NAG Toolbox

nag_mv_rot_procrustes (g03bc)

1 Purpose

nag_mv_rot_procrustes (g03bc) computes Procrustes rotations in which an orthogonal rotation is found so that a transformed matrix best matches a target matrix.

2 Syntax

```
[x, y, yhat, r, alpha, rss, res, ifail] = nag_mv_rot_procrustes(stand, pscale, x, y, 'n', n, 'm', m)
[x, y, yhat, r, alpha, rss, res, ifail] = g03bc(stand, pscale, x, y, '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

Let X and Y be n by m matrices. They can be considered as representing sets of n points in an m-dimensional space. The X matrix may be a matrix of loadings from say factor or canonical variate analysis, and the Y matrix may be a postulated pattern matrix or the loadings from a different sample. The problem is to relate the two sets of points without disturbing the relationships between the points in each set. This can be achieved by translating, rotating and scaling the sets of points. The Y matrix is considered as the target matrix and the X matrix is rotated to match that matrix.

First the two sets of points are translated so that their centroids are at the origin to give X_c and Y_c, i.e., the matrices will have zero column means. Then the rotation of the translated X_c matrix which minimizes the sum of squared distances between corresponding points in the two sets is found. This is computed from the singular value decomposition of the matrix:

$$X_c^{\mathrm{T}} Y_c = UDV^{\mathrm{T}},$$

where U and V are orthogonal matrices and D is a diagonal matrix. The matrix of rotations, R, is computed as:

$$R = UV^{\mathrm{T}}.$$

After rotation, a scaling or dilation factor, α, may be estimated by least squares. Thus, the final set of points that best match Y_c is given by:

$$\hat{Y_c} = \alpha X_c R.$$

Before rotation, both sets of points may be normalized to have unit sums of squares or the X matrix may be normalized to have the same sum of squares as the Y matrix. After rotation, the results may be translated to the original Y centroid.

The ith residual, r_i, is given by the distance between the point given in the ith row of Y and the point given in the ith row of $\hat{Y}$. The residual sum of squares is also computed.

4 References

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

Lawley D N and Maxwell A E (1971) *Factor Analysis as a Statistical Method* (2nd Edition) Butterworths

5 Parameters

5.1 Compulsory Input Parameters

1: **stand** – CHARACTER(1)

Indicates if translation/normalization is required.

stand = 'N'
There is no translation or normalization.

stand = 'Z'
There is translation to the origin (i.e., to zero).

stand = 'C'
There is translation to origin and then to the Y centroid after rotation.

stand = 'U'
There is unit normalization.

stand = 'S'
There is translation and normalization (i.e., there is standardization).

stand = 'M'
There is translation and normalization to Y scale, then translation to the Y centroid after rotation (i.e., they are matched).

Constraint: **stand** = 'N', 'Z', 'C', 'U', 'S' or 'M'.

2: **pscale** – CHARACTER(1)

Indicates if least squares scaling is to be applied after rotation.

pscale = 'S'
Scaling is applied.

pscale = 'U'
No scaling is applied.

Constraint: **pscale** = 'S' or 'U'.

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

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

X, the matrix to be rotated.

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

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

The target matrix, **y**.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the arrays **x**, **y**. (An error is raised if these dimensions are not equal.)

n, the number of points.

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

2: **m** – INTEGER

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

m, the number of dimensions.

Constraint: $\mathbf{m} \geq 1$.

5.3 Output Parameters

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

If **stand** = 'N', **x** will be unchanged.

If **stand** = 'Z', 'C', 'S' or 'M', **x** will be translated to have zero column means.

If **stand** = 'U' or 'S', **x** will be scaled to have unit sum of squares.

If **stand** = 'M', **x** will be scaled to have the same sum of squares as **y**.

2: **y**$(ldy, \mathbf{m})$ – REAL (KIND=nag_wp) array

If **stand** = 'N', **y** will be unchanged.

If **stand** = 'Z' or 'S', **y** will be translated to have zero column means.

If **stand** = 'U' or 'S', **y** will be scaled to have unit sum of squares.

If **stand** = 'C' or 'M', **y** will be translated and then after rotation translated back. The output **y** should be the same as the input **y** except for rounding errors.

3: **yhat**$(ldy, \mathbf{m})$ – REAL (KIND=nag_wp) array

The fitted matrix, $\hat{Y}$.

4: **r**$(ldr, \mathbf{m})$ – REAL (KIND=nag_wp) array

The matrix of rotations, R, see Section 9.

5: **alpha** – REAL (KIND=nag_wp)

If **pscale** = 'S' the scaling factor, α; otherwise **alpha** is not set.

6: **rss** – REAL (KIND=nag_wp)

The residual sum of squares.

7: **res**$(\mathbf{n})$ – REAL (KIND=nag_wp) array

The residuals, r_i, for $i = 1, 2, \ldots, n$.

8: **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, $\mathbf{n} < \mathbf{m}$,
or $\mathbf{m} < 1$,
or $ldx < \mathbf{n}$,
or $ldy < \mathbf{n}$,
or $ldr < \mathbf{m}$,
or **stand** $\neq$ 'N', 'Z', 'C', 'U', 'S' or 'M',
or **pscale** $\neq$ 'S' or 'U'.

ifail $= 2$

On entry, either **x** or **y** contain only zero-points (possibly after translation) when normalization is to be applied.

ifail $= 3$

The $\hat{Y}$ matrix contains only zero-points when least squares scaling is applied.

ifail $= 4$

The singular value decomposition has failed to converge. This is an unlikely error exit.

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 accuracy of the calculation of the rotation matrix largely depends upon the singular value decomposition. See the F08 Chapter Introduction for further details.

8 Further Comments

Note that if the matrix $X_c^{\mathrm{T}}Y$ is not of full rank, then the matrix of rotations, R, may not be unique even if there is a unique solution in terms of the rotated matrix, $\hat{Y}_c$. The matrix R may also include reflections as well as pure rotations, see Krzanowski (1990).

If the column dimensions of the X and Y matrices are not equal, the smaller of the two should be supplemented by columns of zeros. Adding a column of zeros to both X and Y will have the effect of allowing reflections as well as rotations.

9 Example

Three points representing the vertices of a triangle in two dimensions are input. The points are translated and rotated to match the triangle given by $(0,0)$, $(1,0)$, $(0,2)$ and scaling is applied after rotation. The target matrix and fitted matrix are printed along with additional information.

9.1 Program Text

```
function g03bc_example

fprintf('g03bc example results\n\n');

x = [0.63, 0.58;
     1.36, 0.39;
     1.01, 1.76];
y = [0,    0;
     1,    0;
     0,    2];

stand  = 'c';
pscale = 's';

[x, y, yhat, r, alpha, rss, res, ifail] = ...
  g03bc( ...
```

```
  stand, pscale, x, y);

mtitle = 'Rotation matrix';
matrix = 'General';
diag   = ' ';
[ifail] = x04ca( ...
                 matrix, diag, r, mtitle);

fprintf('\nScale factor = %7.3f\n\n', alpha);

mtitle = 'Target matrix';
[ifail] = x04ca( ...
                 matrix, diag, y, mtitle);

fprintf('\n');
mtitle = 'Fitted matrix';
[ifail] = x04ca( ...
                 matrix, diag, yhat, mtitle);

fprintf('\nRSS          = %7.3f\n', rss);
```

9.2 Program Results

```
     g03bc example results

 Rotation matrix
          1       2
 1   0.9673  0.2536
 2  -0.2536  0.9673

Scale factor =   1.556

 Target matrix
             1          2
 1      0.0000     0.0000
 2      1.0000     0.0000
 3      0.0000     2.0000

 Fitted matrix
             1          2
 1     -0.0934     0.0239
 2      1.0805     0.0259
 3      0.0130     1.9502

RSS          =   0.019
```