NAG Toolbox

nag_mv_distance_mat (g03ea)

1 Purpose

nag_mv_distance_mat (g03ea) computes a distance (dissimilarity) matrix.

2 Syntax

```
[s, d, ifail] = nag_mv_distance_mat(update, dist, scal, x, isx, s, d, 'n', n, 'm', m)
[s, d, ifail] = g03ea(update, dist, scal, x, isx, s, d, 'n', n, 'm', m)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 22: **n** was made optional.

3 Description

Given n objects, a distance or dissimilarity matrix is a symmetric matrix with zero diagonal elements such that the ijth element represents how far apart or how dissimilar the ith and jth objects are.

Let X be an n by p data matrix of observations of p variables on n objects, then the distance between object j and object k, d_{jk}, can be defined as:

$$d_{jk} = \left\{ \sum_{i=1}^{p} D\left(x_{ji}/s_i, x_{ki}/s_i\right) \right\}^{\alpha},$$

where x_{ji} and x_{ki} are the jith and kith elements of X, s_i is a standardization for the ith variable and $D(u,v)$ is a suitable function. Three functions are provided in nag_mv_distance_mat (g03ea).

(a) Euclidean distance: $D(u,v) = (u-v)^2$ and $\alpha = \frac{1}{2}$.

(b) Euclidean squared distance: $D(u,v) = (u-v)^2$ and $\alpha = 1$.

(c) Absolute distance (city block metric): $D(u,v) = |u-v|$ and $\alpha = 1$.

Three standardizations are available.

(a) Standard deviation: $s_i = \sqrt{\sum_{j=1}^{n} \left(x_{ji} - \bar{x}\right)^2 / (n-1)}$

(b) Range: $s_i = \max(x_{1i}, x_{2i}, \ldots, x_{ni}) - \min(x_{1i}, x_{2i}, \ldots, x_{ni})$

(c) User-supplied values of s_i.

In addition to the above distances there are a large number of other dissimilarity measures, particularly for dichotomous variables (see Krzanowski (1990) and Everitt (1974)). For the dichotomous case these measures are simple to compute and can, if suitable scaling is used, be combined with the distances computed by nag_mv_distance_mat (g03ea) using the updating option.

Dissimilarity measures for variables can be based on the correlation coefficient for continuous variables and contingency table statistics for dichotomous data, see chapters G02 and G11 respectively.

nag_mv_distance_mat (g03ea) returns the strictly lower triangle of the distance matrix.

4 References

Everitt B S (1974) *Cluster Analysis* Heinemann

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

5 Parameters

5.1 Compulsory Input Parameters

1: **update** – CHARACTER(1)

Indicates whether or not an existing matrix is to be updated.

update = 'U'
The matrix D is updated and distances are added to D.

update = 'I'
The matrix D is initialized to zero before the distances are added to D.

Constraint: **update** = 'U' or 'I'.

2: **dist** – CHARACTER(1)

Indicates which type of distances are computed.

dist = 'A'
Absolute distances.

dist = 'E'
Euclidean distances.

dist = 'S'
Euclidean squared distances.

Constraint: **dist** = 'A', 'E' or 'S'.

3: **scal** – CHARACTER(1)

Indicates the standardization of the variables to be used.

scal = 'S'
Standard deviation.

scal = 'R'
Range.

scal = 'G'
Standardizations given in array **s**.

scal = 'U'
Unscaled.

Constraint: **scal** = 'S', 'R', 'G' or 'U'.

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

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

$\mathbf{x}(i,j)$ must contain the value of the jth variable for the ith object, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, \mathbf{m}$.

5: $\mathbf{isx}(\mathbf{m})$ – INTEGER array

$\mathbf{isx}(j)$ indicates whether or not the jth variable in **x** is to be included in the distance computations.

If $\mathbf{isx}(j) > 0$ the jth variable is included, for $j = 1, 2, \ldots, \mathbf{m}$; otherwise it is not referenced.

Constraint: $\mathbf{isx}(j) > 0$ for at least one j, for $j = 1, 2, \ldots, \mathbf{m}$.

6: **s**(**m**) – REAL (KIND=nag_wp) array

If **scal** = 'G' and $\mathbf{isx}(j) > 0$ then $\mathbf{s}(j)$ must contain the scaling for variable j, for $j = 1, 2, \ldots, \mathbf{m}$.

Constraint: if **scal** = 'G' and $\mathbf{isx}(j) > 0$, $\mathbf{s}(j) > 0.0$, for $j = 1, 2, \ldots, \mathbf{m}$.

7: **d**(**n** × (**n** − **1**)/**2**) – REAL (KIND=nag_wp) array

If **update** = 'U', **d** must contain the strictly lower triangle of the distance matrix D to be updated. D must be stored packed by rows, i.e., $\mathbf{d}((i-1)(i-2)/2 + j)$, $i > j$ must contain d_{ij}.

If **update** = 'I', **d** need not be set.

Constraint: if **update** = 'U', $\mathbf{d}(j) \geq 0.0$, for $j = 1, 2, \ldots, n(n-1)/2$.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the array **x**.

n, the number of observations.

Constraint: $\mathbf{n} \geq 2$.

2: **m** – INTEGER

Default: the dimension of the arrays **isx**, **s** and the second dimension of the array **x**. (An error is raised if these dimensions are not equal.)

The total number of variables in array **x**.

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

5.3 Output Parameters

1: **s**(**m**) – REAL (KIND=nag_wp) array

If **scal** = 'S' and $\mathbf{isx}(j) > 0$ then $\mathbf{s}(j)$ contains the standard deviation of the variable in the jth column of **x**.

If **scal** = 'R' and $\mathbf{isx}(j) > 0$, $\mathbf{s}(j)$ contains the range of the variable in the jth column of **x**.

If **scal** = 'U' and $\mathbf{isx}(j) > 0$, $\mathbf{s}(j) = 1.0$.

If **scal** = 'G', **s** is unchanged.

2: **d**(**n** × (**n** − **1**)/**2**) – REAL (KIND=nag_wp) array

The strictly lower triangle of the distance matrix D stored packed by rows, i.e., d_{ij} is contained in $\mathbf{d}((i-1)(i-2)/2 + j)$, $i > j$.

3: **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 $= 1$

On entry, **n** < 2,
or $ldx <$ **n**,
or **m** ≤ 0,
or **update** $\neq$ 'I' or 'U',
or **dist** $\neq$ 'A', 'E' or 'S',
or **scal** $\neq$ 'S', 'R', 'G' or 'U'.

ifail $= 2$

On entry, **isx**$(j) \leq 0$, for $j = 1, 2, \ldots,$ **m**,
or **update** $=$ 'U' and **d**$(j) < 0.0$, for some $j = 1, 2, \ldots, n(n-1)/2$,
or **scal** $=$ 'S' or 'R' and **x**$(i, j) =$ **x**$(i+1, j)$ for $i = 1, 2, \ldots, n-1$, for some j with **isx**$(i) > 0$.
or **s**$(j) \leq 0.0$ for some j when **scal** $=$ 'G' and **isx**$(j) > 0$.

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 computations are believed to be stable.

8 Further Comments

nag_mv_cluster_hier (g03ec) can be used to perform cluster analysis on the computed distance matrix.

9 Example

A data matrix of five observations and three variables is read in and a distance matrix is calculated from variables 2 and 3 using squared Euclidean distance with no scaling. This matrix is then printed.

9.1 Program Text

```
function g03ea_example

fprintf('g03ea example results\n\n');

x = [1, 1, 1;
     2, 1, 2;
     3, 6, 3;
     4, 8, 2;
     5, 8, 0];
[n,m]  = size(x);

isx    = ones(m,1,nag_int_name);
isx(1) = nag_int(0);
s      = ones(m,1);
ld     = (n*(n-1))/2;
```

```
d      = zeros(ld,1);

% Compute the distance matrix
update = 'I';
dist = 'S';
scal = 'U';
[s, d, ifail] = g03ea( ...
        update, dist, scal, x, isx, s, d);

fprintf(' Distance Matrix\n  ');
fprintf('   %5d', [1:n-1]);
for i = 2:n
  lj = (i-1)*(i-2)/2 + 1;
  uj = i*(i-1)/2;
  fprintf('\n%2d  ', i);
  fprintf('   %5.2f', d(lj:uj));
end
fprintf('\n');
```

9.2 Program Results

```
     g03ea example results

 Distance Matrix
         1       2       3       4
 2     1.00
 3    29.00   26.00
 4    50.00   49.00    5.00
 5    50.00   53.00   13.00    4.00
```