved.

8 Further Comments

Alternative, non-metric, methods of scaling are provided by G03FCF.

The relationship between principal coordinates and principal components, see G03FCF, is discussed in Krzanowski (1990) and Gower (1966).

9 Example

The data, given by Krzanowski (1990), are dissimilarities between water vole populations in Europe. The first two principal coordinates are computed.

9.1 Program Text

```
    Program g03fafe

!     G03FAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g03faf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                           :: ifail, ld, ldx, liwk, lwk, n, ndim
      Character (1)                     :: roots
!     .. Local Arrays ..
```

```
      Real (Kind=nag_wp), Allocatable  :: d(:), eval(:), wk(:), x(:,:)
      Integer, Allocatable             :: iwk(:)
!     .. Executable Statements ..
      Write (nout,*) 'G03FAF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Read in the problem size
      Read (nin,*) n, ndim, roots

      ld = n*(n-1)/2
      ldx = n
      lwk = n*(n+17)/2 - 1
      liwk = 5*n
      Allocate (d(ld),x(ldx,ndim),eval(n),wk(lwk),iwk(liwk))

!     Read in the lower triangular part of the distance matrix
      Read (nin,*) d(1:ld)

!     Perform principal co-ordinate analysis
      ifail = 0
      Call g03faf(roots,n,d,ndim,x,ldx,eval,wk,iwk,ifail)

!     Display results
      Write (nout,*) ' Scaled Eigenvalues'
      Write (nout,*)
      If (roots=='L' .Or. roots=='l') Then
        Write (nout,99999) eval(1:ndim)
      Else
        Write (nout,99999) eval(1:n)
      End If
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04caf('General',' ',n,ndim,x,ldx,'Co-ordinates',ifail)

99999 Format (8F10.4)
    End Program g03fafe
```

9.2 Program Data

```
G03FAF Example Program Data
14 2 'l'
0.099
0.033 0.022
0.183 0.114 0.042
0.148 0.224 0.059 0.068
0.198 0.039 0.053 0.085 0.051
0.462 0.266 0.322 0.435 0.268 0.025
0.628 0.442 0.444 0.406 0.240 0.129 0.014
0.113 0.070 0.046 0.047 0.034 0.002 0.106 0.129
0.173 0.119 0.162 0.331 0.177 0.039 0.089 0.237 0.071
0.434 0.419 0.339 0.505 0.469 0.390 0.315 0.349 0.151 0.430
0.762 0.633 0.781 0.700 0.758 0.625 0.469 0.618 0.440 0.538 0.607
0.530 0.389 0.482 0.579 0.597 0.498 0.374 0.562 0.247 0.383 0.387 0.084
0.586 0.435 0.550 0.530 0.552 0.509 0.369 0.471 0.234 0.346 0.456 0.090 0.038
```

9.3 Program Results

```
 G03FAF Example Program Results

  Scaled Eigenvalues

    0.7871    0.2808

 Co-ordinates
             1       2
  1   0.2408  0.2337
```

```
  2   0.1137  0.1168
  3   0.2394  0.0760
  4   0.2129  0.0605
  5   0.2495 -0.0693
  6   0.1487 -0.0778
  7  -0.0514 -0.1623
  8   0.0115 -0.3446
  9  -0.0039  0.0059
 10   0.0386 -0.0089
 11  -0.0421 -0.0566
 12  -0.5158  0.0291
 13  -0.3180  0.1501
 14  -0.3238  0.0475
```