nag_mv_discrim_group (g03dcc)

1. Purpose

nag_mv_discrim_group (g03dcc) allocates observations to groups according to selected rules. It is intended for use after nag_mv_discrim (g03dac).

2. Specification

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

void nag_mv_discrim_group(Nag_DiscrimMethod type, Nag_GroupCovars equal,
              Nag_PriorProbability priors, Integer nvar,
             Integer ng, Integer nig[], double gmean[], Integer tdg,
             double gc[], double det[], Integer nobs, Integer m,
             Integer isx[], double x[], Integer tdx,
             double prior[], double p[], Integer tdp, Integer iag[],
             Boolean atiq, double ati[], NagError *fail)
```

3. Description

Discriminant analysis is concerned with the allocation of observations to groups using information from other observations whose group membership is known, X_t; these are called the training set. Consider p variables observed on n_g populations or groups. Let $\bar{x}_j$ be the sample mean and S_j the within-group variance-covariance matrix for the jth group; these are calculated from a training set of n observations with n_j observations in the jth group, and let x_k be the kth observation from the set of observations to be allocated to the n_g groups. The observation can be allocated to a group according to a selected rule. The allocation rule or discriminant function will be based on the distance of the observation from an estimate of the location of the groups, usually the group means. A measure of the distance of the observation from the jth group mean is given by the Mahalanobis distance, D^2_{kj}:

$$D^2_{kj} = (x_k - \bar{x}_j)^T S_j^{-1} (x_k - \bar{x}_j). \tag{1}$$

If the pooled estimate of the variance-covariance matrix S is used rather than the within-group variance-covariance matrices, then the distance is:

$$D^2_{kj} = (x_k - \bar{x}_j)^T S^{-1} (x_k - \bar{x}_j). \tag{2}$$

Instead of using the variance-covariance matrices S and S_j, nag_mv_discrim_group uses the upper triangular matrices R and R_j supplied by nag_mv_discrim (g03dac) such that $S = R^T R$ and $S_j = R_j^T R_j$. D^2_{kj} can then be calculated as $z^T z$ where $R_j z = (x_k - \bar{x}_j)$ or $Rz = (x_k - \bar{x}_j)$ as appropriate.

In addition to the distances, a set of prior probabilities of group membership, π_j, for $j = 1, 2, \ldots, n_g$, may be used, with $\sum \pi_j = 1$. The prior probabilities reflect the user's view as to the likelihood of the observations coming from the different groups. Two common cases for prior probabilities are $\pi_1 = \pi_2 = \ldots = \pi_{n_g}$, that is, equal prior probabilities, and $\pi_j = n_j/n$, for $j = 1, 2, \ldots, n_g$, that is, prior probabilities proportional to the number of observations in the groups in the training set.

nag_mv_discrim_group uses one of four allocation rules. In all four rules the p variables are assumed to follow a multivariate Normal distribution with mean μ_j and variance-covariance matrix Σ_j if the observation comes from the jth group. The different rules depend on whether or not the within-group variance-covariance matrices are assumed equal, i.e., $\Sigma_1 = \Sigma_2 = \ldots = \Sigma_{n_g}$, and whether a predictive or estimative approach is used. If $p(x_k|\mu_j, \Sigma_j)$ is the probability of observing the observation x_k from group j, then the posterior probability of belonging to group j is:

$$p(j|x_k, \mu_j, \Sigma_j) \propto p(x_k|\mu_j, \Sigma_j)\pi_j. \tag{3}$$

In the estimative approach, the parameters μ_j and Σ_j in (3) are replaced by their estimates calculated from X_t. In the predictive approach, a non-informative prior distribution is used for the parameters and a posterior distribution for the parameters, $p(\mu_j, \Sigma_j|X_t)$, is found. A predictive distribution is then obtained by integrating $p(j|x_k, \mu_j, \Sigma_j)p(\mu_j, \Sigma_j|X)$ over the parameter space. This predictive distribution then replaces $p(x_k|\mu_j, \Sigma_j)$ in (3). See Aitchison and Dunsmore (1975), Aitchison *et al.* (1977) and Moran and Murphy (1979) for further details.

The observation is allocated to the group with the highest posterior probability. Denoting the posterior probabilities, $p(j|x_k, \mu_j, \Sigma_j)$, by q_j, the four allocation rules are:

(i) Estimative with equal variance-covariance matrices – Linear Discrimination.

$$\log q_j \propto -\tfrac{1}{2}D_{kj}^2 + \log \pi_j$$

(ii) Estimative with unequal variance-covariance matrices – Quadratic Discrimination.

$$\log q_j \propto -\tfrac{1}{2}D_{kj}^2 + \log \pi_j - \tfrac{1}{2}\log|S_j|$$

(iii) Predictive with equal variance-covariance matrices.

$$q_j^{-1} \propto ((n_j+1)/n_j)^{p/2}\{1 + [n_j/((n-n_g)(n_j+1))]D_{kj}^2\}^{(n+1-n_g)/2}$$

(iv) Predictive with unequal variance-covariance matrices.

$$q_j^{-1} \propto C\{((n_j^2-1)/n_j)|S_j|\}^{p/2}\{1 + (n_j/(n_j^2-1))D_{kj}^2\}^{n_j/2}$$

where

$$C = \frac{\Gamma(\frac{1}{2}(n_j - p))}{\Gamma(\frac{1}{2}n_j)}$$

In the above the appropriate value of D_{kj}^2 from (1) or (2) is used. The values of the q_j are standardized so that,

$$\sum_{j=1}^{n_g} q_j = 1.$$

Moran and Murphy (1979) show the similarity between the predictive methods and methods based upon likelihood ratio tests.

In addition to allocating the observation to a group, nag_mv_discrim_group computes an atypicality index, $I_j(x_k)$. This represents the probability of obtaining an observation more typical of group j than the observed x_k (see Aitchison and Dunsmore (1975) and Aitchison *et al.* (1977)). The atypicality index is computed as:

$$I_j(x_k) = P(B \leq z : \tfrac{1}{2}p, \tfrac{1}{2}(n_j - d))$$

where $P(B \leq \beta : a, b)$ is the lower tail probability from a beta distribution where, for unequal within-group variance-covariance matrices,

$$z = D_{kj}^2/(D_{kj}^2 + (n_j^2 - 1)/n_j),$$

and for equal within-group variance-covariance matrices,

$$z = D_{kj}^2/(D_{kj}^2 + (n - n_g)(n_j - 1)/n_j).$$

If $I_j(x_k)$ is close to 1 for all groups it indicates that the observation may come from a grouping not represented in the training set. Moran and Murphy (1979) provide a frequentist interpretation of $I_j(x_k)$.

4. Parameters

type

Input: indicates whether the estimative or predictive approach is to be used.

If **type** = **Nag_DiscrimEstimate**, the estimative approach is used.
If **type** = **Nag_DiscrimPredict**, the predictive approach is used.

Constraint: **type** = **Nag_DiscrimEstimate** or **Nag_DiscrimPredict**.

equal

Input: indicates whether or not the within-group variance-covariance matrices are assumed to be equal and the pooled variance-covariance matrix used.

If **equal** = **Nag_EqualCovar**, the within-group variance-covariance matrices are assumed equal and the matrix R stored in the first $p(p+1)/2$ elements of **gc** is used.
If **equal** = **Nag_NotEqualCovar**, the within-group variance-covariance matrices are assumed to be unequal and the matrices R_i, for $i = 1, 2, \ldots, n_g$, stored in the remainder of **gc** are used.

Constraint: **equal** = **Nag_EqualCovar** or **Nag_NotEqualCovar**.

priors

Input: indicates the form of the prior probabilities to be used.

If **priors** = **Nag_EqualPrior**, equal prior probabilities are used.
If **priors** = **Nag_GroupSizePrior**, prior probabilities proportional to the group sizes in the training set, n_j, are used.
If **priors** = **Nag_UserPrior**, the prior probabilities are input in **prior**.

Constraint: **priors** = **Nag_EqualPrior**, **Nag_GroupSizePrior** or **Nag_UserPrior**.

nvar

Input: the number of variables, p, in the variance-covariance matrices as specified to nag_mv_discrim (g03dac).

Constraint: **nvar** ≥ 1.

ng

Input: the number of groups, n_g.

Constraint: **ng** ≥ 2.

nig[ng]

Input: the number of observations in each group training set, n_j.

Constraints:

If **equal** = **Nag_EqualCovar**, **nig**$[j-1] > 0$, for $j = 1, 2, \ldots, n_g$ and $\sum_{j=1}^{n_g}$ **nig**$[j-1] >$ **ng** + **nvar**.
If **equal** = **Nag_NotEqualCovar**, **nig**$[j-1] >$ **nvar**, for $j = 1, 2, \ldots, n_g$.

gmean[ng][tdg]

Input: the jth row of **gmean** contains the means of the p variables for the jth group, for $j = 1, 2, \ldots, n_j$. These are returned by nag_mv_discrim (g03dac).

tdg

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

Constraint: **tdg** $\geq$ **nvar**

gc[(ng+1)*nvar*(nvar+1)/2]

Input: the first $p(p+1)/2$ elements of **gc** should contain the upper triangular matrix R and the next n_g blocks of $p(p+1)/2$ elements should contain the upper triangular matrices R_j.

All matrices must be stored packed by column. These matrices are returned by nag_mv_discrim (g03dac). If **equal**=**Nag_EqualCovar** only the first $p(p+1)/2$ elements are referenced, if **equal** = **Nag_NotEqualCovar** only the elements $p(p+1)/2$ to $(n_g+1)p(p+1)/2-1$ are referenced.

Constraints:

If **equal** = **Nag_EqualCovar**, the diagonal elements of R must be $\neq 0.0$,
If **equal** = **Nag_NotEqualCovar**, the diagonal elements of the R_j must be $\neq 0.0$, for $j = 1, 2, \ldots, n_g$.

det[ng]

Input: if **equal** = **Nag_NotEqualCovar** the logarithms of the determinants of the within-group variance-covariance matrices as returned by nag_mv_discrim (g03dac). Otherwise **det** is not referenced.

nobs

Input: the number of observations in **x** which are to be allocated.

Constraint: **nobs** ≥ 1.

m

Input: the number of variables in the data array **x**.

Constraint: **m** $\geq$ **nvar**.

isx[m]

Input: **isx**$[l-1]$ indicates if the lth variable in **x** is to be included in the distance calculations. If **isx**$[l-1] > 0$ the lth variable is included, for $l = 1, 2, \ldots,$**m**; otherwise the lth variable is not referenced.
Constraint: **isx**$[l-1] > 0$ for **nvar** values of l.

x[nobs][tdx]

Input: **x**$[k-1][l-1]$ must contain the kth observation for the lth variable, for $k = 1, 2, \ldots,$**nobs**; $l = 1, 2, \ldots,$**m**.

tdx

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

prior[ng]

Input: if **priors** = **Nag_UserPrior**, the prior probabilities for the n_g groups.

Constraint: if **priors** = **Nag_UserPrior**, then **prior**$[j-1] > 0.0$ for $j = 1, 2, \ldots, n_g$ and $\left|1 - \sum_{j=1}^{n_g} \mathbf{prior}[j-1]\right| \leq 10 \times$ ***machine precision***.

Output: if **priors** = **Nag_GroupSizePrior**, the computed prior probabilities in proportion to group sizes for the n_g groups. If **priors** = **Nag_UserPrior**, the input prior probabilities will be unchanged, and if **priors** = **Nag_EqualPrior**, **prior** is not set.

p[nobs][tdp]

Output: **p**$[k-1][j-1]$ contains the posterior probability p_{kj} for allocating the kth observation to the jth group, for $k = 1, 2, \ldots,$**nobs**; $j = 1, 2, \ldots, n_g$.

tdp

Input: the last dimension of the array **p** and **ati** as declared in the calling program.
Constraint: **tdp** $\geq$ **ng**.

iag[nobs]

Output: the groups to which the observations have been allocated.

atiq

Input: **atiq** must be **TRUE** if atypicality indices are required. If **atiq** is **FALSE**, the array **ati** is not set.

ati[nobs][tdp]

Output: if **atiq** is **TRUE**, **ati**$[k-1][j-1]$ will contain the atypicality index for the kth observation with respect to the jth group, for $k = 1, 2, \ldots,$**nobs**; $j = 1, 2, \ldots, n_g$. If **atiq** is **FALSE**, **ati** is not set.

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 **type** had an illegal value.
On entry, parameter **equal** had an illegal value.
On entry, parameter **priors** had an illegal value.

NE_INT_ARG_LT

On entry, **nvar** must not be less than 1: **nvar** = ⟨*value*⟩.
On entry, **ng** must not be less than 2: **ng** = ⟨*value*⟩.
On entry, **nobs** must not be less than 1: **nobs** = ⟨*value*⟩.

NE_2_INT_ARG_LT

On entry, **m** = ⟨*value*⟩ while **nvar** = ⟨*value*⟩.
These parameters must satisfy **m** $\geq$ **nvar**.
On entry, **tdx** = ⟨*value*⟩ while **m** = ⟨*value*⟩.
These parameters must satisfy **tdx** $\geq$ **m**.
On entry, **tdp** = ⟨*value*⟩ while **ng** = ⟨*value*⟩.
These parameters must satisfy **tdp** $\geq$ **ng**.
On entry, **tdg** = ⟨*value*⟩ while **nvar** = ⟨*value*⟩.
These parameters must satisfy **tdg** $\geq$ **nvar**.

NE_VAR_INCL_INDICATED

The number of variables, **nvar** 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_INTARR

On entry, **nig**[⟨*value*⟩] = ⟨*value*⟩.
Constraint: **nig**[$i-1$] > 0, $i = 1, 2, \ldots,$**ng** when **equal** = **Nag_EqualCovar**.

NE_INTARR_INT

On entry, **nig**[⟨*value*⟩] = ⟨*value*⟩, **nvar** = ⟨*value*⟩.
Constraint: **nig**[$i-1$] > **nvar**, $i = 1, 2, \ldots,$**ng** when **equal** = **Nag_NotEqualCovar**.

NE_REALARR

On entry, **prior**[⟨*value*⟩] = ⟨*value*⟩.
Constraint: **prior**[$j-1$] > 0, j=1,2,...,**ng** when **priors** = **Nag_UserPrior**.

NE_PRIOR_SUM

On entry, $\sum_{j=1}^{\mathbf{ng}}$ **prior**[$j-1$] = ⟨*value*⟩.
Constraint: $\sum_{j=1}^{\mathbf{ng}}$ **prior**[$j-1$] must be within 10 × ***machine precision*** of 1 when **priors** = **Nag_UserPrior**.

NE_GROUP_SUM

On entry, the $\sum_{j=1}^{\mathbf{ng}}$ **nig**[$j-1$] = ⟨*value*⟩, **ng** = ⟨*value*⟩, **nvar** = ⟨*value*⟩.
Constraint: $\sum_{j=1}^{\mathbf{ng}}$ **nig**[$j-1$] > **ng** + **nvar** when **equal** = **Nag_EqualCovar**.

NE_DIAG_0_COND

A diagonal element of R is zero when **equal** = **Nag_EqualCovar**.

NE_DIAG_0_J_COND

A diagonal element of R is zero for some j, when **equal** = **Nag_NotEqualCovar**

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

The distances D_{kj}^2 can be computed using nag_mv_discrim_mahaldist (g03dbc) if other forms of discrimination are required.

6.1. Accuracy

The accuracy of the returned posterior probabilities will depend on the accuracy of the input R or R_j matrices. The atypicality index should be accurate to four significant places.

6.2. References

Aitchison J and Dunsmore I R (1975) *Statistical Prediction Analysis* Cambridge.
Aitchison J, Habbema J D F and Kay J W (1977) A critical comparison of two methods of statistical discrimination *Appl. Statist.* **26** 15–25.
Kendall M G and Stuart A (1976) *The Advanced Theory of Statistics (Volume 3)* Griffin (3rd Edition).
Krzanowski W J (1990) *Principles of Multivariate Analysis* Oxford University Press.
Moran M A and Murphy B J (1979) A closer look at two alternative methods of statistical discrimination *Appl. Statist.* **28** 223–232.
Morrison D F (1967) *Multivariate Statistical Methods* McGraw-Hill.

7. See Also

nag_mv_discrim_mahaldist (g03dbc)
nag_mv_discrim (g03dac)

8. Example

The data, taken from Aitchison and Dunsmore (1975), is concerned with the diagnosis of three 'types' of Cushing's syndrome. The variables are the logarithms of the urinary excretion rates (mg/24hr) of two steroid metabolites. Observations for a total of 21 patients are input and the group means and R matrices are computed by nag_mv_discrim (g03dac). A further six observations of unknown type are input and allocations made using the predictive approach and under the assumption that the within-group covariance matrices are not equal. The posterior probabilities of group membership, q_j, and the atypicality index are printed along with the allocated group. The atypicality index shows that observations 5 and 6 do not seem to be typical of the three types present in the initial 21 observations.

8.1. Program Text

```
/* nag_mv_discrim_group (g03dcc) 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 21
#define MMAX 2
#define GPMAX 3

main()
{
  double stat;
  double ati[NMAX][GPMAX], det[GPMAX],
  gc[(GPMAX+1)*MMAX*(MMAX+1)/2], gmean[GPMAX][MMAX],
  p[NMAX][GPMAX], prior[GPMAX],
  wt[NMAX], x[NMAX][MMAX];
  double df;
  double sig;
  double *wtptr=0;

  Integer nobs, nvar;
  Integer i, j, m, n;
```

```
Integer  iag[NMAX], ing[NMAX], isx[MMAX],
nig[GPMAX];
Integer ng;
Integer tdgmean=MMAX, tdp=GPMAX, tdx=MMAX;

Boolean atiq = TRUE;

char char_type[2];
char char_equal[2];
char weight[2];

Nag_DiscrimMethod type;
Nag_GroupCovars equal;

Vprintf("g03dcc Example Program Results\n\n");

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

Vscanf("%ld",&n);
Vscanf("%ld",&m);
Vscanf("%ld",&nvar);
Vscanf("%ld",&ng);
Vscanf("%s",weight);

if (n <= NMAX && m <= MMAX)
  {
    if (*weight == 'W')
      {
        for (i = 0; i < n; ++i)
          {
            for (j = 0; j < m; ++j)
              Vscanf("%lf",&x[i][j]);
            Vscanf("%ld",&ing[i]);
            Vscanf("%lf",&wt[i]);
          }
        wtptr = wt;
      }
    else
      {
        for (i = 0; i < n; ++i)
          {
            for (j = 0; j < m; ++j)
              Vscanf("%lf",&x[i][j]);
            Vscanf("%ld",&ing[i]);
          }
      }
    for (j = 0; j < m; ++j)
      Vscanf("%ld",&isx[j]);

    g03dac(n, m, (double *)x, tdx, isx, nvar, ing, ng, wtptr, nig,
           (double *)gmean, tdgmean, det, gc, &stat, &df, &sig, NAGERR_DEFAULT);
    Vscanf("%ld",&nobs);
    Vscanf("%s",char_equal);
    Vscanf("%s",char_type);
    if (nobs <= NMAX)
      {
        for (i = 0; i < nobs; ++i)
          {
            for (j = 0; j < m; ++j)
              {
                Vscanf("%lf",&x[i][j]);
              }
          }

        if (*char_type == 'E')
          type = Nag_DiscrimEstimate;
        else if (*char_type == 'P')
          type = Nag_DiscrimPredict;
```

```
          if (*char_equal == 'E')
            equal = Nag_EqualCovar;
          else if (*char_equal == 'U')
            equal = Nag_NotEqualCovar;

          g03dcc(type, equal, Nag_EqualPrior, nvar, ng, nig, (double *)gmean,
                 tdgmean, gc, det, nobs, m, isx, (double *)x, tdx, prior, (double *)p,
                 tdp, iag, atiq, (double *)ati, NAGERR_DEFAULT);

          Vprintf("\n");
          Vprintf("    Obs       Posterior        Allocated ");
          Vprintf("      Atypicality ");
          Vprintf("\n");
          Vprintf("              probabilities    to group      index ");
          Vprintf("\n");
          Vprintf("\n");
          for (i = 0; i < nobs; ++i)
            {
              Vprintf(" %6ld     ",i+1);
              for (j = 0; j < ng; ++j)
                {
                  Vprintf("%6.3f",p[i][j]);
                }
              Vprintf("  %6ld    ",iag[i]);
              for (j = 0; j < ng; ++j)
                {
                  Vprintf("%6.3f",ati[i][j]);
                }
              Vprintf("\n");
            }
        }

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

8.2. Program Data

```
g03dcc Example Program Data
  21 2 2 3 U
  1.1314    2.4596    1
  1.0986    0.2624    1
  0.6419   -2.3026    1
  1.3350   -3.2189    1
  1.4110    0.0953    1
  0.6419   -0.9163    1
  2.1163    0.0000    2
  1.3350   -1.6094    2
  1.3610   -0.5108    2
  2.0541    0.1823    2
  2.2083   -0.5108    2
  2.7344    1.2809    2
  2.0412    0.4700    2
  1.8718   -0.9163    2
  1.7405   -0.9163    2
  2.6101    0.4700    2
  2.3224    1.8563    3
  2.2192    2.0669    3
  2.2618    1.1314    3
  3.9853    0.9163    3
  2.7600    2.0281    3
   1         1
   6 U P
  1.6292   -0.9163
  2.5572    1.6094
  2.5649   -0.2231
```

```
 0.9555   -2.3026
 3.4012   -2.3026
 3.0204   -0.2231
```

8.3. Program Results

```
g03dcc Example Program Results


   Obs        Posterior          Allocated         Atypicality
              probabilities      to group        index

     1       0.094 0.905 0.002        2      0.596 0.254 0.975
     2       0.005 0.168 0.827        3      0.952 0.836 0.018
     3       0.019 0.920 0.062        2      0.954 0.797 0.912
     4       0.697 0.303 0.000        1      0.207 0.860 0.993
     5       0.317 0.013 0.670        3      0.991 1.000 0.984
     6       0.032 0.366 0.601        3      0.981 0.978 0.887
```