NAG Toolbox for MATLAB

g10ab

1 Purpose

g10ab fits a cubic smoothing spline for a given smoothing parameter.

2 Syntax

```
[yhat, c, rss, df, res, h, wk, ifail] = g10ab(mode, weight, x, y, wt,
rho, c, wk, 'n', n)
```

3 Description

g10ab 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 non-linear 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

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

for a matrix H. The residual degrees of freedom for the spline is 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 g10ac for further details.

g10ab 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 g10za.

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 need only be carried out once and then phase repeated for different values of ρ. If the spline is being fitted as part of an iterative weighted least-squares procedure phases and have to be repeated for each set of weights. In either case, phase 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 Parameters

5.1 Compulsory Input Parameters

1: **mode – string**

Indicates in which mode the function is to be used.

> If **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 g10ab or when the coefficients are not required.
>
> If **mode** = 'Q', fitting only is performed. Initialization must have been performed previously by a call to g10ab with **mode** = 'P'. This quick fit may be called repeatedly with different values of **rho** without re-initialization.
>
> If **mode** = 'F', initialization and full fitting is performed and the function coefficients are calculated.

Constraint: **mode** = 'P', 'Q' or 'F'.

2: **weight – string**

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: **x**(**n**) **– double array**

The distinct and ordered values x_i for $i = 1, 2, \ldots, n$.

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

4: **y**(**n**) **– double array**

The values y_i, for $i = 1, 2, \ldots, n$.

5: **wt**($*$) **– double array**

Note: the dimension of the array **wt** must be at least 1 if **weight** = 'U' and at least **n** if **weight** = 'W'.

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', $\mathbf{wt}(i) > 0.0$, for $i = 1, 2, \ldots, n$.

6: **rho – double scalar**

ρ, the smoothing parameter.

Constraint: **rho** ≥ 0.0.

7: **c(ldc,3) – double array**

ldc, the first dimension of the array, must be at least **n** $- 1$.

If **mode** = 'Q', **c** must be unaltered from the previous call to g10ab with **mode** = 'P'. Otherwise **c** need not be set.

8: **wk(9 × n + 14) – double array**

If **mode** = 'Q', **wk** must be unaltered from the previous call to g10ab with **mode** = 'P'. Otherwise **wk** is used as workspace and need not be set.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the arrays **x**, **y**, **yhat**, **res**, **h**, **wk**. (An error is raised if these dimensions are not equal.)

n, the number of distinct observations.

Constraint: **n** ≥ 3.

5.3 Input Parameters Omitted from the MATLAB Interface

ldc

5.4 Output Parameters

1: **yhat(n) – double array**

The fitted values, $\hat{y}_i$, for $i = 1, 2, \ldots, n$.

2: **c(ldc,3) – double array**

If **mode** = 'F', **c** contains the spline coefficients. More precisely, the value of the spline at t is given by $((\mathbf{c}(i,3) \times d + \mathbf{c}(i,2)) \times d + \mathbf{c}(i,1)) \times d + \hat{y}_i$, 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 g10ab with **mode** = 'Q'.

3: **rss – double scalar**

The (weighted) residual sum of squares.

4: **df – double scalar**

The residual degrees of freedom.

5: **res(n) – double array**

The (weighted) residuals, r_i, for $i = 1, 2, \ldots, n$.

6: **h(n) – double array**

The leverages, h_{ii}, for $i = 1, 2, \ldots, n$.

7: **wk**$(\mathbf{9} \times \mathbf{n} + \mathbf{14})$ – **double array**

If **mode** = 'P' or 'Q', **wk** contains information that will be used in a subsequent call to g10ab with **mode** = 'Q'.

8: **ifail** – **int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **n** < 3,
or **ldc** < **n** − 1,
or **rho** < 0.0,
or **mode** ≠ 'Q', 'P' or 'F',
or **weight** ≠ 'W' or 'U'.

ifail = 2

On entry, **weight** = 'W' and at least one element of **wt** ≤ 0.0.

ifail = 3

On entry, $\mathbf{x}(i) \geq \mathbf{x}(i+1)$, for some i, $i = 1, 2, \ldots, n-1$.

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 Further Comments

The time taken by g10ab is of order n.

Regression splines with a small $(< n)$ number of knots can be fitted by e02ba and e02be.

9 Example

```
mode = ’F’;
weight = ’W’;
x = [0.9;
     1;
     1.8;
     1.9;
     2.2;
     4.2;
     4.8;
     5.1;
     5.2;
     5.8;
     6.9;
     7.9;
     8.1;
     8.5;
     8.8;
     8.9;
     9.8;
     9.9;
     10.4;
     10.5;
     10.6;
     10.8;
     11;
     11.1;
     11.3;
     11.5;
     11.8;
     11.9;
     12.4;
     12.5;
     12.7;
     12.8;
     13.2;
     13.8;
     14.5;
     15.5;
     15.6];
y = [3;
     3.9;
     3.4;
     3.7;
     3.9;
     5.1;
     4.2;
     4.6;
     4.85;
     5.6;
     5.1;
     4.8;
     5.2;
     5.3;
     4.1;
     4.9;
     4.8;
     4.9;
     5;
     5.2;
     5;
     5.1;
     4.4;
     4.9;
     5.1;
     5.5;
     4.6;
     5.1;
     5.2;
     4.1;
     3.4;
     6.6;
     5.3;
     3.7;
     5.7;
     4.9;
     4.9];
wt = [1;
     1;
     1;
     1;
     1;
     1;
     2;
     1;
     2;
     1;
     1;
     2;
     1;
     1;
     1;
     1;
     1;
     1;
     1;
     1;
     2;
     1;
     1;
     2;
     1;
     1;
     1;
     1;
     1;
     1;
     1;
     1;
     2;
     1;
     1;
     1;
     1];
rho = 10;
c = zeros(36, 3);
wk = zeros(447, 1);
[yhat, cOut, rss, df, res, h, wkOut, ifail] = ...
    g10ab(mode, weight, x, y, wt, rho, c, wk)
```

```
yhat =
    3.3674
    3.4008
    3.6642
    3.7016
    3.8214
    4.5265
    4.6471
    4.7561
    4.7993
    5.0458
    5.1204
    4.9590
    4.9262
    4.8595
    4.8172
    4.8095
    4.8676
    4.8818
    4.9445
    4.9521
    4.9572
    4.9613
    4.9614
    4.9618
    4.9623
    4.9568
    4.9338
    4.9251
    4.8943
    4.8944
    4.9051
    4.9138
    4.9239
    4.8930
    4.9938
    4.9773
    4.9682
cOut =
    0.3359         0   -0.1653
    0.3310   -0.0496    0.0593
    0.3655    0.0928   -0.0596
    0.3823    0.0749   -0.0603
    0.4109    0.0206   -0.0249
    0.1944   -0.1289    0.2331
    0.2915    0.2907   -0.1692
    0.4202    0.1384   -0.2394
    0.4407    0.0665   -0.1938
    0.3112   -0.2823    0.0555
   -0.1085   -0.0992    0.0463
   -0.1679    0.0398   -0.0967
   -0.1636   -0.0182    0.0264
   -0.1655    0.0135    0.2246
   -0.0968    0.2156   -0.0980
   -0.0566    0.1862   -0.0574
    0.1392    0.0314   -0.0878
    0.1428    0.0050   -0.0796
    0.0882   -0.1143   -0.0546
    0.0637   -0.1307    0.0569
    0.0392   -0.1137    0.0954
    0.0052   -0.0564    0.1578
    0.0016    0.0382   -0.0948
    0.0064    0.0098   -0.1504
   -0.0077   -0.0804   -0.0884
   -0.0505   -0.1335    0.1560
   -0.0885    0.0069    0.0058
   -0.0869    0.0086    0.0845
   -0.0150    0.1353    0.2220
    0.0188    0.2019   -0.1354
    0.0833    0.1207   -0.8125
    0.0831   -0.1230   -0.0539
   -0.0413   -0.1878    0.2844
    0.0406    0.3242   -0.2523
    0.1237   -0.2056    0.0655
   -0.0911   -0.0092    0.0307
rss =
   11.2876
df =
   27.7845
res =
   -0.3674
    0.4992
   -0.2642
   -0.0016
    0.0786
    0.5735
   -0.6323
   -0.1561
    0.0717
    0.5542
   -0.0204
   -0.2248
    0.2738
    0.4405
   -0.7172
    0.0905
   -0.0676
    0.0182
    0.0555
    0.2479
    0.0605
    0.1387
   -0.5614
   -0.0874
    0.1377
    0.5432
   -0.3338
    0.1749
    0.3057
   -0.7944
   -1.5051
```

```
    1.6862
    0.5318
   -1.1930
    0.7062
   -0.0773
   -0.0682
h =
    0.4813
    0.3955
    0.2548
    0.2670
    0.3463
    0.3688
    0.3608
    0.1614
    0.3365
    0.2948
    0.3899
    0.4175
    0.1879
    0.1839
    0.1993
    0.2076
    0.2013
    0.1893
    0.1286
    0.1188
    0.2232
    0.1046
    0.1050
    0.2154
    0.1177
    0.1300
    0.1453
    0.1489
    0.1482
    0.1463
    0.1470
    0.1503
    0.3694
    0.2850
    0.3666
    0.4252
    0.4961
wkOut =
     array elided
ifail =
           0
```