nag_mv_canon_var (g03acc)

1. Purpose

nag_mv_canon_var (g03acc) performs a canonical variate (canonical descrimination) analysis.

2. Specification

```
#include <nag.h>
#include <nagg03.h>

void nag_mv_canon_var(Nag_Weightstype weight, Integer n, Integer m, double x[],
             Integer tdx, Integer isx[], Integer nx, Integer ing[], Integer ng,
             double wt[], Integer nig[], double cvm[], Integer tdcvm,
             double e[], Integer tde, Integer *ncv, double cvx[],
             Integer tdcvx, double tol, Integer *irankx, NagError *fail)
```

3. Description

Let a sample of n observations on n_x variables in a data matrix come from n_g groups with $n_1, n_2, \ldots, n_{n_g}$ observations in each group, $\sum n_i = n$. Canonical variate analysis finds the linear combination of the n_x variables that maximizes the ratio of between-group to within-group variation. The variables formed, the canonical variates can then be used to discriminate between groups.

The canonical variates can be calculated from the eigenvectors of the within-group sums of squares and cross-products matrix. However, nag_mv_canon_var calculates the canonical variates by means of a singular value decomposition (SVD) of a matrix V. Let the data matrix with variable (column) means subtracted be X, and let its rank be k; then the k by $(n_g - 1)$ matrix V is given by:

$V = Q_X^T Q_g$, where Q_g is an n by $(n_g - 1)$ orthogonal matrix that defines the groups and Q_X is the first k rows of the orthogonal matrix Q either from the QR decomposition of X:

$$X = QR$$

if X is of full column rank, i.e., $k = n_x$, else from the SVD of X:

$$X = QDP^T.$$

Let the SVD of V be:

$$V = U_x \Delta U_g^T$$

then the non-zero elements of the diagonal matrix Δ, δ_i, for $i = 1, 2, \ldots, l$, are the l canonical correlations associated with the l canonical variates, where $l = \min(k, n_g)$.

The eigenvalues, λ_i^2, of the within-group sums of squares matrix are given by:

$$\lambda_i^2 = \frac{\delta_i^2}{1 - \delta_i^2}.$$

and the value of $\pi_i = \lambda_i^2 / \sum \lambda_i^2$ gives the proportion of variation explained by the ith canonical variate. The values of the π_i's give an indication as to how many canonical variates are needed to adequately describe the data, i.e., the dimensionality of the problem.

To test for a significant dimensionality greater than i the χ^2 statistic:

$$(n - 1 - n_g - \tfrac{1}{2}(k - n_g)) \sum_{j=i+1}^{l} \log(1 + \lambda_j^2)$$

can be used. This is asymptotically distributed as a χ^2 distribution with $(k-i)(n_g-1-i)$ degrees of freedom. If the test for $i = h$ is not significant, then the remaining tests for $i > h$ should be ignored.

The loadings for the canonical variates are calculated from the matrix U_x. This matrix is scaled so that the canonical variates have unit within group variance.

In addition to the canonical variates loadings the means for each canonical variate are calculated for each group.

Weights can be used with the analysis, in which case the weighted means are subtracted from each column and then each row is scaled by an amount $\sqrt{w_i}$, where w_i is the weight for the ith observation (row).

4. Parameters

weight

Input: indicates the type of weights to be used in the analysis.

If **weight** = **Nag_NoWeights**, then no weights are used.

If **weight** = **Nag_Weightsfreq**, then the weights are treated as frequencies and the effective number of observations is the sum of the weights.

If **weight** = **Nag_Weightsvar**, then the weights are treated as being inversely proportional to the variance of the observations and the effective number of observations is the number of observations with non-zero weights.

Constraint: **weight** = **Nag_NoWeights**, **Nag_Weightsfreq** or **Nag_Weightsvar**.

n

Input: the number of observations, n.
Constraint: **n** $\geq$ **nx** + **ng**.

m

Input: the total number of variables, m.
Constraint: **m** $\geq$ **nx**.

x[n][tdx]

Input: **x**$[i-1][j-1]$ must contain the ith observation for the jth variable, for $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$.

tdx

Input: the last dimension of the array **x** as declared in the calling program.
Constraint: **tdx** $\geq$ **m**.

isx[m]

Input: **isx**$[j-1]$ indicates whether or not the jth variable is to be included in the analysis.

If **isx**$[j-1] > 0$, then the variable contained in the jth column of **x** is included in the canonical variate analysis, for $j = 1, 2, \ldots, m$.

Constraint: **isx**$[j-1] > 0$ for **nx** values of j.

nx

Input: the number of variables in the analysis, n_x.
Constraint: **nx** ≥ 1.

ing[n]

Input: **ing**$[i-1]$ indicates which group the ith observation is in, for $i = 1, 2, \ldots, n$. The effective number of groups is the number of groups with non-zero membership.
Constraint: $1 \leq$ **ing** $[i-1] \leq$ **ng**, for $i = 1, 2, \ldots, n$.

ng

Input: The number of groups, n_g.

Constraint: **ng** ≥ 2.

wt[n]

Input: if **weight** = **Nag_Weightsfreq** or **Nag_Weightsvar** then the elements of **wt** must contain the weights to be used in the analysis.

If **wt**$[i-1] = 0.0$ then the ith observation is not included in the analysis.

Constraints:

wt$[i-1] \geq 0.0$, for $i = 1, 2, \ldots, n$,

$\sum_{i=1}^{n}$**wt**$[i-1] \geq$ **nx** + effective number of groups.

Note: If **weight** = **Nag_NoWeights** then **wt** is not referenced and may be set to the null pointer **NULL**, i.e (double *)0.

nig[ng]

Output: **nig**$[j-1]$ gives the number of observations in group j, for $j = 1, 2, \ldots, n_g$.

cvm[ng][tdcvm]

Output: **cvm**$[i-1][j-1]$ contains the mean of the jth canonical variate for the ith group, for $i = 1, 2, \ldots, n_g$; $j = 1, 2, \ldots, l$; the remaining columns, if any, are used as workspace.

tdcvm

Input: the last dimension of the array **cvm** as declared in the calling program.

Constraint: **tdcvm** $\geq$ **nx**.

e[min(nx,ng-1)][tde]

Output: the statistics of the canonical variate analysis.

e$[i-1][0]$, the canonical correlations, δ_i, for $i = 1, 2, \ldots, l$.

e$[i-1][1]$, the eigenvalues of the within-group sum of squares matrix, λ_i^2, for $i = 1, 2, \ldots, l$.

e$[i-1][2]$, the proportion of variation explained by the ith canonical variate, for $i = 1, 2, \ldots, l$.

e$[i-1][3]$, the χ^2 statistic for the ith canonical variate, for $i = 1, 2, \ldots, l$.

e$[i-1][4]$, the degrees of freedom for χ^2 statistic for the ith canonical variate, for $i = 1, 2, \ldots, l$.

e$[i-1][5]$, the significance level for the χ^2 statistic for the ith canonical variate, for $i = 1, 2, \ldots, l$.

tde

Input: the last dimension of the array **e** as declared in the calling program.

Constraint: **tde** ≥ 6.

ncv

Output: the number of canonical variates, l. This will be the minimum of $n_g - 1$ and the rank of **x**.

cvx[nx][tdcvx]

Output: the canonical variate loadings. **cvx**$[i-1][j-1]$ contains the loading coefficient for the ith variable on the jth canonical variate, for $i = 1, 2, \ldots, n_x$; $j = 1, 2, \ldots, l$; the remaining columns, if any, are used as workspace.

tdcvx

Input: the last dimension of the array **cvx** as declared in the calling program.

Constraint: **tdcvx** $\geq$ **ng**-1.

tol

Input: the value of **tol** is used to decide if the variables are of full rank and, if not, what is the rank of the variables. The smaller the value of **tol** the stricter the criterion for selecting the singular value decomposition. If a non-negative value of **tol** less than ***machine precision*** is entered, then the square root of ***machine precision*** is used instead.

Constraint: **tol** ≥ 0.0.

irankx

Output: the rank of the dependent variables.

If the variables are of full rank then **irankx** = **nx**.

If the variables are not of full rank then **irankx** is an estimate of the rank of the dependent variables. **irankx** is calculated as the number of singular values greater than **tol**$\times$(largest singular value).

fail
The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_BAD_PARAM
On entry, parameter **weight** had an illegal value.

NE_INT_ARG_LT
On entry, **nx** must not be less than 1: **nx** = ⟨*value*⟩.
On entry, **ng** must not be less than 2: **ng** = ⟨*value*⟩.
On entry, **tde** must not be less than 6: **tde** = ⟨*value*⟩.

NE_REAL_ARG_LT
On entry, **tol** must not be less than 0.0: **tol** = ⟨*value*⟩.

NE_2_INT_ARG_LT
On entry, **m** = ⟨*value*⟩ while **nx** = ⟨*value*⟩.
These parameters must satisfy **m** ≥ **nx**.
On entry, **tdx** = ⟨*value*⟩ while **m** = ⟨*value*⟩.
These parameters must satisfy **tdx** ≥ **m**.
On entry, **tdcvx** = ⟨*value*⟩ while **ng** = ⟨*value*⟩.
These parameters must satisfy **tdcvx** ≥ **ng**−1.
On entry, **tdcvm** = ⟨*value*⟩ while **nx** = ⟨*value*⟩.
These parameters must satisfy **tdcvm** ≥ **nx**.

NE_3_INT_ARG_CONS
On entry, **n** = ⟨*value*⟩, **nx** = ⟨*value*⟩ and **ng** = ⟨*value*⟩.
These parameters must satisfy **n** ≥ **nx** + **ng**.

NE_INTARR_INT
On entry, **ing**[⟨*value*⟩] = ⟨*value*⟩, **ng** = ⟨*value*⟩.
Constraint: $1 \leq$ **ing**$[i-1] \leq$ **ng**, $i = 1, 2, \ldots, n$.

NE_WT_ARGS
The **wt** array argument must not be NULL when the **weight** argument indicates weights.

NE_NEG_WEIGHT_ELEMENT
On entry, **wt**[⟨*value*⟩] = ⟨*value*⟩.
Constraint: When referenced, all elements of **wt** must be non-negative.

NE_VAR_INCL_INDICATED
The number of variables, **nx** in the analysis = ⟨*value*⟩, while number of variables included in the analysis via array **isx** = ⟨*value*⟩.
Constraint: these two numbers must be the same.

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

NE_CANON_CORR_1
A canonical correlation is equal to one.
This will happen if the variables provide an exact indication as to which group every observation is allocated.

NE_GROUPS
Either the effective number of groups is less than two or the effective number of groups plus the number of variables, **nx** is greater than the the effective number of observations.

NE_RANK_ZERO
The rank of the variables is zero.
This will happen if all the variables are constants.

NE_ALLOC_FAIL
Memory allocation failed.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

6. Further Comments

6.1. Accuracy

As the computation involves the use of orthogonal matrices and a singular value decomposition rather than the traditional computing of a sum of squares matrix and the use of an eigenvalue decomposition, nag_mv_canon_var should be less affected by ill conditioned problems.

6.2. References

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

Gnanadesikan R (1977) *Methods for Statistical Data Analysis of Multivariate Observations* Wiley.

Hammarling S (1985) The singular value decomposition in multivariate statistics *SIGNUM* **20(3)** 2–25.

Kendall M G and Stuart A (1979) *The Advanced Theory of Statistics (3 Volumes)* Griffin (4th Edition).

7. See Also

None.

8. Example

A sample of nine observations, each consisting of three variables plus group indicator, is read in. There are three groups. An unweighted canonical variate analysis is performed and the results printed.

8.1. Program Text

```
/* nag_mv_canon_var (g03acc) Example Program.
 *
 * Copyright 1998 Numerical Algorithms Group.
 *
 * Mark 5, 1998.
 *
 */
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg03.h>

#define NMAX 9
#define MMAX 3
#define TDE 6

main()
{

  double e[MMAX][6];
  double x[NMAX][MMAX];
  double wt[NMAX];
  double cvm[MMAX][MMAX], tol, cvx[MMAX][MMAX];

  Integer i, j, m, n;
  Integer ng;
  Integer nx;
  Integer ing[NMAX], nig[MMAX], ncv;
  Integer irx, isx[2*MMAX];
  Integer tdx=MMAX, tdc=MMAX, tde=TDE;

  char wtchar[2];

  Nag_Weightstype weight;
```

```
Vprintf("g03acc Example Program Results\n\n");

/*  Skip heading in data file */
Vscanf("%*[^\n]");

Vscanf("%ld",&n);
Vscanf("%ld",&m);
Vscanf("%ld",&nx);
Vscanf("%ld",&ng);
Vscanf("%s",wtchar);
if (n <= NMAX && m <= MMAX)
  {
    if (*wtchar == 'W' || *wtchar == 'V')
      {
        for (i = 0; i < n; ++i)
          {
            for (j = 0; j < m; ++j)
              Vscanf("%lf",&x[i][j]);
            Vscanf("%lf",&wt[i]);
            Vscanf("%ld",&ing[i]);
          }
        if (*wtchar == 'W')
          weight = Nag_Weightsfreq;
        else
          weight = Nag_Weightsvar;
      }
    else
      {
        for (i = 0; i < n; ++i)
          {
            for (j = 0; j < m; ++j)
              Vscanf("%lf",&x[i][j]);
            Vscanf("%ld",&ing[i]);
          }
        weight = Nag_NoWeights;
      }
    for (j = 0; j < m; ++j)
      Vscanf("%ld",&isx[j]);

    tol = 1e-6;
    g03acc(weight, n, m, (double *)x, tdx, isx, nx, ing, ng, wt, nig,
           (double *)cvm, tdc, (double *)e, tde, &ncv, (double *)cvx,
           tdc, tol, &irx, NAGERR_DEFAULT);

    Vprintf("%s%2ld\n\n","Rank of x = ",irx);
    Vprintf("Canonical    Eigenvalues     Percentage   CHISQ\
      DF          SIG \n");
    Vprintf("Correlations                 Variation\n");
    for (i = 0; i < ncv; ++i)
      {
        for (j = 0; j < 6; ++j)
          Vprintf("%12.4f",e[i][j]);
        Vprintf("\n");
      }
    Vprintf("\nCanonical Coefficients for X\n");
    for (i = 0; i < nx; ++i)
      {
        for (j = 0; j < ncv; ++j)
          Vprintf("%9.4f",cvx[i][j]);
        Vprintf("\n");
      }
    Vprintf("\nCanonical variate means\n");
    for (i = 0; i < ng; ++i)
      {
        for (j = 0; j < ncv; ++j)
          Vprintf("%9.4f",cvm[i][j]);
        Vprintf("\n");
      }
    exit(EXIT_SUCCESS);
```

```
    }
  else
    {
      Vprintf("Incorrect input value of n or m.\n");
      exit(EXIT_FAILURE);
    }
}
```

8.2. Program Data

```
g03acc Example Program Data
 9 3 3 3 U
 13.3 10.6 21.2 1
 13.6 10.2 21.0 2
 14.2 10.7 21.1 3
 13.4  9.4 21.0 1
 13.2  9.6 20.1 2
 13.9 10.4 19.8 3
 12.9 10.0 20.5 1
 12.2  9.9 20.7 2
 13.9 11.0 19.1 3
  1    1    1
```

8.3. Program Results

```
g03acc Example Program Results

Rank of x =  3

Canonical    Eigenvalues     Percentage   CHISQ        DF          SIG
Correlations                 Variation
      0.8826      3.5238      0.9795      7.9032      6.0000      0.2453
      0.2623      0.0739      0.0205      0.3564      2.0000      0.8368

Canonical Coefficients for X
  -1.7070   0.7277
  -1.3481   0.3138
   0.9327   1.2199

Canonical variate means
   0.9841   0.2797
   1.1805  -0.2632
  -2.1646  -0.0164
```