G10ACF – NAG Fortran Library Routine Document

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

G10ACF estimates the values of the smoothing parameter and fits a cubic smoothing spline to a set of data.

2 Specification

```
      SUBROUTINE G10ACF(METHOD, WEIGHT, N, X, Y, WT, YHAT, C, LDC, RSS,
     1                  DF, RES, H, CRIT, RHO, U, TOL, MAXCAL, WK, IFAIL)
      INTEGER           N, LDC, MAXCAL, IFAIL
      real              X(N), Y(N), WT(*), YHAT(N), C(LDC,3), RSS, DF,
     1                  RES(N), H(N), CRIT, RHO, U, TOL, WK(7*(N+2))
      CHARACTER*1       METHOD, WEIGHT
```

3 Description

For a set of n observations (x_i, y_i), $i = 1, 2, \ldots, n$, the spline provides a flexible smooth function for situations in which a simple polynomial or non-linear regression model is not suitable.

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

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

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 fitted see Hutchinson and de Hoog [3] and Reinsch [4].

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

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

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

The parameter ρ can be estimated in a number of ways.

(1) The degrees of freedom for the spline can be specified, i.e., find ρ such that trace$(H) = \nu_0$ for given ν_0.

(2) Minimise the cross-validation (CV), i.e., find ρ such that the CV is minimised, where

$$\mathrm{CV} = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} \left[\frac{r_i}{1 - h_{ii}}\right]^2.$$

(3) Minimise the generalised cross-validation (GCV), i.e., find ρ such that the GCV is minimised, where

$$\mathrm{GCV} = \frac{n^2}{\sum_{i=1}^{n} w_i} \left[\frac{\sum_{i=1}^{n} r_i^2}{\left(\sum_{i=1}^{n} (1 - h_{ii})\right)^2}\right].$$

G10ACF requires the x_i's 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 algorithm is based on Hutchinson [2]. C05AZF is used to solve for ρ given ν_0 and the method of E04ABF is used to minimise the GCV or CV.

4 References

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

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

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

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

5 Parameters

1: METHOD — CHARACTER*1 *Input*

On entry: indicates whether the smoothing parameter is to be found by minimization of the CV or GCV functions, or by finding the smoothing parameter corresponding to a specified degrees of freedom value.

If METHOD = 'C' cross-validation is used.

If METHOD = 'D' the degrees of freedom are specified.

If METHOD = 'G' generalized cross-validation is used.

Constraint: METHOD = 'C', 'D' or 'G'.

2: WEIGHT — CHARACTER*1 *Input*

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

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

If WEIGHT = 'U' the data is treated as unweighted.

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

3: N — INTEGER *Input*

On entry: the number of observations, n.

Constraint: N ≥ 3.

4: X(N) — ***real*** array *Input*

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

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

5: Y(N) — ***real*** array *Input*

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

6: WT($*$) — ***real*** array *Input*

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

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

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

7: YHAT(N) — ***real*** array *Output*

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

8: C(LDC,3) — ***real*** array *Output*

On exit: the spline coefficients. More precisely, the value of the spline approximation 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 $x_i \le t < x_{i+1}$ and $d = t - x_i$.

9: LDC — INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which G10ACF is called.

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

10: RSS — ***real*** *Output*

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

11: DF — ***real*** *Output*

On exit: the residual degrees of freedom. If METHOD = 'D' this will be $n -$ CRIT to the required accuracy.

12: RES(N) — ***real*** array *Output*

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

13: H(N) — ***real*** array *Output*

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

14: CRIT — ***real*** *Input/Output*

On entry: if METHOD = 'D', the required degrees of freedom for the spline. If METHOD = 'C' or 'G', CRIT need not be set.

Constraint: 2.0 < CRIT $\leq$ N.

On exit: if METHOD = 'C', the value of the cross-validation, or if METHOD = 'G' the value of the generalized cross-validation function, evaluated at the value of ρ returned in RHO.

15: RHO — ***real*** *Output*

On exit: the smoothing parameter, ρ.

16: U — ***real*** *Input*

On entry: the upper bound on the smoothing parameter. See Section 8 for details on how this parameter is used.

Constraint: U > TOL.

Suggested value: U = 1000.0.

17: TOL — ***real*** *Input*

On entry: the accuracy to which the smoothing parameter RHO is required. TOL should be preferably not much less than $\sqrt{\epsilon}$, where ϵ is the ***machine precision***.

Constraint: TOL $\geq$ ***machine precision***.

18: MAXCAL — INTEGER *Input*

On entry: the maximum number of spline evaluations to be used in finding the value of ρ.

Constraint: MAXCAL $\geq$ 3.

Suggested value: MAXCAL = 30.

19: WK(7∗(N+2)) — ***real*** array *Workspace*

20: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (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 detected by the routine:

IFAIL = 1

On entry, N < 3,
or LDC < N − 1,
or METHOD is not 'C', 'G' or 'D',
or WEIGHT is not 'W' or 'U',
or METHOD = 'D' and CRIT ≤ 2.0,
or METHOD = 'D' and CRIT > N,
or TOL < ***machine precision***,
or U ≤ TOL,
or MAXCAL < 3.

IFAIL = 2

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

IFAIL = 3

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

IFAIL = 4

METHOD = 'D' and the required value of ρ for specified degrees of freedom > U. Try a larger value of U, see Section 8.

IFAIL = 5

METHOD = 'D' and the accuracy given by TOL cannot be achieved. Try increasing the value of TOL.

IFAIL = 6

A solution to the accuracy given by TOL has not been achieved in MAXCAL iterations. Try increasing the value of TOL and/or MAXCAL.

IFAIL = 7

METHOD = 'C' or 'G' and the optimal value of ρ > U. Try a larger value of U, see Section 8.

7 Accuracy

When minimising the cross-validation or generalised cross-validation, the error in the estimate of ρ should be within $\pm 3(\mathrm{TOL} \times \mathrm{RHO} + \mathrm{TOL})$. When finding ρ for a fixed number of degrees of freedom the error in the estimate of ρ should be within $\pm 2 \times \mathrm{TOL} \times \max(1, \mathrm{RHO})$.

Given the value of ρ, the accuracy of the fitted spline 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 Further Comments

The time to fit the spline for a given value of ρ is of order n.

When finding the value of ρ that gives the required degrees of freedom, the algorithm examines the interval 0.0 to U. For small degrees of freedom the value of ρ can be large, as in the theoretical case of two degrees of freedom when the spline reduces to a straight line and ρ is infinite. If the CV or GCV is to be minimised then the algorithm searches for the minimum value in the interval 0.0 to U. If the function is decreasing in that range then the boundary value of U will be returned. In either case, the larger the value of U the more likely is the interval to contain the required solution, but the process will be less efficient.

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

9 Example

The data, given by Hastie and Tibshirani [1], 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 spline with 5 degrees of freedom is fitted by G10ACF. The fitted values and residuals are printed.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     G10ACF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDC
      PARAMETER        (NMAX=50,LDC=49)
*     .. Local Scalars ..
      real             CRIT, DF, RHO, RSS, TOL, U
      INTEGER          I, IFAIL, MAXCAL, N, NORD
      CHARACTER        METHOD, WEIGHT
*     .. Local Arrays ..
      real             C(LDC,3), H(NMAX), RES(NMAX), WK(7*(NMAX+2)),
     +                 WT(NMAX), WWT(NMAX), X(NMAX), XORD(NMAX),
     +                 Y(NMAX), YHAT(NMAX), YORD(NMAX)
      INTEGER          IWRK(NMAX)
*     .. External Subroutines ..
      EXTERNAL         G10ACF, G10ZAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G10ACF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
         READ (NIN,*) METHOD, WEIGHT
         IF (WEIGHT.EQ.'U' .OR. WEIGHT.EQ.'u') THEN
            READ (NIN,*) (X(I),Y(I),I=1,N)
         ELSE
            READ (NIN,*) (X(I),Y(I),WT(I),I=1,N)
         END IF
         READ (NIN,*) U, TOL, MAXCAL, CRIT
         IFAIL = 0
*
```

```
         IFAIL = 0
*
*        Sort data, removing ties and weighting accordingly
*
         CALL G10ZAF(WEIGHT,N,X,Y,WT,NORD,XORD,YORD,WWT,RSS,IWRK,IFAIL)
*
*        Fit cubic spline
*
         CALL G10ACF(METHOD,'W',NORD,XORD,YORD,WWT,YHAT,C,LDC,RSS,DF,
     +               RES,H,CRIT,RHO,U,TOL,MAXCAL,WK,IFAIL)
*
*        Print results
*
         WRITE (NOUT,*)
         WRITE (NOUT,99999) RSS
         WRITE (NOUT,99998) DF
         WRITE (NOUT,99997) RHO
         WRITE (NOUT,99996)
         DO 20 I = 1, NORD
            WRITE (NOUT,99995) I, XORD(I), YORD(I), YHAT(I), H(I)
   20    CONTINUE
      END IF
      STOP
*
99999 FORMAT (' Residual sum of squares = ',F10.2)
99998 FORMAT (' Degrees of freedom = ',F10.2)
99997 FORMAT (' RHO = ',F10.2)
99996 FORMAT (/'     Input data',16X,'Output results',/'   I    X     ',
     +       '  Y   ',9X,'YHAT      H')
99995 FORMAT (I4,2F8.3,6X,2F8.3)
      END
```

9.2 Program Data

```
G10ACF Example Program Data
43
'D', 'U'
 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
10000  0.001  40  12.0
```

9.3 Program Results

```
G10ACF Example Program Results

Residual sum of squares =      10.35
Degrees of freedom =      25.00
RHO =       2.68

    Input data                Output results
  I    X       Y             YHAT      H
  1   0.900   3.000          3.373   0.534
  2   1.000   3.900          3.406   0.427
  3   1.800   3.400          3.642   0.313
  4   1.900   3.700          3.686   0.313
  5   2.200   3.900          3.839   0.448
  6   4.200   5.100          4.614   0.564
  7   4.800   4.200          4.576   0.442
  8   5.100   4.600          4.715   0.189
  9   5.200   4.850          4.783   0.407
 10   5.800   5.600          5.193   0.455
 11   6.900   5.100          5.184   0.592
 12   7.900   4.800          4.958   0.530
 13   8.100   5.200          4.931   0.235
 14   8.500   5.300          4.845   0.245
 15   8.800   4.100          4.763   0.271
 16   8.900   4.900          4.748   0.292
 17   9.800   4.800          4.850   0.301
 18   9.900   4.900          4.875   0.277
 19  10.400   5.000          4.970   0.173
 20  10.500   5.200          4.977   0.154
 21  10.600   5.000          4.979   0.285
 22  10.800   5.100          4.970   0.136
 23  11.000   4.400          4.961   0.137
 24  11.100   4.900          4.964   0.284
 25  11.300   5.100          4.975   0.162
 26  11.500   5.500          4.975   0.186
 27  11.800   4.600          4.930   0.213
 28  11.900   5.100          4.911   0.220
 29  12.400   5.200          4.852   0.206
 30  12.500   4.100          4.857   0.196
 31  12.700   3.400          4.900   0.189
 32  12.800   6.600          4.932   0.193
 33  13.200   5.300          4.955   0.488
 34  13.800   3.700          4.797   0.408
 35  14.500   5.700          5.076   0.559
 36  15.500   4.900          4.979   0.445
 37  15.600   4.900          4.946   0.535
```