NAG Library Routine Document

G02GKF

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

G02GKF calculates the estimates of the arguments of a generalized linear model for given constraints from the singular value decomposition results.

2 Specification

```
SUBROUTINE G02GKF (IP, ICONST, V, LDV, C, LDC, B, S, SE, COV, WK, IFAIL)
INTEGER            IP, ICONST, LDV, LDC, IFAIL
REAL (KIND=nag_wp) V(LDV,IP+7), C(LDC,ICONST), B(IP), S, SE(IP),          &
                   COV(IP*(IP+1)/2),                                      &
                   WK(2*IP*IP+IP*ICONST+2*ICONST*ICONST+4*ICONST)
```

3 Description

G02GKF computes the estimates given a set of linear constraints for a generalized linear model which is not of full rank. It is intended for use after a call to G02GAF, G02GBF, G02GCF or G02GDF.

In the case of a model not of full rank the routines use a singular value decomposition to find the parameter estimates, $\hat{\beta}_{\text{svd}}$, and their variance-covariance matrix. Details of the SVD are made available in the form of the matrix P^*:

$$P^* = \begin{pmatrix} D^{-1}P_1^{\mathrm{T}} \\ P_0^{\mathrm{T}} \end{pmatrix}$$

as described by G02GAF, G02GBF, G02GCF and G02GDF. Alternative solutions can be formed by imposing constraints on the arguments. If there are p arguments and the rank of the model is k then $n_{\text{c}} = p - k$ constraints will have to be imposed to obtain a unique solution.

Let C be a p by n_{c} matrix of constraints, such that

$$C^{\mathrm{T}}\beta = 0,$$

then the new parameter estimates $\hat{\beta}_{\text{c}}$ are given by:

$$\begin{aligned} \hat{\beta}_{\text{c}} &= A\hat{\beta}_{\text{svd}} \\ &= \left(I - P_0(C^{\mathrm{T}}P_0)^{-1}\right)\hat{\beta}_{\text{svd}}, \qquad \text{where } I \text{ is the identity matrix,} \end{aligned}$$

and the variance-covariance matrix is given by

$$AP_1D^{-2}P_1^{\mathrm{T}}A^{\mathrm{T}}$$

provided $(C^{\mathrm{T}}P_0)^{-1}$ exists.

4 References

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

McCullagh P and Nelder J A (1983) *Generalized Linear Models* Chapman and Hall

Searle S R (1971) *Linear Models* Wiley

5 Arguments

1: IP – INTEGER *Input*

On entry: p, the number of terms in the linear model.

Constraint: IP ≥ 1.

2: ICONST – INTEGER *Input*

On entry: the number of constraints to be imposed on the arguments, n_{c}.

Constraint: $0 <$ ICONST $<$ IP.

3: V(LDV, IP + 7) – REAL (KIND=nag_wp) array *Input*

On entry: the array V as returned by G02GAF, G02GBF, G02GCF or G02GDF.

4: LDV – INTEGER *Input*

On entry: the first dimension of the array V as declared in the (sub)program from which G02GKF is called.

Constraint: LDV $\geq$ IP.

LDV should be as supplied to G02GAF, G02GBF, G02GCF or G02GDF

5: C(LDC, ICONST) – REAL (KIND=nag_wp) array *Input*

On entry: contains the ICONST constraints stored by column, i.e., the ith constraint is stored in the ith column of C.

6: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which G02GKF is called.

Constraint: LDC $\geq$ IP.

7: B(IP) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the parameter estimates computed by using the singular value decomposition, $\hat{\beta}_{\mathrm{svd}}$.

On exit: the parameter estimates of the arguments with the constraints imposed, $\hat{\beta}_{\mathrm{c}}$.

8: S – REAL (KIND=nag_wp) *Input*

On entry: the estimate of the scale argument.

For results from G02GAF and G02GDF then S is the scale argument for the model.

For results from G02GBF and G02GCF then S should be set to 1.0.

Constraint: S > 0.0.

9: SE(IP) – REAL (KIND=nag_wp) array *Output*

On exit: the standard error of the parameter estimates in B.

10: COV(IP $\times$ (IP + 1)/2) – REAL (KIND=nag_wp) array *Output*

On exit: the upper triangular part of the variance-covariance matrix of the IP parameter estimates given in B. They are stored packed by column, i.e., the covariance between the parameter estimate given in B(i) and the parameter estimate given in B(j), $j \geq i$, is stored in COV($(j \times (j-1)/2 + i)$).

11: WK($2 \times \text{IP} \times \text{IP} + \text{IP} \times \text{ICONST} + 2 \times \text{ICONST} \times \text{ICONST} + 4 \times \text{ICONST}$)
– REAL (KIND=nag_wp) array *Workspace*

Note: a simple upper bound for the size of the workspace is $5 \times \text{IP} \times \text{IP} + 4 \times \text{IP}$.

12: 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, $\text{IP} < 1$.
or $\text{ICONST} \geq \text{IP}$,
or $\text{ICONST} \leq 0$,
or $\text{LDV} < \text{IP}$,
or $\text{LDC} < \text{IP}$,
or $\text{S} \leq 0.0$.

$\text{IFAIL} = 2$

C does not give a model of full rank.

$\text{IFAIL} = -99$

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.

$\text{IFAIL} = -399$

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.

$\text{IFAIL} = -999$

Dynamic memory allocation failed.

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

7 Accuracy

It should be noted that due to rounding errors an argument that should be zero when the constraints have been imposed may be returned as a value of order ***machine precision***.

8 Parallelism and Performance

G02GKF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G02GKF 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

G02GKF is intended for use in situations in which dummy (0−1) variables have been used such as in the analysis of designed experiments when you do not wish to change the arguments of the model to give a full rank model. The routine is not intended for situations in which the relationships between the independent variables are only approximate.

10 Example

A loglinear model is fitted to a 3 by 5 contingency table by G02GCF. The model consists of terms for rows and columns. The table is

$$\begin{array}{rrrrr} 141 & 67 & 114 & 79 & 39 \\ 131 & 66 & 143 & 72 & 35 \\ 36 & 14 & 38 & 28 & 16 \end{array}.$$

The constraints that the sum of row effects and the sum of column effects are zero are then read in and the parameter estimates with these constraints imposed are computed by G02GKF and printed.

10.1 Program Text

```
    Program g02gkfe

!     G02GKF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g02gcf, g02gkf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: a, dev, eps, tol
      Integer                          :: i, iconst, idf, ifail, ip, iprint,   &
                                          irank, ldc, ldv, ldx, lwk, lwt, m,   &
                                          maxit, n
      Character (1)                    :: link, mean, offset, weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: b(:), c(:,:), cov(:), se(:), v(:,:), &
                                          wk(:), wt(:), x(:,:), y(:)
      Integer, Allocatable             :: isx(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: count
!     .. Executable Statements ..
      Write (nout,*) 'G02GKF Example Program Results'
      Write (nout,*)

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

!     Read in the problem size
```

```
      Read (nin,*) link, mean, offset, weight, n, m

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

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

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

!     Calculate IP
      ip = count(isx(1:m)>0)
      If (mean=='M' .Or. mean=='m') Then
        ip = ip + 1
      End If

!     Read in power for exponential link
      If (link=='E' .Or. link=='e') Then
        Read (nin,*) a
      End If

      ldv = n
      lwk = (ip*ip+3*ip+22)/2
      Allocate (b(ip),se(ip),cov(ip*(ip+1)/2),v(ldv,ip+7),wk(lwk))

!     Read in the offset
      If (offset=='Y' .Or. offset=='y') Then
        Read (nin,*) v(1:n,7)
      End If

!     Read in control parameters
      Read (nin,*) iprint, eps, tol, maxit

!     Fit generalized linear model with Poisson errors
      ifail = -1
      Call g02gcf(link,mean,offset,weight,n,x,ldx,m,isx,ip,y,wt,a,dev,idf,b,   &
        irank,se,cov,v,ldv,tol,maxit,iprint,eps,wk,ifail)
      If (ifail/=0) Then
        If (ifail<7) Then
          Go To 100
        End If
      End If

!     Display initial results
      Write (nout,99999) 'Deviance = ', dev
      Write (nout,99998) 'Degrees of freedom = ', idf
      Write (nout,*)

!     Calculate the number of constraints required
      iconst = ip - irank

!     Going to reallocate workspace, so deallocate it
      Deallocate (wk)
      lwk = 2*ip*ip + ip*iconst + 2*iconst*iconst + 4*iconst

      ldc = ip
      Allocate (c(ldc,iconst),wk(lwk))

!     Read in constraints
      Read (nin,*,Iostat=ifail)(c(i,1:iconst),i=1,ip)
      If (ifail/=0) Then
```

```
        Write (nout,99996)                                                &
          ' ** Insufficient constraints supplied, was expecting ', iconst
        Go To 100
      End If

!     Re-estimate the model given the constraints
      ifail = 0
      Call g02gkf(ip,iconst,v,ldv,c,ldc,b,1.0E0_nag_wp,se,cov,wk,ifail)

!     Display the constrained parameter estimates
      Write (nout,*) '      Estimate     Standard error'
      Write (nout,*)
      Write (nout,99997)(b(i),se(i),i=1,ip)

100   Continue

99999 Format (1X,A,E12.4)
99998 Format (1X,A,I2)
99997 Format (1X,2F14.4)
99996 Format (1X,A,I5)
    End Program g02gkfe
```

10.2 Program Data

```
G02GKF Example Program Data
'L' 'M' 'N' 'U' 15 8                  :: LINK,MEAN,OFFSET,WEIGHT,N,M
1.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 141.0
1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0  67.0
1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 114.0
1.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0  79.0
1.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0  39.0
0.0 1.0 0.0 1.0 0.0 0.0 0.0 0.0 131.0
0.0 1.0 0.0 0.0 1.0 0.0 0.0 0.0  66.0
0.0 1.0 0.0 0.0 0.0 1.0 0.0 0.0 143.0
0.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0  72.0
0.0 1.0 0.0 0.0 0.0 0.0 0.0 1.0  35.0
0.0 0.0 1.0 1.0 0.0 0.0 0.0 0.0  36.0
0.0 0.0 1.0 0.0 1.0 0.0 0.0 0.0  14.0
0.0 0.0 1.0 0.0 0.0 1.0 0.0 0.0  38.0
0.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0  28.0
0.0 0.0 1.0 0.0 0.0 0.0 0.0 1.0  16.0 :: End of X,Y
 1   1   1   1   1   1   1   1        :: ISX
0  1.0E-6  5.0E-5  0                  :: IPRINT,EPS,TOL,MAXIT
0.0 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                               :: End of constraints, C
```

10.3 Program Results

```
 G02GKF Example Program Results

 Deviance =   0.9038E+01
 Degrees of freedom =  8

       Estimate     Standard error

         3.9831         0.0396
         0.3961         0.0458
         0.4118         0.0457
        -0.8079         0.0622
```

```
 0.5112        0.0562
-0.2285        0.0727
 0.4680        0.0569
-0.0316        0.0675
-0.7191        0.0887
```