NAG Library Routine Document

G02GPF

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

G02GPF allows prediction from a generalized linear model fit via G02GAF, G02GBF, G02GCF or G02GDF.

2 Specification

```
SUBROUTINE G02GPF (ERRFN, LINK, MEAN, OFFSET, WEIGHT, N, X, LDX, M, ISX,    &
                   IP, T, OFF, WT, S, A, B, COV, VFOBS, ETA, SEETA,         &
                   PRED, SEPRED, IFAIL)

INTEGER            N, LDX, M, ISX(M), IP, IFAIL
REAL (KIND=nag_wp) X(LDX,*), T(*), OFF(*), WT(*), S, A, B(IP),              &
                   COV(IP*(IP+1)/2), ETA(N), SEETA(N), PRED(N),             &
                   SEPRED(N)
LOGICAL            VFOBS
CHARACTER(1)       ERRFN, LINK, MEAN, OFFSET, WEIGHT
```

3 Description

A generalized linear model consists of the following elements:

(i) A suitable distribution for the dependent variable y.

(ii) A linear model, with linear predictor $\eta = X\beta$, where X is a matrix of independent variables and β a column vector of p parameters.

(iii) A link function $g(.)$ between the expected value of y and the linear predictor, that is $E(y) = \mu = g(\eta)$.

In order to predict from a generalized linear model, that is estimate a value for the dependent variable, y, given a set of independent variables X, the matrix X must be supplied, along with values for the parameters β and their associated variance-covariance matrix, C. Suitable values for β and C are usually estimated by first fitting the prediction model to a training dataset with known responses, using for example G02GAF, G02GBF, G02GCF or G02GDF. The predicted variable, and its standard error can then be obtained from:

$$\hat{y} = g^{-1}(\eta), \quad \text{se}(\hat{y}) = \sqrt{\left(\frac{\delta g^{-1}(x)}{\delta x}\right)_{\eta}} \text{se}(\eta) + I_{\text{fobs}} \text{Var}(y)$$

where

$$\eta = o + X\beta, \quad \text{se}(\eta) = \text{diag}\sqrt{XCX^{\mathrm{T}}},$$

o is a vector of offsets and $I_{\text{fobs}} = 0$, if the variance of future observations is not taken into account, and 1 otherwise. Here diag A indicates the diagonal elements of matrix A.

If required, the variance for the ith future observation, $\text{Var}(y_i)$, can be calculated as:

$$\text{Var}(y_i) = \frac{\phi V(\theta)}{w_i}$$

where w_i is a weight, ϕ is the scale (or dispersion) parameter, and $V(\theta)$ is the variance function. Both the scale parameter and the variance function depend on the distribution used for the y, with:

Poisson $V(\theta) = \mu_i,\ \phi = 1$

binomial $V(\theta) = \frac{\mu_i(t_i - \mu_i)}{t_i},\ \phi = 1$

Normal $V(\theta) = 1$

gamma $V(\theta) = \mu_i^2$

In the cases of a Normal and gamma error structure, the scale parameter (ϕ), is supplied by you. This value is usually obtained from the routine used to fit the prediction model. In many cases, for a Normal error structure, $\phi = \hat{\sigma}^2$, i.e., the estimated variance.

4 References

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

5 Parameters

1: ERRFN – CHARACTER(1) *Input*

On entry: indicates the distribution used to model the dependent variable, y.

ERRFN = 'B'
The binomial distribution is used.

ERRFN = 'G'
The gamma distribution is used.

ERRFN = 'N'
The Normal (Gaussian) distribution is used.

ERRFN = 'P'
The Poisson distribution is used.

Constraint: ERRFN = 'B', 'G', 'N' or 'P'.

2: LINK – CHARACTER(1) *Input*

On entry: indicates which link function to be used.

LINK = 'C'
A complementary log-log link is used.

LINK = 'E'
An exponent link is used.

LINK = 'G'
A logistic link is used.

LINK = 'I'
An identity link is used.

LINK = 'L'
A log link is used.

LINK = 'P'
A probit link is used.

LINK = 'R'
A reciprocal link is used.

LINK = 'S'
A square root link is used.

Details on the functional form of the different links can be found in the G02 Chapter Introduction.

Constraints:

if ERRFN = 'B', LINK = 'C', 'G' or 'P';
otherwise LINK = 'E', 'I', 'L', 'R' or 'S'.

3: MEAN – CHARACTER(1) *Input*

On entry: indicates if a mean term is to be included.

MEAN = 'M'
A mean term, intercept, will be included in the model.

MEAN = 'Z'
The model will pass through the origin, zero-point.

Constraint: MEAN = 'M' or 'Z'.

4: OFFSET – CHARACTER(1) *Input*

On entry: indicates if an offset is required.

OFFSET = 'Y'
An offset must be supplied in OFF.

OFFSET = 'N'
OFF is not referenced.

Constraint: OFFSET = 'Y' or 'N'.

5: WEIGHT – CHARACTER(1) *Input*

On entry: if VFOBS = .TRUE. indicates if weights are used, otherwise WEIGHT is not referenced.

WEIGHT = 'U'
No weights are used.

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

Constraint: if VFOBS = .TRUE., WEIGHT = 'U' or 'W'.

6: N – INTEGER *Input*

On entry: n, the number of observations.

Constraint: N ≥ 1.

7: X(LDX, $*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array X must be at least M.

On entry: X(i,j) must contain the ith observation for the jth independent variable, for $i = 1, 2, \ldots,$ N and $j = 1, 2, \ldots,$ M.

8: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G02GPF is called.

Constraint: LDX $\geq$ N.

9: M – INTEGER *Input*

On entry: m, the total number of independent variables.

Constraint: M ≥ 1.

10: ISX(M) – INTEGER array *Input*

On entry: indicates which independent variables are to be included in the model.

If ISX(j) > 0, the jth independent variable is included in the regression model.

Constraints:

ISX(j) ≥ 0, for $i = 1, 2, \ldots, \mathrm{M}$;
if MEAN = 'M', exactly IP $-$ 1 values of ISX must be > 0;
if MEAN = 'Z', exactly IP values of ISX must be > 0.

11: IP – INTEGER *Input*

On entry: the number of independent variables in the model, including the mean or intercept if present.

Constraint: IP > 0.

12: T($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array must be at least N if ERRFN = 'B', and at least 1 otherwise.

On entry: if ERRFN = 'B', T(i) must contain the binomial denominator, t_i, for the ith observation.

Otherwise T is not referenced.

Constraint: if ERRFN = 'B', T(i) ≥ 0.0, for $i = 1, 2, \ldots, n$.

13: OFF($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array must be at least N if OFFSET = 'Y', and at least 1 otherwise.

On entry: if OFFSET = 'Y', OFF(i) must contain the offset o_i, for the ith observation.

Otherwise OFF is not referenced.

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

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

On entry: if WEIGHT = 'W' and VFOBS = .TRUE., WT(i) must contain the weight, w_i, for the ith observation.

If the variance of future observations is not included in the standard error of the predicted variable, WT is not referenced.

Constraint: if VFOBS = .TRUE. and WEIGHT = 'W', WT(i) $\geq 0.$, for $i = 1, 2, \ldots, i$.

15: S – REAL (KIND=nag_wp) *Input*

On entry: if ERRFN = 'N' or 'G' and VFOBS = .TRUE., the scale parameter, ϕ.

Otherwise S is not referenced and $\phi = 1$.

Constraint: if ERRFN = 'N' or 'G' and VFOBS = .TRUE., S > 0.0.

16: A – REAL (KIND=nag_wp) *Input*

On entry: if LINK = 'E', A must contain the power of the exponential.

If LINK $\neq$ 'E', A is not referenced.

Constraint: if LINK = 'E', A $\neq 0.0$.

17: B(IP) – REAL (KIND=nag_wp) array *Input*

On entry: the model parameters, β.

If MEAN = 'M', B(1) must contain the mean parameter and B($i+1$) the coefficient of the variable contained in the jth independent X, where ISX(j) is the ith positive value in the array ISX.

If MEAN = 'Z', B(i) must contain the coefficient of the variable contained in the jth independent X, where ISX(j) is the ith positive value in the array ISX.

18: COV(IP × (IP + 1)/2) – REAL (KIND=nag_wp) array *Input*

On entry: the upper triangular part of the variance-covariance matrix, C, of the model parameters. This matrix should be supplied packed by column, i.e., the covariance between parameters β_i and β_j, that is the values stored in B(i) and B(j), should be supplied in COV($j \times (j-1)/2 + i$), for $i = 1, 2, \ldots,$ IP and $j = i, \ldots,$ IP.

Constraint: the matrix represented in COV must be a valid variance-covariance matrix.

19: VFOBS – LOGICAL *Input*

On entry: if VFOBS = .TRUE., the variance of future observations is included in the standard error of the predicted variable (i.e., $I_{\text{fobs}} = 1$), otherwise $I_{\text{fobs}} = 0$.

20: ETA(N) – REAL (KIND=nag_wp) array *Output*

On exit: the linear predictor, η.

21: SEETA(N) – REAL (KIND=nag_wp) array *Output*

On exit: the standard error of the linear predictor, se(η).

22: PRED(N) – REAL (KIND=nag_wp) array *Output*

On exit: the predicted value, $\hat{y}$.

23: SEPRED(N) – REAL (KIND=nag_wp) array *Output*

On exit: the standard error of the predicted value, se($\hat{y}$). If PRED(i) could not be calculated, then G02GPF returns IFAIL = 22, and SEPRED(i) is set to -99.0.

24: 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, because for this routine the values of the output parameters may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **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).

Note: G02GPF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

IFAIL = 1

On entry, ERRFN is invalid: ERRFN = ⟨*value*⟩.

IFAIL $= 2$

On entry, ERRFN and LINK combination is invalid: ERRFN $= \langle value \rangle$, LINK $= \langle value \rangle$.

On entry, LINK is invalid: LINK $= \langle value \rangle$.

IFAIL $= 3$

On entry, MEAN is invalid: MEAN $= \langle value \rangle$.

IFAIL $= 4$

On entry, OFFSET is invalid: OFFSET $= \langle value \rangle$.

IFAIL $= 5$

On entry, WEIGHT is invalid: WEIGHT $= \langle value \rangle$.

IFAIL $= 6$

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

IFAIL $= 8$

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

IFAIL $= 9$

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

IFAIL $= 10$

On entry, ISX not consistent with IP: $\langle value \rangle$ values > 0, expected $\langle value \rangle$.

IFAIL $= 11$

On entry, IP $= \langle value \rangle$.
Constraint: IP > 0.

IFAIL $= 12$

On entry, $i = \langle value \rangle$ and T$(i) = \langle value \rangle$.
Constraint: T$(i) \geq 0.0$, for all i.

IFAIL $= 14$

On entry, $i = \langle value \rangle$ and WT$(i) = \langle value \rangle$.
Constraint: WT$(i) \geq 0.0$, for all i.

IFAIL $= 15$

On entry, S $= \langle value \rangle$.
Constraint: S > 0.0.

IFAIL $= 16$

On entry, A $= 0.0$.

IFAIL $= 18$

On entry, COV$(i) < 0.0$ for at least one diagonal element: $i = \langle value \rangle$, COV$(i) = \langle value \rangle$.

IFAIL $= 22$

At least one predicted value could not be calculated as required. SEPRED is set to -99.0 for affected predicted values.

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

Not applicable.

8 Parallelism and Performance

G02GPF is not threaded by NAG in any implementation.

G02GPF 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

The model

$$y = \frac{1}{\beta_1 + \beta_2 x} + \epsilon$$

is fitted to a training dataset with five observations. The resulting model is then used to predict the response for two new observations.

10.1 Program Text

```
    Program g02gpfe

!     G02GPF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g02gaf, g02gpf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
```

```
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: a, eps, rss, s, tol
      Integer                          :: i, idf, ifail, ip, iprint, irank,    &
                                          ldv, ldx, loff, lt, lwk, lwt, m,     &
                                          maxit, n, tldx
      Logical                          :: vfobs
      Character (1)                    :: errfn, link, mean, offset, weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: b(:), cov(:), eta(:), off(:),        &
                                          pred(:), se(:), seeta(:), sepred(:), &
                                          t(:), twt(:), tx(:,:), ty(:),        &
                                          v(:,:), wk(:), wt(:), x(:,:), y(:)
      Integer, Allocatable             :: isx(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: count
!     .. Executable Statements ..
      Write (nout,*) 'G02GPF Example Program Results'
      Write (nout,*)

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

!     Read in training data for model that will be used for prediction
      Read (nin,*) link, mean, offset, weight, n, m, s

      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      tldx = n
      Allocate (tx(tldx,m),ty(n),twt(lwt),isx(m))

!     Read in data
      If (lwt>0) Then
        Read (nin,*)(tx(i,1:m),ty(i),twt(i),i=1,n)
      Else
        Read (nin,*)(tx(i,1:m),ty(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

!     Call routine to fit generalized linear model, with Normal errors,
!     to training data
```

```
      ifail = -1
      Call g02gaf(link,mean,offset,weight,n,tx,tldx,m,isx,ip,ty,twt,s,a,rss, &
        idf,b,irank,se,cov,v,ldv,tol,maxit,iprint,eps,wk,ifail)
      If (ifail/=0) Then
        If (ifail<6) Then
          Go To 100
        End If
      End If

!     Display parameter estimates for training data
      Write (nout,99999) 'Residual sum of squares = ', rss
      Write (nout,99998) 'Degrees of freedom      = ', idf
      Write (nout,*)
      Write (nout,*) '      Estimate     Standard error'
      Write (nout,*)
      Write (nout,99997)(b(i),se(i),i=1,ip)

!     Skip second lot of headings in data file
      Read (nin,*)

!     Read in size of prediction data
      Read (nin,*) n, vfobs, offset, weight

!     Used G02GAF to fit training model, so error structure is normal is
!     do not require T array
      errfn = 'N'
      lt = 0

      ldx = n
      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      If (offset=='Y' .Or. offset=='y') Then
        loff = n
      Else
        loff = 0
      End If
      Allocate (x(ldx,m),y(n),wt(lwt),off(loff),eta(n),seeta(n),pred(n), &
        sepred(n),t(lt))

!     Read in predication 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 offset for the prediction data
      If (offset=='Y' .Or. offset=='y') Then
        Read (nin,*) off(1:n)
      End If

!     Call prediction routine
      ifail = -1
      Call g02gpf(errfn,link,mean,offset,weight,n,x,ldx,m,isx,ip,t,off,wt,s,a, &
        b,cov,vfobs,eta,seeta,pred,sepred,ifail)
      If (ifail/=0) Then
        Write (nout,99995) ifail
        If (ifail/=22) Then
          Go To 100
        End If
      End If

!     Display predicted values
      Write (nout,*)
      Write (nout,*) '  I      ETA         SE(ETA)      Predicted  ', &
        ' SE(Predicted)'
      Write (nout,*)
      Write (nout,99996)(i,eta(i),seeta(i),pred(i),sepred(i),i=1,n)
```

```
100   Continue

99999 Format (1X,A,E12.4)
99998 Format (1X,A,I0)
99997 Format (1X,2F14.4)
99996 Format (1X,I3,')',F10.5,3F14.5)
99995 Format (1X,A,I5)
    End Program g02gpfe
```

10.2 Program Data

```
G02GPF Example Program Data
Training Data
'R' 'M' 'N' 'U' 5 1  0.0   :: LINK,MEAN,OFFSET,WEIGHT,N,M,S
1.0 25.0
2.0 10.0
3.0  6.0
4.0  4.0
5.0  3.0                   :: End of X,Y
1                          :: ISX
0  1.0E-6  5.0E-5  0       :: IPRINT,EPS,TOL,MAXIT
Prediction Data
2 .TRUE. 'N' 'U'           :: N,VFOBS,OFFSET,WEIGHT
32.0
18.0                       :: End of X
```

10.3 Program Results

```
 G02GPF Example Program Results

 Residual sum of squares =   0.3872E+00
 Degrees of freedom      = 3

       Estimate     Standard error

        -0.0239        0.0028
         0.0638        0.0026

   I      ETA         SE(ETA)      Predicted   SE(Predicted)

   1)   2.01807       0.08168       0.49552       0.35981
   2)   1.12472       0.04476       0.88911       0.36098
```