NAG Library Routine Document

G10ACF

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, DF, RES,     &
                   H, CRIT, RHO, U, TOL, MAXCAL, WK, IFAIL)

INTEGER            N, LDC, MAXCAL, IFAIL
REAL (KIND=nag_wp) X(N), Y(N), WT(*), YHAT(N), C(LDC,3), RSS, DF, 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), 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 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 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 fitted 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:

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

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

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

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

(ii) Minimize the cross-validation (CV), i.e., find ρ such that the CV is minimized, where

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

(iii) Minimize the generalized cross-validation (GCV), i.e., find ρ such that the GCV is minimized, 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 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 (1986). C05AZF is used to solve for ρ given ν_0 and the method of E04ABF/E04ABA is used to minimize the GCV or CV.

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 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.

METHOD = 'C'
Cross-validation is used.

METHOD = 'D'
The degrees of freedom are specified.

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.

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 observations.

Constraint: $\mathrm{N} \geq 3$.

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 1 if WEIGHT = 'U' and at least N if WEIGHT = 'W'.

On entry: if WEIGHT = 'W', WT must contain the n weights.

If WEIGHT = 'U', WT is not referenced and unit weights are assumed.

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

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

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

8: C(LDC,3) – REAL (KIND=nag_wp) 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 $x_i \leq t < x_{i+1}$ 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 (KIND=nag_wp) *Output*

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

11: DF – REAL (KIND=nag_wp) *Output*

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

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

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

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

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

14: CRIT – REAL (KIND=nag_wp) *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 (KIND=nag_wp) *Output*

On exit: the smoothing parameter, ρ.

16: U – REAL (KIND=nag_wp) *Input*

On entry: the upper bound on the smoothing parameter. If $U \leq TOL$, $U = 1000.0$ will be used instead. See Section 8 for details on how this parameter is used.

17: TOL – REAL (KIND=nag_wp) *Input*

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

18: MAXCAL – INTEGER *Input*

On entry: the maximum number of spline evaluations to be used in finding the value of ρ. If $MAXCAL < 3$, $MAXCAL = 100$ will be used instead.

19: WK$(7 \times (N + 2))$ – REAL (KIND=nag_wp) array *Workspace*

20: 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, if you are not familiar with this parameter, 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 METHOD $\neq$ 'C', 'G' or 'D',
or WEIGHT $\neq$ 'W' or 'U',
or METHOD = 'D' and $CRIT \leq 2.0$,
or METHOD = 'D' and $CRIT > N$.

$IFAIL = 2$

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

$IFAIL = 3$

On entry, $X(i) \geq X(i + 1)$, 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 $\rho >$ U. Try a larger value of U; see Section 8.

7 Accuracy

When minimizing the cross-validation or generalized 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 minimized 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

This example uses the data given by Hastie and Tibshirani (1990), which consists of 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

```
    Program g10acfe

!     G10ACF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g10acf, g10zaf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: crit, df, rho, rss, tol, u
      Integer                          :: i, ifail, ldc, lwk, lwt, maxcal, n,  &
                                          nord
      Character (1)                    :: method, weight
```

```
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: c(:,:), h(:), res(:), wk(:), wt(:),  &
                                          wwt(:), x(:), xord(:), y(:),         &
                                          yhat(:), yord(:)
      Integer, Allocatable             :: iwrk(:)
!     .. Executable Statements ..
      Write (nout,*) 'G10ACF Example Program Results'
      Write (nout,*)

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

      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      ldc = n - 1
      lwk = 7*(n+2)
      Allocate (x(n),y(n),wt(lwt),xord(n),yord(n),wwt(n),yhat(n),c(ldc,3), &
        res(n),h(n),wk(lwk),iwrk(n))

!     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

!     Read in control parameters
      Read (nin,*) u, tol, maxcal, crit

!     Sort data, removing ties and weighting accordingly
      ifail = 0
      Call g10zaf(weight,n,x,y,wt,nord,xord,yord,wwt,rss,iwrk,ifail)

!     Fit cubic spline
      ifail = 0
      Call g10acf(method,'W',nord,xord,yord,wwt,yhat,c,ldc,rss,df,res,h,crit, &
        rho,u,tol,maxcal,wk,ifail)

!     Display results
      Write (nout,99999) 'Residual sum of squares = ', rss
      Write (nout,99999) 'Degrees of freedom = ', df
      Write (nout,99999) 'RHO = ', rho
      Write (nout,*)
      Write (nout,*) '     Input data                Output results'
      Write (nout,*) '   I    X       Y            YHAT      H'
      Write (nout,99998)(i,xord(i),yord(i),yhat(i),h(i),i=1,nord)

99999 Format (1X,A,F10.2)
99998 Format (I4,2F8.3,6X,2F8.3)
    End Program g10acfe
```

9.2 Program Data

```
G10ACF Example Program Data
'D'  'U'  43                              :: METHOD,WEIGHT,N
 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
10000  0.001  40  12.0                     :: U,TOL,MAXCAL,CRIT
```

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   0.234
   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   0.276
   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
```