NAG Library Routine Document

G03EFF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G03EFF performs K-means cluster analysis.

2 Specification

```
SUBROUTINE G03EFF (WEIGHT, N, M, X, LDX, ISX, NVAR, K, CMEANS, LDC, WT,    &
                   INC, NIC, CSS, CSW, MAXIT, IWK, WK, IFAIL)
INTEGER            N, M, LDX, ISX(M), NVAR, K, LDC, INC(N), NIC(K),        &
                   MAXIT, IWK(N+3*K), IFAIL
REAL (KIND=nag_wp) X(LDX,M), CMEANS(LDC,NVAR), WT(*), CSS(K), CSW(K),      &
                   WK(N+2*K)
CHARACTER(1)       WEIGHT
```

3 Description

Given n objects with p variables measured on each object, x_{ij}, for $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,p$, G03EFF allocates each object to one of K groups or clusters to minimize the within-cluster sum of squares:

$$\sum_{k=1}^{K}\sum_{i \in S_k}\sum_{j=1}^{p}\left(x_{ij} - \bar{x}_{kj}\right)^2,$$

where S_k is the set of objects in the kth cluster and $\bar{x}_{kj}$ is the mean for the variable j over cluster k. This is often known as K-means clustering.

In addition to the data matrix, a K by p matrix giving the initial cluster centres for the K clusters is required. The objects are then initially allocated to the cluster with the nearest cluster mean. Given the initial allocation, the procedure is to iteratively search for the K-partition with locally optimal within-cluster sum of squares by moving points from one cluster to another.

Optionally, weights for each object, w_i, can be used so that the clustering is based on within-cluster weighted sums of squares:

$$\sum_{k=1}^{K}\sum_{i \in S_k}\sum_{j=1}^{p} w_i\left(x_{ij} - \tilde{x}_{kj}\right)^2,$$

where $\tilde{x}_{kj}$ is the weighted mean for variable j over cluster k.

The routine is based on the algorithm of Hartigan and Wong (1979).

4 References

Everitt B S (1974) *Cluster Analysis* Heinemann

Hartigan J A and Wong M A (1979) Algorithm AS 136: A K-means clustering algorithm *Appl. Statist.* **28** 100–108

Kendall M G and Stuart A (1976) *The Advanced Theory of Statistics (Volume 3)* (3rd Edition) Griffin

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

5 Parameters

1: WEIGHT – CHARACTER(1) *Input*

On entry: indicates if weights are to be used.

WEIGHT = 'U'
No weights are used.

WEIGHT = 'W'
Weights are used and must be supplied in WT.

Constraint: WEIGHT = 'U' or 'W'.

2: N – INTEGER *Input*

On entry: n, the number of objects.

Constraint: $\mathrm{N} > 1$.

3: M – INTEGER *Input*

On entry: the total number of variables in array X.

Constraint: $\mathrm{M} \geq \mathrm{NVAR}$.

4: $\mathrm{X}(\mathrm{LDX}, \mathrm{M})$ – REAL (KIND=nag_wp) array *Input*

On entry: $\mathrm{X}(i, j)$ must contain the value of the jth variable for the ith object, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, \mathrm{M}$.

5: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G03EFF is called.

Constraint: $\mathrm{LDX} \geq \mathrm{N}$.

6: ISX(M) – INTEGER array *Input*

On entry: $\mathrm{ISX}(j)$ indicates whether or not the jth variable is to be included in the analysis. If $\mathrm{ISX}(j) > 0$, the variable contained in the jth column of X is included, for $j = 1, 2, \ldots, \mathrm{M}$.

Constraint: $\mathrm{ISX}(j) > 0$ for NVAR values of j.

7: NVAR – INTEGER *Input*

On entry: p, the number of variables included in the sums of squares calculations.

Constraint: $1 \leq \mathrm{NVAR} \leq \mathrm{M}$.

8: K – INTEGER *Input*

On entry: K, the number of clusters.

Constraint: $\mathrm{K} \geq 2$.

9: $\mathrm{CMEANS}(\mathrm{LDC}, \mathrm{NVAR})$ – REAL (KIND=nag_wp) array *Input/Output*

On entry: $\mathrm{CMEANS}(i, j)$ must contain the value of the jth variable for the ith initial cluster centre, for $i = 1, 2, \ldots, K$ and $j = 1, 2, \ldots, p$.

On exit: $\mathrm{CMEANS}(i, j)$ contains the value of the jth variable for the ith computed cluster centre, for $i = 1, 2, \ldots, K$ and $j = 1, 2, \ldots, p$.

10: LDC – INTEGER *Input*

On entry: the first dimension of the array CMEANS as declared in the (sub)program from which G03EFF is called.

Constraint: LDC $\geq$ K.

11: WT($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array WT must be at least N if WEIGHT = 'W', and at least 1 otherwise.

On entry: if WEIGHT = 'W', the first n elements of WT must contain the weights to be used.

If WT(i) = 0.0, the ith observation is not included in the analysis. The effective number of observation is the sum of the weights.

If WEIGHT = 'U', WT is not referenced and the effective number of observations is n.

Constraint: if WEIGHT = 'W', WT(i) $\geq$ 0.0 and WT(i) $>$ 0.0 for at least two values of i, for $i = 1, 2, \ldots, n$.

12: INC(N) – INTEGER array *Output*

On exit: INC(i) contains the cluster to which the ith object has been allocated, for $i = 1, 2, \ldots, n$.

13: NIC(K) – INTEGER array *Output*

On exit: NIC(i) contains the number of objects in the ith cluster, for $i = 1, 2, \ldots, K$.

14: CSS(K) – REAL (KIND=nag_wp) array *Output*

On exit: CSS(i) contains the within-cluster (weighted) sum of squares of the ith cluster, for $i = 1, 2, \ldots, K$.

15: CSW(K) – REAL (KIND=nag_wp) array *Output*

On exit: CSW(i) contains the within-cluster sum of weights of the ith cluster, for $i = 1, 2, \ldots, K$. If WEIGHT = 'U', the sum of weights is the number of objects in the cluster.

16: MAXIT – INTEGER *Input*

On entry: the maximum number of iterations allowed in the analysis.

Suggested value: MAXIT = 10.

Constraint: MAXIT $>$ 0.

17: IWK(N + 3 × K) – INTEGER array *Workspace*

18: WK(N + 2 × K) – REAL (KIND=nag_wp) array *Workspace*

19: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, WEIGHT $\neq$ 'W' or 'U',
or N < 2,
or NVAR < 1,
or M $<$ NVAR,
or K < 2,
or LDX $<$ N,
or LDC $<$ K,
or MAXIT ≤ 0.

IFAIL $= 2$

On entry, WEIGHT = 'W' and a value of WT$(i) < 0.0$ for some i,
or WEIGHT = 'W' and WT$(i) = 0.0$ for all or all but one values of i.

IFAIL $= 3$

On entry, the number of positive values in ISX does not equal NVAR.

IFAIL $= 4$

On entry, at least one cluster is empty after the initial assignment. Try a different set of initial cluster centres in CMEANS and also consider decreasing the value of K. The empty clusters may be found by examining the values in NIC.

IFAIL $= 5$

Convergence has not been achieved within the maximum number of iterations given by MAXIT. Try increasing MAXIT and, if possible, use the returned values in CMEANS as the initial cluster centres.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

G03EFF produces clusters that are locally optimal; the within-cluster sum of squares may not be decreased by transferring a point from one cluster to another, but different partitions may have the same or smaller within-cluster sum of squares.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time per iteration is approximately proportional to npK.

10 Example

The data consists of observations of five variables on twenty soils (see Hartigan and Wong (1979)). The data is read in, the K-means clustering performed and the results printed.

10.1 Program Text

```
    Program g03effe

!     G03EFF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g03eff, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, k, ldc, ldx, lwt, m,       &
                                          maxit, n, nvar
      Character (1)                    :: weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: cmeans(:,:), css(:), csw(:), wk(:),  &
                                          wt(:), x(:,:)
      Integer, Allocatable             :: inc(:), isx(:), iwk(:), nic(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: count
!     .. Executable Statements ..
      Write (nout,*) 'G03EFF Example Program Results'
      Write (nout,*)

!     Skip heading in the data file
      Read (nin,*)

!     Read in the problem size and control parameters
      Read (nin,*) weight, n, m, k, maxit

      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      ldx = n
      Allocate (x(ldx,m),wt(n),isx(m))

!     Read in data
      If (lwt>0) Then
        Read (nin,*)(x(i,1:m),wt(i),i=1,n)
      Else
        Read (nin,*)(x(i,1:m),i=1,n)
      End If

!     Read in variable inclusion flags
      Read (nin,*) isx(1:m)

!     Calculate NVAR
      nvar = count(isx(1:m)==1)
```

```
      ldc = k
      Allocate (cmeans(ldc,nvar),inc(n),nic(k),css(k),csw(k),iwk(n+3*k), &
        wk(n+2*k))

!     Read in the initial cluster centres
      Read (nin,*)(cmeans(i,1:nvar),i=1,k)

!     Perform k means clustering
      ifail = 0
      Call g03eff(weight,n,m,x,ldx,isx,nvar,k,cmeans,ldc,wt,inc,nic,css,csw, &
        maxit,iwk,wk,ifail)

!     Display results
      Write (nout,*) ' The cluster each point belongs to'
      Write (nout,99999) inc(1:n)
      Write (nout,*)
      Write (nout,*) ' The number of points in each cluster'
      Write (nout,99999) nic(1:k)
      Write (nout,*)
      Write (nout,*) ' The within-cluster sum of weights of each cluster'
      Write (nout,99998) csw(1:k)
      Write (nout,*)
      Write (nout,*) ' The within-cluster sum of squares of each cluster'
      Write (nout,99997) css(1:k)
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04caf('General',' ',k,nvar,cmeans,ldc,'The final cluster centres', &
        ifail)

99999 Format (1X,10I6)
99998 Format (1X,5F9.2)
99997 Format (1X,5F13.4)
    End Program g03effe
```

10.2 Program Data

```
G03EFF Example Program Data
'u' 20 5 3 10                : WEIGHT,N,M,K,MAXIT
77.3 13.0  9.7 1.5 6.4
82.5 10.0  7.5 1.5 6.5
66.9 20.6 12.5 2.3 7.0
47.2 33.8 19.0 2.8 5.8
65.3 20.5 14.2 1.9 6.9
83.3 10.0  6.7 2.2 7.0
81.6 12.7  5.7 2.9 6.7
47.8 36.5 15.7 2.3 7.2
48.6 37.1 14.3 2.1 7.2
61.6 25.5 12.9 1.9 7.3
58.6 26.5 14.9 2.4 6.7
69.3 22.3  8.4 4.0 7.0
61.8 30.8  7.4 2.7 6.4
67.7 25.3  7.0 4.8 7.3
57.2 31.2 11.6 2.4 6.5
67.2 22.7 10.1 3.3 6.2
59.2 31.2  9.6 2.4 6.0
80.2 13.2  6.6 2.0 5.8
82.2 11.1  6.7 2.2 7.2
69.7 20.7  9.6 3.1 5.9         : End of X
1 1 1 1 1                      : ISX
82.5 10.0  7.5 1.5 6.5
47.8 36.5 15.7 2.3 7.2
67.2 22.7 10.1 3.3 6.2         : End of CMEANS
```

10.3 Program Results

```
 G03EFF Example Program Results

  The cluster each point belongs to
      1     1     3     2     3     1     1     2     2     3
      3     3     3     3     3     3     3     1     1     3

  The number of points in each cluster
      6     3    11

  The within-cluster sum of weights of each cluster
      6.00     3.00    11.00

  The within-cluster sum of squares of each cluster
       46.5717      20.3800     468.8964

 The final cluster centres
              1          2          3          4          5
 1     81.1833    11.6667     7.1500     2.0500     6.6000
 2     47.8667    35.8000    16.3333     2.4000     6.7333
 3     64.0455    25.2091    10.7455     2.8364     6.6545
```