NAG Toolbox

nag_smooth_kerndens_gauss (g10ba)

1 Purpose

nag_smooth_kerndens_gauss (g10ba) performs kernel density estimation using a Gaussian kernel.

Note: This function is scheduled to be withdrawn, please see g10ba in Advice on Replacement Calls for Withdrawn/Superseded Routines..

2 Syntax

```
[smooth, t, fft, ifail] = nag_smooth_kerndens_gauss(x, window, slo, shi, usefft, fft, 'n', n, 'ns', ns)
[smooth, t, fft, ifail] = g10ba(x, window, slo, shi, usefft, fft, 'n', n, 'ns', ns)
```

3 Description

Given a sample of n observations, $x_1, x_2, \ldots, x_n$, from a distribution with unknown density function, $f(x)$, an estimate of the density function, $\hat{f}(x)$, may be required. The simplest form of density estimator is the histogram. This may be defined by:

$$\hat{f}(x) = \tfrac{1}{nh} n_j, \quad a + (j-1)h < x < a + jh, \quad j = 1, 2, \ldots, n_s,$$

where n_j is the number of observations falling in the interval $a + (j-1)h$ to $a + jh$, a is the lower bound to the histogram and $b = n_s h$ is the upper bound. The value h is known as the window width. To produce a smoother density estimate a kernel method can be used. A kernel function, $K(t)$, satisfies the conditions:

$$\int_{-\infty}^{\infty} K(t)\, dt = 1 \quad \text{and} \quad K(t) \geq 0.$$

The kernel density estimator is then defined as

$$\hat{f}(x) = \tfrac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right).$$

The choice of K is usually not important but to ease the computational burden use can be made of the Gaussian kernel defined as

$$K(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}.$$

The smoothness of the estimator depends on the window width h. The larger the value of h the smoother the density estimate. The value of h can be chosen by examining plots of the smoothed density for different values of h or by using cross-validation methods (see Silverman (1990)).

Silverman (1982) and Silverman (1990) show how the Gaussian kernel density estimator can be computed using a fast Fourier transform (**fft**). In order to compute the kernel density estimate over the range a to b the following steps are required.

(i) Discretize the data to give n_s equally spaced points t_l with weights ξ_l (see Jones and Lotwick (1984)).

(ii) Compute the **fft** of the weights ξ_l to give Y_l.

(iii) Compute $\zeta_l = e^{-\frac{1}{2}h^2 s_l^2} Y_l$ where $s_l = 2\pi l/(b-a)$.

(iv) Find the inverse **fft** of ζ_l to give $\hat{f}(x)$.

To compute the kernel density estimate for further values of h only steps (iii) and (iv) need be repeated.

4 References

Jones M C and Lotwick H W (1984) Remark AS R50. A remark on algorithm AS 176. Kernel density estimation using the Fast Fourier Transform *Appl. Statist.* **33** 120–122

Silverman B W (1982) Algorithm AS 176. Kernel density estimation using the fast Fourier transform *Appl. Statist.* **31** 93–99

Silverman B W (1990) *Density Estimation* Chapman and Hall

5 Parameters

5.1 Compulsory Input Parameters

1: **x**(**n**) – REAL (KIND=nag_wp) array

The n observations, x_i, for $i = 1, 2, \ldots, n$.

2: **window** – REAL (KIND=nag_wp)

h, the window width.

Constraint: **window** > 0.0.

3: **slo** – REAL (KIND=nag_wp)

a, the lower limit of the interval on which the estimate is calculated. For most applications **slo** should be at least three window widths below the lowest data point.

Constraint: **slo** $<$ **shi**.

4: **shi** – REAL (KIND=nag_wp)

b, the upper limit of the interval on which the estimate is calculated. For most applications **shi** should be at least three window widths above the highest data point.

5: **usefft** – LOGICAL

Must be set to *false* if the values of Y_l are to be calculated by nag_smooth_kerndens_gauss (g10ba) and to *true* if they have been computed by a previous call to nag_smooth_kerndens_gauss (g10ba) and are provided in **fft**. If **usefft** $= true$ then the arguments **n**, **slo**, **shi**, **ns** and **fft** must remain unchanged from the previous call to nag_smooth_kerndens_gauss (g10ba) with **usefft** $= false$.

6: **fft**(**ns**) – REAL (KIND=nag_wp) array

If **usefft** $= true$, **fft** must contain the fast Fourier transform of the weights of the discretized data, ξ_l, for $l = 1, 2, \ldots, n_s$. Otherwise **fft** need not be set.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **x**.

n, the number of observations in the sample.

Constraint: **n** > 0.

2: **ns** – INTEGER

Default: the dimension of the array **fft**.

The number of points at which the estimate is calculated, n_s.

Constraints:

ns ≥ 2;
The largest prime factor of **ns** must not exceed 19, and the total number of prime factors of **ns**, counting repetitions, must not exceed 20.

5.3 Output Parameters

1: **smooth**(**ns**) – REAL (KIND=nag_wp) array

The n_s values of the density estimate, $\hat{f}(t_l)$, for $l = 1, 2, \ldots, n_s$.

2: **t**(**ns**) – REAL (KIND=nag_wp) array

The points at which the estimate is calculated, t_l, for $l = 1, 2, \ldots, n_s$.

3: **fft**(**ns**) – REAL (KIND=nag_wp) array

The fast Fourier transform of the weights of the discretized data, ξ_l, for $l = 1, 2, \ldots, n_s$.

4: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **n** ≤ 0,
or **ns** < 2,
or **shi** $\leq$ **slo**,
or **window** ≤ 0.0.

ifail $= 2$

On entry, nag_smooth_kerndens_gauss (g10ba) has been called with **usefft** $= true$ but the function has not been called previously with **usefft** $= false$,
or nag_smooth_kerndens_gauss (g10ba) has been called with **usefft** $= true$ but some of the arguments **n**, **slo**, **shi**, **ns** have been changed since the previous call to nag_smooth_kerndens_gauss (g10ba) with **usefft** $= false$.

ifail $= 3$

On entry, at least one prime factor of **ns** is greater than 19 or **ns** has more than 20 prime factors.

ifail $= 4$ (*warning*)

On entry, the interval given by **slo** to **shi** does not extend beyond three window widths at either extreme of the dataset. This may distort the density estimate in some cases.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

See Jones and Lotwick (1984) for a discussion of the accuracy of this method.

8 Further Comments

The time for computing the weights of the discretized data is of order n, while the time for computing the **fft** is of order $n_s \log(n_s)$, as is the time for computing the inverse of the **fft**.

9 Example

Data is read from a file and the density estimated. The first 20 values are then printed. The full estimated density function is shown in the accompanying plot.

9.1 Program Text

```
function g10ba_example

fprintf('g10ba example results\n\n');

% sample data
x = [  0.114 -0.232 -0.570  1.853 -0.994 ...
      -0.374 -1.028  0.509  0.881 -0.453 ...
       0.588 -0.625 -1.622 -0.567  0.421 ...
      -0.475  0.054  0.817  1.015  0.608 ...
      -1.353 -0.912 -1.136  1.067  0.121 ...
      -0.075 -0.745  1.217 -1.058 -0.894 ...
       1.026 -0.967 -1.065  0.513  0.969 ...
       0.582 -0.985  0.097  0.416 -0.514 ...
       0.898 -0.154  0.617 -0.436 -1.212 ...
      -1.571  0.210 -1.101  1.018 -1.702 ...
      -2.230 -0.648 -0.350  0.446 -2.667 ...
       0.094 -0.380 -2.852 -0.888 -1.481 ...
      -0.359 -0.554  1.531  0.052 -1.715 ...
       1.255 -0.540  0.362 -0.654 -0.272 ...
      -1.810  0.269 -1.918  0.001  1.240 ...
      -0.368 -0.647 -2.282  0.498  0.001 ...
      -3.059 -1.171  0.566  0.948  0.925 ...
       0.825  0.130  0.930  0.523  0.443 ...
      -0.649  0.554 -2.823  0.158 -1.180 ...
       0.610  0.877  0.791 -0.078  1.412 ];

% Control parameters
window = 0.4;
slo    = -5;
shi    = 5;
usefft = false;
fft    = zeros(100,1);

% Perform kernel density estimation
[smooth, t, fft, ifail] = g10ba( ...
                                 x, window, slo, shi, usefft, fft);

% Display the results
fprintf('Window Width Used = %11.4e\n', window);
fprintf('Interval = (%11.4e, %11.4e)\n\n', slo, shi);
fprintf('First 20 output values:\n\n');
fprintf('      Time        Density\n');
fprintf('      Point       Estimate\n');
fprintf(' -------------------------\n');
fprintf('%13.3e%13.3e\n', [t(1:20), smooth(1:20)]');
```

```
fig1 = figure;
plot(t,smooth);
title('Plot of the Smoothed Density (window = 0.4)');
xlabel('t');
ylabel('Density estimate');
set(gca, 'XTick', [-5:5]);
```

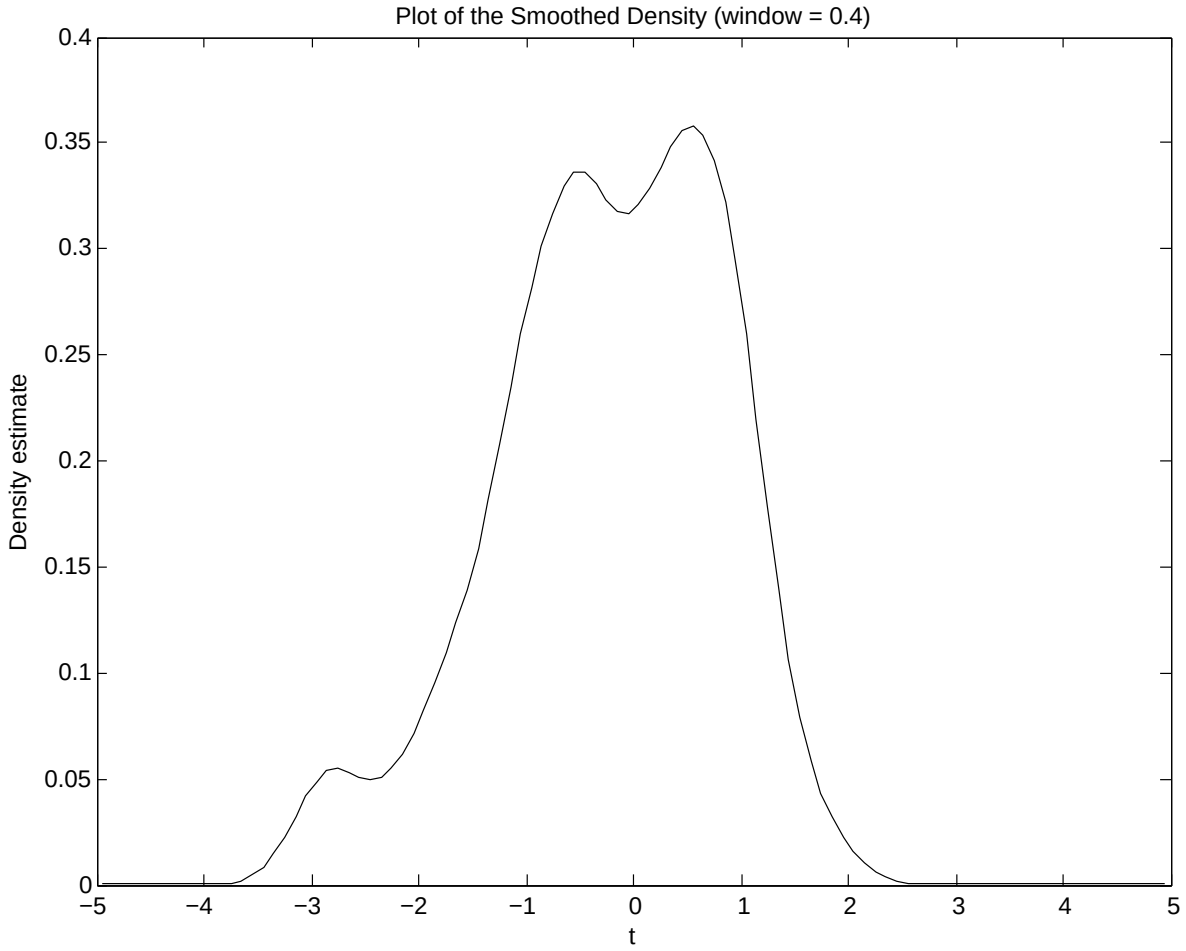
9.2 Program Results

```
     g10ba example results

Window Width Used =  4.0000e-01
Interval = (-5.0000e+00,  5.0000e+00)

First 20 output values:

      Time        Density
      Point       Estimate
 ---------------------------
   -4.950e+00    4.108e-12
   -4.850e+00    3.915e-11
   -4.750e+00    3.309e-10
   -4.650e+00    2.480e-09
   -4.550e+00    1.649e-08
   -4.450e+00    9.730e-08
   -4.350e+00    5.097e-07
   -4.250e+00    2.372e-06
   -4.150e+00    9.817e-06
   -4.050e+00    3.615e-05
   -3.950e+00    1.186e-04
   -3.850e+00    3.475e-04
   -3.750e+00    9.100e-04
   -3.650e+00    2.136e-03
   -3.550e+00    4.504e-03
   -3.450e+00    8.556e-03
   -3.350e+00    1.468e-02
   -3.250e+00    2.283e-02
   -3.150e+00    3.225e-02
   -3.050e+00    4.154e-02
```


Plot of the Smoothed Density (window = 0.4)
Density estimate
t
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
−5
−4
−3
−2
−1
0
1
2
3
4
5