NAG Library Routine Document

G10ABF

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

G10ABF fits a cubic smoothing spline for a given smoothing parameter.

2 Specification

```
SUBROUTINE G10ABF (MODE, WEIGHT, N, X, Y, WT, RHO, YHAT, C, LDC, RSS,      &
                   DF, RES, H, COMM, IFAIL)

INTEGER            N, LDC, IFAIL
REAL (KIND=nag_wp) X(N), Y(N), WT(*), RHO, YHAT(N), C(LDC,3), RSS, DF,     &
                   RES(N), H(N), COMM(9*N+14)
CHARACTER(1)       MODE, WEIGHT
```

3 Description

G10ABF fits a cubic smoothing spline to a set of n observations (x_i, y_i), for $i = 1, 2, \ldots, n$. The spline provides a flexible smooth function for situations in which a simple polynomial or nonlinear regression model is unsuitable.

Cubic smoothing splines arise as the unique real-valued solution function f, with absolutely continuous first derivative and squared-integrable second derivative, which minimizes:

$$\sum_{i=1}^{n} w_i (y_i - f(x_i))^2 + \rho \int_{-\infty}^{\infty} (f''(x))^2 \, dx,$$

where w_i is the (optional) weight for the ith observation and ρ is the smoothing parameter. This criterion consists of two parts: the first measures the fit of the curve, and the second the smoothness of the curve. The value of the smoothing parameter ρ weights these two aspects; larger values of ρ give a smoother fitted curve but, in general, a poorer fit. For details of how the cubic spline can be estimated see Hutchinson and de Hoog (1985) and Reinsch (1967).

The fitted values, $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)^{\mathrm{T}}$, and weighted residuals, r_i, can be written as

$$\hat{y} = Hy \quad \text{and} \quad r_i = \sqrt{w_i}(y_i - \hat{y}_i)$$

for a matrix H. The residual degrees of freedom for the spline is trace$(I - H)$ and the diagonal elements of H, h_{ii}, are the leverages.

The parameter ρ can be chosen in a number of ways. The fit can be inspected for a number of different values of ρ. Alternatively the degrees of freedom for the spline, which determines the value of ρ, can be specified, or the (generalized) cross-validation can be minimized to give ρ; see G10ACF for further details.

G10ABF requires the x_i to be strictly increasing. If two or more observations have the same x_i-value then they should be replaced by a single observation with y_i equal to the (weighted) mean of the y values and weight, w_i, equal to the sum of the weights. This operation can be performed by G10ZAF.

The computation is split into three phases.

(i) Compute matrices needed to fit spline.

(ii) Fit spline for a given value of ρ.

(iii) Compute spline coefficients.

When fitting the spline for several different values of ρ, phase (i) need only be carried out once and then phase (ii) repeated for different values of ρ. If the spline is being fitted as part of an iterative weighted least squares procedure phases (i) and (ii) have to be repeated for each set of weights. In either case, phase (iii) will often only have to be performed after the final fit has been computed.

The algorithm is based on Hutchinson (1986).

4 References

Hastie T J and Tibshirani R J (1990) *Generalized Additive Models* Chapman and Hall

Hutchinson M F (1986) Algorithm 642: A fast procedure for calculating minimum cross-validation cubic smoothing splines *ACM Trans. Math. Software* **12** 150–153

Hutchinson M F and de Hoog F R (1985) Smoothing noisy data with spline functions *Numer. Math.* **47** 99–106

Reinsch C H (1967) Smoothing by spline functions *Numer. Math.* **10** 177–183

5 Arguments

1: MODE – CHARACTER(1) *Input*

On entry: indicates in which mode the routine is to be used.

MODE = 'P'
Initialization and fitting is performed. This partial fit can be used in an iterative weighted least squares context where the weights are changing at each call to G10ABF or when the coefficients are not required.

MODE = 'Q'
Fitting only is performed. Initialization must have been performed previously by a call to G10ABF with MODE = 'P'. This quick fit may be called repeatedly with different values of RHO without re-initialization.

MODE = 'F'
Initialization and full fitting is performed and the function coefficients are calculated.

Constraint: MODE = 'P', 'Q' or 'F'.

2: WEIGHT – CHARACTER(1) *Input*

On entry: indicates whether user-defined weights are to be used.

WEIGHT = 'W'
User-defined weights should be supplied in WT.

WEIGHT = 'U'
The data is treated as unweighted.

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

3: N – INTEGER *Input*

On entry: n, the number of distinct observations.

Constraint: N ≥ 3.

4: X(N) – REAL (KIND=nag_wp) array *Input*

On entry: the distinct and ordered values x_i, for $i = 1, 2, \ldots, n$.

Constraint: $\mathrm{X}(i) < \mathrm{X}(i+1)$, for $i = 1, 2, \ldots, n-1$.

5: Y(N) – REAL (KIND=nag_wp) array *Input*

On entry: the values y_i, for $i = 1, 2, \ldots, n$.

6: WT(∗) – REAL (KIND=nag_wp) array *Input*

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

On entry: if WEIGHT = 'W', WT must contain the n weights. Otherwise WT is not referenced and unit weights are assumed.

Constraint: if WEIGHT = 'W', WT$(i) > 0.0$, for $i = 1, 2, \ldots, n$.

7: RHO – REAL (KIND=nag_wp) *Input*

On entry: ρ, the smoothing parameter.

Constraint: RHO ≥ 0.0.

8: YHAT(N) – REAL (KIND=nag_wp) array *Output*

On exit: the fitted values, $\hat{y}_i$, for $i = 1, 2, \ldots, n$.

9: C(LDC, 3) – REAL (KIND=nag_wp) array *Input/Output*

On entry: if MODE = 'Q', C must be unaltered from the previous call to G10ABF with MODE = 'P'. Otherwise C need not be set.

On exit: if MODE = 'F', C contains the spline coefficients. More precisely, the value of the spline at t is given by $((\mathrm{C}(i,3) \times d + \mathrm{C}(i,2)) \times d + \mathrm{C}(i,1)) \times d + \hat{y}_i$, where $x_i \leq t < x_{i+1}$ and $d = t - x_i$.

If MODE = 'P' or 'Q', C contains information that will be used in a subsequent call to G10ABF with MODE = 'Q'.

10: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which G10ABF is called.

Constraint: LDC $\geq$ N $-$ 1.

11: RSS – REAL (KIND=nag_wp) *Output*

On exit: the (weighted) residual sum of squares.

12: DF – REAL (KIND=nag_wp) *Output*

On exit: the residual degrees of freedom.

13: RES(N) – REAL (KIND=nag_wp) array *Output*

On exit: the (weighted) residuals, r_i, for $i = 1, 2, \ldots, n$.

14: H(N) – REAL (KIND=nag_wp) array *Output*

On exit: the leverages, h_{ii}, for $i = 1, 2, \ldots, n$.

15: COMM($9 \times$ N $+ 14$) – REAL (KIND=nag_wp) array *Communication Array*

On entry: if MODE = 'Q', COMM must be unaltered from the previous call to G10ABF with MODE = 'P'. Otherwise COMM need not be set.

On exit: if MODE = 'P' or 'Q', COMM contains information that will be used in a subsequent call to G10ABF with MODE = 'Q'.

16: 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: 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, N < 3,
or LDC $<$ N $- 1$,
or RHO < 0.0,
or MODE $\neq$ 'Q', 'P' or 'F',
or WEIGHT $\neq$ 'W' or 'U'.

IFAIL $= 2$

On entry, WEIGHT $=$ 'W' and at least one element of WT ≤ 0.0.

IFAIL $= 3$

On entry, X(i) $\geq$ X($i + 1$), for some i, $i = 1, 2, \ldots, n - 1$.

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

Accuracy depends on the value of ρ and the position of the x values. The values of $x_i - x_{i-1}$ and w_i are scaled and ρ is transformed to avoid underflow and overflow problems.

8 Parallelism and Performance

G10ABF is not threaded in any implementation.

9 Further Comments

The time taken by G10ABF is of order n.

Regression splines with a small $(< n)$ number of knots can be fitted by E02BAF and E02BEF.

10 Example

The data, given by Hastie and Tibshirani (1990), is the age, x_i, and C-peptide concentration (pmol/ml), y_i, from a study of the factors affecting insulin-dependent diabetes mellitus in children. The data is input, reduced to a strictly ordered set by G10ZAF and a series of splines fit using a range of values for the smoothing parameter, ρ.

10.1 Program Text

```
    Program g10abfe

!     G10ABF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g10abf, g10zaf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, j, lcomm, ldc, lwt, n,     &
                                          nord, nrho
      Character (1)                    :: mode, weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: c(:,:), comm(:), df(:), h(:),        &
                                          res(:), rho(:), rss(:), wt(:),       &
                                          wwt(:), x(:), xord(:), y(:),         &
                                          yhat(:,:), yord(:)
      Integer, Allocatable             :: iwrk(:)
!     .. Executable Statements ..
      Write (nout,*) ' G10ABF Example Program Results'
      Write (nout,*)

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

!     Read in problem size and control parameters
      Read (nin,*) n, weight, nrho

      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      lcomm = 9*n + 14
      ldc = n - 1
      Allocate (x(n),y(n),wt(lwt),xord(n),yord(n),wwt(n),yhat(n,nrho),         &
        c(ldc,3),res(n),h(n),comm(lcomm),iwrk(n),rho(nrho),rss(nrho),df(nrho))

!     Read in the smoothing parameters
      Read (nin,*) rho(1:nrho)

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

!     Reorder data into increasing X and remove tied observations, weighting
!     accordingly
      ifail = 0
      Call g10zaf(weight,n,x,y,wt,nord,xord,yord,wwt,rss(1),iwrk,ifail)

!     Fit cubic spline the first time
!     NB: These are weighted as G10ZAF creates weights
      ifail = 0
```

```
      mode = 'P'
      Call g10abf(mode,'W',nord,xord,yord,wwt,rho(1),yhat(1,1),c,ldc,rss(1),   &
        df(1),res,h,comm,ifail)

!     Fit cubic spline the remaining NRHO - 1 times
      mode = 'Q'
      Do i = 2, nrho
        ifail = 0
        Call g10abf(mode,'W',nord,xord,yord,wwt,rho(i),yhat(1,i),c,ldc,rss(i), &
          df(i),res,h,comm,ifail)
      End Do

!     Display results
      Write (nout,99999) 'Smoothing coefficient (rho) = ', rho(1:nrho)
      Write (nout,99998) 'Residual sum of squares     = ', rss(1:nrho)
      Write (nout,99998) 'Degrees of freedom          = ', df(1:nrho)
      Write (nout,*)
      Write (nout,*) '    X       Y                            Fitted Values'
      Do i = 1, nord
        Write (nout,99997) xord(i), yord(i), (yhat(i,j),j=1,nrho)
      End Do

99999 Format (1X,A,10(2X,F8.2))
99998 Format (1X,A,10(F10.3))
99997 Format (1X,2F8.4,14X,10(2X,F8.4))
    End Program g10abfe
```

10.2 Program Data

```
G10ABF Example Program Data
43  'U' 3      :: N,MODE,WEIGHT,NRHO
1.0 10.0 100.0 :: RHO
 5.2  4.8
 8.8  4.1
10.5  5.2
10.6  5.5
10.4  5.0
 1.8  3.4
12.7  3.4
15.6  4.9
 5.8  5.6
 1.9  3.7
 2.2  3.9
 4.8  4.5
 7.9  4.8
 5.2  4.9
 0.9  3.0
11.8  4.6
 7.9  4.8
11.5  5.5
10.6  4.5
 8.5  5.3
11.1  4.7
12.8  6.6
11.3  5.1
 1.0  3.9
14.5  5.7
11.9  5.1
 8.1  5.2
13.8  3.7
15.5  4.9
 9.8  4.8
11.0  4.4
12.4  5.2
11.1  5.1
 5.1  4.6
 4.8  3.9
 4.2  5.1
 6.9  5.1
13.2  6.0
```

```
 9.9  4.9
12.5  4.1
13.2  4.6
 8.9  4.9
10.8  5.1      :: End of X,Y
```

10.3 Program Results

```
  G10ABF Example Program Results

 Smoothing coefficient (rho) =       1.00     10.00    100.00
 Residual sum of squares     =      9.118    11.288    11.881
 Degrees of freedom          =     22.505    27.785    31.191

    X       Y                          Fitted Values
  0.9000  3.0000                 3.3784    3.3674    3.3699
  1.0000  3.9000                 3.4173    3.4008    3.4063
  1.8000  3.4000                 3.6144    3.6642    3.6973
  1.9000  3.7000                 3.6639    3.7016    3.7341
  2.2000  3.9000                 3.8607    3.8214    3.8449
  4.2000  5.1000                 4.7441    4.5265    4.5194
  4.8000  4.2000                 4.4914    4.6471    4.6746
  5.1000  4.6000                 4.6708    4.7561    4.7470
  5.2000  4.8500                 4.7704    4.7993    4.7702
  5.8000  5.6000                 5.3426    5.0458    4.8879
  6.9000  5.1000                 5.1728    5.1204    4.9753
  7.9000  4.8000                 4.9467    4.9590    4.9537
  8.1000  5.2000                 4.9556    4.9262    4.9452
  8.5000  5.3000                 4.8742    4.8595    4.9276
  8.8000  4.1000                 4.7305    4.8172    4.9168
  8.9000  4.9000                 4.7024    4.8095    4.9143
  9.8000  4.8000                 4.8394    4.8676    4.9170
  9.9000  4.9000                 4.8746    4.8818    4.9191
 10.4000  5.0000                 4.9971    4.9445    4.9303
 10.5000  5.2000                 4.9997    4.9521    4.9321
 10.6000  5.0000                 4.9921    4.9572    4.9335
 10.8000  5.1000                 4.9603    4.9613    4.9354
 11.0000  4.4000                 4.9396    4.9614    4.9363
 11.1000  4.9000                 4.9494    4.9618    4.9366
 11.3000  5.1000                 4.9926    4.9623    4.9366
 11.5000  5.5000                 5.0116    4.9568    4.9355
 11.8000  4.6000                 4.9372    4.9338    4.9315
 11.9000  5.1000                 4.9042    4.9251    4.9300
 12.4000  5.2000                 4.7929    4.8943    4.9240
 12.5000  4.1000                 4.8042    4.8944    4.9237
 12.7000  3.4000                 4.9020    4.9051    4.9244
 12.8000  6.6000                 4.9752    4.9138    4.9252
 13.2000  5.3000                 5.0173    4.9239    4.9276
 13.8000  3.7000                 4.6164    4.8930    4.9304
 14.5000  5.7000                 5.1883    4.9938    4.9518
 15.5000  4.9000                 4.9854    4.9773    4.9687
 15.6000  4.9000                 4.9167    4.9682    4.9697
```

Example Program
Cubic Smoothing Spline
Study of the factors affecting insulin-dependent diabetes mellitus in children
Hastie and Tibshirani (1990)