NAG Toolbox for MATLAB

g03ea

1 Purpose

g03ea computes a distance (dissimilarity) matrix.

2 Syntax

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

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 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 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.

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 – string**

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 – string**

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 – string**

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: **n – int32 scalar**

n, the number of observations.

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

5: **x(ldx,m) – double array**

ldx, the first dimension of the array, must be at least **n**.

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

6: **isx(m) – int32 array**

isx(j) indicates whether or not the jth variable in **x** is to be included in the distance computations. If **isx**(j) > 0 the jth variable is included, for $j = 1, 2, \ldots,$ **m**; otherwise it is not referenced.

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

7: **s(m) – double array**

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

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

8: **d(n × (n − 1)/2) – double array**

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

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

5.2 Optional Input Parameters

1: **m – int32 scalar**

Default: The dimension of the arrays **x**, **isx**. (An error is raised if these dimensions are not equal.) the total number of variables in array **x**.

Constraint: **m** > 0.

5.3 Input Parameters Omitted from the MATLAB Interface

ldx

5.4 Output Parameters

1: **s(m) – double array**

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

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

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

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

2: **d(n × (n − 1)/2) – double array**

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

3: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

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

7 Accuracy

The computations are believed to be stable.

8 Further Comments

g03ea can be used to perform cluster analysis on the computed distance matrix.

9 Example

```
update = 'I';
dist = 'S';
scal = 'U';
n = int32(5);
x = [1, 1, 1;
     2, 1, 2;
     3, 6, 3;
     4, 8, 2;
     5, 8, 0];
isx = [int32(0);
     int32(1);
     int32(1)];
s = [1;
     1;
     1];
d = [5.495816452771857e+222;
     4.352225583017331e+209;
     0;
     0;
     0;
     0;
     0;
     3.619376430094708e-86;
     8.544060345986362e-72;
     8.417743908128819e+246];
[sOut, dOut, ifail] = g03ea(update, dist, scal, n, x, isx, s, d)
```

```
sOut =
     1
     1
     1
dOut =
     1
    29
    26
    50
    49
     5
    50
    53
    13
```

```
     4
ifail =
           0
```