NAG Library Routine Document

G03GAF

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

G03GAF performs a mixture of Normals (Gaussians) for a given (co)variance structure.

2 Specification

```
SUBROUTINE G03GAF (N, M, X, LDX, ISX, NVAR, NG, POPT, PROB, LPROB,          &
                   NITER, RITER, W, G, SOPT, S, LDS, SDS, F, TOL,           &
                   LOGLIK, IFAIL)
INTEGER            N, M, LDX, ISX(M), NVAR, NG, POPT, LPROB, NITER,         &
                   RITER, SOPT, LDS, SDS, IFAIL
REAL (KIND=nag_wp) X(LDX,M), PROB(LPROB,NG), W(NG), G(NVAR,NG),             &
                   S(LDS,SDS,*), F(N,NG), TOL, LOGLIK
```

3 Description

A Normal (Gaussian) mixture model is a weighted sum of k group Normal densities given by,

$$p(x \mid w, \mu, \Sigma) = \sum_{j=1}^{k} w_j g\left(x \mid \mu_j, \Sigma_j\right), \quad x \in \mathbb{R}^p$$

where:

x is a p-dimensional object of interest;

w_j is the mixture weight for the jth group and $\sum_{j=1}^{k} w_j = 1$;

μ_j is a p-dimensional vector of means for the jth group;

Σ_j is the covariance structure for the jth group;

$g(\cdot)$ is the p-variate Normal density:

$$g\left(x \mid \mu_j, \Sigma_j\right) = \frac{1}{(2\pi)^{p/2}\left|\Sigma_j\right|^{1/2}} \exp\left[-\frac{1}{2}\left(x - \mu_j\right)\Sigma_j^{-1}\left(x - \mu_j\right)^{\mathrm{T}}\right].$$

Optionally, the (co)variance structure may be pooled (common to all groups) or calculated for each group, and may be full or diagonal.

4 References

Hartigan J A (1975) *Clustering Algorithms* Wiley

5 Arguments

1: N – INTEGER *Input*

On entry: n, the number of objects. There must be more objects than parameters in the model.

Constraints:

if $\text{SOPT} = 1$, $\text{N} > \text{NG} \times (\text{NVAR} \times \text{NVAR} + \text{NVAR})$;
if $\text{SOPT} = 2$, $\text{N} > \text{NVAR} \times (\text{NG} + \text{NVAR})$;
if $\text{SOPT} = 3$, $\text{N} > 2 \times \text{NG} \times \text{NVAR}$;
if $\text{SOPT} = 4$, $\text{N} > \text{NVAR} \times (\text{NG} + 1)$;
if $\text{SOPT} = 5$, $\text{N} > \text{NVAR} \times \text{NG} + 1$.

2: M – INTEGER *Input*

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

Constraint: $\text{M} \geq 1$.

3: X(LDX, M) – REAL (KIND=nag_wp) array *Input*

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

4: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G03GAF is called.

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

5: ISX(M) – INTEGER array *Input*

On entry: if $\text{NVAR} = \text{M}$ all available variables are included in the model and ISX is not referenced; otherwise the jth variable will be included in the analysis if $\text{ISX}(j) = 1$ and excluded if $\text{ISX}(j) = 0$, for $j = 1, 2, \ldots, \text{M}$.

Constraint: if $\text{NVAR} \neq \text{M}$, $\text{ISX}(j) = 1$ for NVAR values of j and $\text{ISX}(j) = 0$ for the remaining $\text{M} - \text{NVAR}$ values of j, for $j = 1, 2, \ldots, \text{M}$.

6: NVAR – INTEGER *Input*

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

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

7: NG – INTEGER *Input*

On entry: k, the number of groups in the mixture model.

Constraint: $\text{NG} \geq 1$.

8: POPT – INTEGER *Input*

On entry: if $\text{POPT} = 1$, the initial membership probabilities in PROB are set internally; otherwise these probabilities must be supplied.

9: PROB(LPROB, NG) – REAL (KIND=nag_wp) array *Input/Output*

On entry: if $\text{POPT} \neq 1$, $\text{PROB}(i, j)$ is the probability that the ith object belongs to the jth group. (These probabilities are normalised internally.)

On exit: $\text{PROB}(i, j)$ is the probability of membership of the ith object to the jth group for the fitted model.

10: LPROB – INTEGER *Input*

On entry: the first dimension of the array PROB as declared in the (sub)program from which G03GAF is called.

Constraint: $\text{LPROB} \geq \text{N}$.

11: NITER – INTEGER *Input/Output*

On entry: the maximum number of iterations.

Suggested value: 15

On exit: the number of completed iterations.

Constraint: NITER ≥ 1.

12: RITER – INTEGER *Input*

On entry: if RITER > 0, membership probabilities are rounded to 0.0 or 1.0 after the completion of every RITER iterations.

Suggested value: 5

13: W(NG) – REAL (KIND=nag_wp) array *Output*

On exit: w_j, the mixing probability for the jth group.

14: G(NVAR, NG) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{G}(i,j)$ gives the estimated mean of the ith variable in the jth group.

15: SOPT – INTEGER *Input*

On entry: determines the (co)variance structure:

SOPT $= 1$
: Groupwise covariance matrices.

SOPT $= 2$
: Pooled covariance matrix.

SOPT $= 3$
: Groupwise variances.

SOPT $= 4$
: Pooled variances.

SOPT $= 5$
: Overall variance.

Constraint: SOPT $= 1$, 2, 3, 4 or 5.

16: S(LDS, SDS, $*$) – REAL (KIND=nag_wp) array *Output*

Note: the last dimension of the array S must be at least NG if SOPT $= 1$, and at least 1 otherwise.

On exit: if SOPT $= 1$, $\mathrm{S}(i,j,k)$ gives the (i,j)th element of the kth group.

If SOPT $= 2$, $\mathrm{S}(i,j,1)$ gives the (i,j)th element of the pooled covariance.

If SOPT $= 3$, $\mathrm{S}(j,k,1)$ gives the jth variance in the kth group.

If SOPT $= 4$, $\mathrm{S}(j,1,1)$ gives the jth pooled variance.

If SOPT $= 5$, $\mathrm{S}(1,1,1)$ gives the overall variance.

17: LDS – INTEGER *Input*

On entry: the first dimension of the (co)variance structure S.

Constraints:

if SOPT $= 5$, LDS $= 1$;
otherwise LDS $=$ NVAR.

18: SDS – INTEGER *Input*

On entry: the second dimension of the (co)variance structure S.

Constraints:

if $\text{SOPT} = 1$ or 2, SDS must be at least NVAR;
if $\text{SOPT} = 3$, SDS must be at least NG;
if $\text{SOPT} = 4$ or 5, SDS must be at least 1.

19: F(N, NG) – REAL (KIND=nag_wp) array *Output*

On exit: $\text{F}(i,j)$ gives the p-variate Normal (Gaussian) density of the ith object in the jth group.

20: TOL – REAL (KIND=nag_wp) *Input*

On entry: iterations cease the first time an improvement in log-likelihood is less than TOL. If $\text{TOL} \leq 0$ a value of 10^{-3} is used.

21: LOGLIK – REAL (KIND=nag_wp) *Output*

On exit: the log-likelihood for the fitted mixture model.

22: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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: $\text{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

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

Errors or warnings detected by the routine:

$\text{IFAIL} = 1$

On entry, $\text{N} = \langle value \rangle$ and $p = \langle value \rangle$.
Constraint: $\text{N} > p$, the number of parameters, i.e., too few objects have been supplied for the model.

$\text{IFAIL} = 2$

On entry, $\text{M} = \langle value \rangle$.
Constraint: $\text{M} \geq 1$.

$\text{IFAIL} = 4$

On entry, $\text{LDX} = \langle value \rangle$ and $\text{N} = \langle value \rangle$.
Constraint: $\text{LDX} \geq \text{N}$.

$\text{IFAIL} = 5$

On entry, $\text{NVAR} = \langle value \rangle$ and $\text{M} = \langle value \rangle$.
Constraint: $1 \leq \text{NVAR} \leq \text{M}$.

IFAIL $= 6$

On entry, NVAR $\neq$ M and ISX is invalid.

IFAIL $= 7$

On entry, NG $= \langle value \rangle$.
Constraint: NG ≥ 1.

IFAIL $= 8$

On entry, POPT $\neq 1$ or 2.

IFAIL $= 9$

On entry, row $\langle value \rangle$ of supplied PROB does not sum to 1.

IFAIL $= 10$

On entry, LPROB $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: LPROB $\geq$ N.

IFAIL $= 11$

On entry, NITER $= \langle value \rangle$.
Constraint: NITER ≥ 1.

IFAIL $= 16$

On entry, SOPT < 1 or SOPT > 5.

IFAIL $= 18$

On entry, LDS $= \langle value \rangle$ was invalid.

IFAIL $= 19$

On entry, SDS $= \langle value \rangle$ was invalid.

IFAIL $= 44$

A covariance matrix is not positive definite, try a different initial allocation.

IFAIL $= 45$

An iteration cannot continue due to an empty group, try a different initial allocation.

IFAIL $= -99$

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

G03GAF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G03GAF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

This example fits a Gaussian mixture model with pooled covariance structure to New Haven schools test data, see Table 5.1 (p. 118) in Hartigan (1975).

10.1 Program Text

```
    Program g03gafe

!     G03GAF Example Program Text
!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g03gaf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: loglik, tol
      Integer                          :: i, ifail, ldprob, lds, ldx, m, n,    &
                                          ng, niter, nvar, popt, riter, sds,   &
                                          sopt
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: f(:,:), g(:,:), prob(:,:), s(:,:,:), &
                                          w(:), x(:,:)
      Integer, Allocatable             :: isx(:)
!     .. Executable Statements ..
      Write (nout,*) 'G03GAF Example Program Results'
      Write (nout,*)
      Flush (nout)

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

!     Problem size
      Read (nin,*) n, m, nvar

!     Number of groups
      Read (nin,*) ng

!     Scaling option
      Read (nin,*) sopt

!     Initial probabilities option
```

```
      Read (nin,*) popt

!     Maximum number of iterations
      Read (nin,*) niter

!     Leading dimensions
      ldx = n
      ldprob = n

      Select Case (sopt)
      Case (1)
        Allocate (s(nvar,nvar,ng))
        lds = nvar
        sds = nvar
      Case (2)
        Allocate (s(nvar,nvar,1))
        lds = nvar
        sds = nvar
      Case (3)
        Allocate (s(nvar,ng,1))
        lds = nvar
        sds = ng
      Case (4)
        Allocate (s(nvar,1,1))
        lds = nvar
        sds = 1
      Case Default
        Allocate (s(1,1,1))
        lds = 1
        sds = 1
      End Select

      Allocate (x(ldx,m),prob(ldprob,ng),g(nvar,ng),w(ng),isx(m),f(n,ng))

!     Data matrix X
      Read (nin,*)(x(i,1:m),i=1,n)

!     Included variables
      If (nvar/=m) Then
        Read (nin,*) isx(1:m)
      End If

!     Optionally read initial probabilities of group membership
      If (popt==2) Then
        Read (nin,*)(prob(i,1:ng),i=1,n)
      End If

      tol = 0.0E0_nag_wp
      riter = 5

      ifail = 0
      Call g03gaf(n,m,x,ldx,isx,nvar,ng,popt,prob,ldprob,niter,riter,w,g,sopt, &
        s,lds,sds,f,tol,loglik,ifail)

!     Results
      Write (nout,*)
      ifail = 0
      Call x04caf('g','n',1,ng,w,1,'Mixing proportions',ifail)

      Write (nout,*)
      ifail = 0
      Call x04caf('g','n',nvar,ng,g,nvar,'Group means',ifail)

      Write (nout,*)
      Select Case (sopt)
      Case (1)
        Do i = 1, ng
          ifail = 0
          Call x04caf('g','n',nvar,nvar,s(1,1,i),lds,                          &
            'Variance-covariance matrix',ifail)
        End Do
```

```
      Case (2)
        ifail = 0
        Call x04caf('g','n',nvar,nvar,s,lds,                                  &
          'Pooled Variance-covariance matrix',ifail)
      Case (3)
        ifail = 0
        Call x04caf('g','n',nvar,ng,s,lds,'Groupwise Variance',ifail)
      Case (4)
        ifail = 0
        Call x04caf('g','n',nvar,1,s,lds,'Pooled Variance',ifail)
      Case (5)
        ifail = 0
        Call x04caf('g','n',1,1,s,lds,'Overall Variance',ifail)
      End Select

      Write (nout,*)
      ifail = 0
      Call x04caf('g','n',n,ng,f,n,'Densities',ifail)

      Write (nout,*)
      ifail = 0
      Call x04caf('g','n',n,ng,prob,n,'Membership probabilities',ifail)

      Write (nout,*)
      Write (nout,'(1X,A,1X,I16)') 'No. iterations:', niter

      Write (nout,'(1X,A,1X,F16.4)') 'Log-likelihood:', loglik

      Deallocate (x,prob,g,s,w,isx,f)
    End Program g03gafe
```

10.2 Program Data

```
G03GAF Example Program Data
25 4 4          : N M IP
2               : NG
2               : SOPT
2               : POPT
15              : NITER
2.7 3.2 4.5 4.8
3.9 3.8 5.9 6.2
4.8 4.1 6.8 5.5
3.1 3.5 4.3 4.6
3.4 3.7 5.1 5.6
3.1 3.4 4.1 4.7
4.6 4.4 6.6 6.1
3.1 3.3 4.0 4.9
3.8 3.7 4.7 4.9
5.2 4.9 8.2 6.9
3.9 3.8 5.2 5.4
4.1 4.0 5.6 5.6
5.7 5.1 7.0 6.3
3.0 3.2 4.5 5.0
2.9 3.3 4.5 5.1
3.4 3.3 4.4 5.0
4.0 4.2 5.2 5.4
3.0 3.0 4.6 5.0
4.0 4.1 5.9 5.8
3.0 3.2 4.4 5.1
3.6 3.6 5.3 5.4
3.1 3.2 4.6 5.0
3.2 3.3 5.4 5.3
3.0 3.4 4.2 4.7
3.8 4.0 6.9 6.7 : X
1.0 0.0
1.0 0.0
1.0 0.0
1.0 0.0
1.0 0.0
1.0 0.0
```

```
1.0 0.0
1.0 0.0
1.0 0.0
1.0 0.0
1.0 0.0
1.0 0.0
0.0 1.0
0.0 1.0
0.0 1.0
0.0 1.0
0.0 1.0
0.0 1.0
0.0 1.0
0.0 1.0
0.0 1.0
0.0 1.0
0.0 1.0
0.0 1.0
0.0 1.0 : P
```

10.3 Program Results

```
 G03GAF Example Program Results


 Mixing proportions
          1       2
 1   0.4798  0.5202

 Group means
             1          2
 1      4.0041     3.3350
 2      3.9949     3.4434
 3      5.5894     4.9870
 4      5.4432     5.3602

 Pooled Variance-covariance matrix
             1          2          3          4
 1      0.4539     0.2891     0.6075     0.3413
 2      0.2891     0.2048     0.4101     0.2490
 3      0.6075     0.4101     1.0648     0.6011
 4      0.3413     0.2490     0.6011     0.3759

 Densities
                1            2
  1    2.5836E-01   1.1853E-02
  2    3.7065E-07   1.1241E-01
  3    5.3069E-03   1.8080E-06
  4    4.2461E-01   2.8584E-05
  5    5.0387E-02   1.1544E+00
  6    1.1260E+00   7.2224E-02
  7    2.0911E+00   2.1224E-02
  8    5.7856E-03   1.3227E+00
  9    1.1609E+00   2.9411E-02
 10    8.9826E-02   2.4260E-05
 11    3.0170E-01   1.0106E+00
 12    1.2930E+00   3.5422E-01
 13    2.8644E-02   6.7851E-07
 14    2.0759E-02   3.1690E+00
 15    7.6461E-02   1.5231E+00
 16    3.0279E-04   8.4017E-01
 17    5.6101E-01   4.6699E-05
 18    2.6573E-05   6.4442E-01
 19    2.1250E+00   5.1006E-02
 20    8.6822E-04   2.7626E+00
 21    1.9223E-01   2.3971E+00
 22    1.2469E-02   2.8179E+00
 23    1.8389E-02   5.3572E-01
 24    1.2409E+00   9.6489E-03
 25    2.1037E-05   4.8674E-02
```

```
Membership probabilities
                  1           2
  1    9.5018E-01   4.9823E-02
  2    3.3259E-06   1.0000E+00
  3    9.9961E-01   3.8664E-04
  4    9.9992E-01   7.9913E-05
  5    3.8999E-02   9.6100E-01
  6    9.3270E-01   6.7295E-02
  7    9.8881E-01   1.1190E-02
  8    4.1252E-03   9.9587E-01
  9    9.7252E-01   2.7479E-02
 10    9.9969E-01   3.0805E-04
 11    2.1722E-01   7.8278E-01
 12    7.6938E-01   2.3062E-01
 13    9.9997E-01   2.6937E-05
 14    6.1133E-03   9.9389E-01
 15    4.4189E-02   9.5581E-01
 16    3.5006E-04   9.9965E-01
 17    9.9990E-01   9.7029E-05
 18    4.0270E-05   9.9996E-01
 19    9.7380E-01   2.6202E-02
 20    3.0204E-04   9.9970E-01
 21    6.9471E-02   9.3053E-01
 22    4.1603E-03   9.9584E-01
 23    3.0839E-02   9.6916E-01
 24    9.9116E-01   8.8421E-03
 25    4.1534E-04   9.9958E-01

No. iterations:                14
Log-likelihood:          -29.6831
```