NAG Toolbox

nag_smooth_kerndens_gauss2 (g10bb)

1 Purpose

nag_smooth_kerndens_gauss2 (g10bb) performs kernel density estimation using a Gaussian kernel.

2 Syntax

```
[window, slo, shi, smooth, t, rcomm, ifail] = nag_smooth_kerndens_gauss2(x,
fcall, rcomm, 'n', n, 'wtype', wtype, 'window', window, 'slo', slo, 'shi', shi,
'ns', ns)
[window, slo, shi, smooth, t, rcomm, ifail] = g10bb(x, fcall, rcomm, 'n', n,
'wtype', wtype, 'window', window, 'slo', slo, 'shi', shi, '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, $b = n_s h$ is the upper bound and n_s is the total number of intervals. 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

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

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

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

If **fcall** $= 0$, **x** must be unchanged since the last call to nag_smooth_kerndens_gauss2 (g10bb).

2: **fcall** – INTEGER

If **fcall** $= 1$ then the values of Y_l are to be calculated by this call to nag_smooth_kerndens_gauss2 (g10bb), otherwise it is assumed that the values of Y_l were calculated by a previous call to this routine and the relevant information is stored in **rcomm**.

Constraint: **fcall** $= 0$ or 1.

3: **rcomm**(**ns** + **20**) – REAL (KIND=nag_wp) array

Communication array, used to store information between calls to nag_smooth_kerndens_gauss2 (g10bb).

If **fcall** $= 0$, **rcomm** must be unchanged since the last call to nag_smooth_kerndens_gauss2 (g10bb).

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the number of observations in the sample.

If **fcall** $= 0$, **n** must be unchanged since the last call to nag_smooth_kerndens_gauss2 (g10bb).

Constraint: **n** > 0.

2: **wtype** – INTEGER

Suggested value: **wtype** $= 2$ and **window** $= 1.0$.

Default: 2

How the window width, h, is to be calculated:

wtype $= 1$
: h is supplied in **window**.

wtype $= 2$
: h is to be calculated from the data, with

$$h = m \times \left(\frac{0.9 \times \min(q_{75} - q_{25}, \sigma)}{n^{0.2}} \right)$$

where $q_{75} - q_{25}$ is the inter-quartile range and σ the standard deviation of the sample, x, and m is a multipler supplied in **window**. The 25% and 75% quartiles, q_{25} and q_{75}, are calculated using nag_stat_quantiles (g01am). This is the "rule-of-thumb" suggested by Silverman (1990).

Constraint: **wtype** $= 1$ or 2.

3: **window** – REAL (KIND=nag_wp)

Suggested value: **window** $= 1.0$ and **wtype** $= 2$.

Default: 1.0

If **wtype** $= 1$, then h, the window width. Otherwise, m, the multiplier used in the calculation of h.

Constraint: **window** > 0.0.

4: **slo** – REAL (KIND=nag_wp)

Suggested value: **slo** $= 3.0$ and **shi** $= 0.0$ which would cause a and b to be set 3 window widths below and above the lowest and highest data points respectively.

Default: 3.0

If **slo** $<$ **shi** then a, the lower limit of the interval on which the estimate is calculated. Otherwise, a and b, the lower and upper limits of the interval, are calculated as follows:

$$\begin{aligned} a &= \min_i\{x_i\} - \textbf{slo} \times h \\ b &= \max_i\{x_i\} + \textbf{slo} \times h \end{aligned}$$

where h is the window width.

For most applications a should be at least three window widths below the lowest data point.

If **fcall** $= 0$, **slo** must be unchanged since the last call to nag_smooth_kerndens_gauss2 (g10bb).

5: **shi** – REAL (KIND=nag_wp)

Default: 0.0

If **slo** $<$ **shi** then b, the upper limit of the interval on which the estimate is calculated. Otherwise a value for b is calculated from the data as stated in the description of **slo** and the value supplied in **shi** is not used.

For most applications b should be at least three window widths above the highest data point.

If **fcall** $= 0$, **shi** must be unchanged since the last call to nag_smooth_kerndens_gauss2 (g10bb).

6: **ns** – INTEGER

Suggested value: **ns** $= 512$.

Default: 512

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

If **fcall** $= 0$, **ns** must be unchanged since the last call to nag_smooth_kerndens_gauss2 (g10bb).

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: **window** – REAL (KIND=nag_wp)

Suggested value: **window** $= 1.0$ and **wtype** $= 2$.

Default: 1.0

h, the window width actually used.

2: **slo** – REAL (KIND=nag_wp)

Suggested value: **slo** $= 3.0$ and **shi** $= 0.0$ which would cause a and b to be set 3 window widths below and above the lowest and highest data points respectively.

Default: 3.0

a, the lower limit actually used.

3: **shi** – REAL (KIND=nag_wp)

Default: 0.0

b, the upper limit actually used.

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

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

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

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

6: **rcomm**(**ns** + **20**) – REAL (KIND=nag_wp) array

The last **ns** elements of **rcomm** contain the fast Fourier transform of the weights of the discretized data, that is **rcomm**$(l + 20) = Y_l$, for $l = 1, 2, \ldots, n_s$.

7: **ifail** – INTEGER

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

6 Error Indicators and Warnings

Note: nag_smooth_kerndens_gauss2 (g10bb) may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the function:

ifail $= 11$

Constraint: **n** > 0.

ifail $= 12$

Constraint: if **fcall** $= 0$, **n** must be unchanged since previous call.

ifail $= 31$

Constraint: **wtype** $= 1$ or 2.

ifail $= 41$

Constraint: **window** > 0.0.

ifail $= 51$

Constraint: if **fcall** $= 0$, **slo** must be unchanged since previous call.

ifail $= 61$ (*warning*)

slo is not at least three window widths below the lowest data point or **shi** is not at least three window widths above the highest data point. All output values have been returned.

ifail $= 62$

Constraint: if **fcall** $= 0$, **shi** must be unchanged since previous call.

ifail $= 71$

Constraint: **ns** ≥ 2.

ifail $= 72$

Constraint: largest prime factor of **ns** must not exceed 19.

ifail $= 73$

Constraint: total number of prime factors of **ns** must not exceed 20.

ifail $= 74$

Constraint: if **fcall** $= 0$, **ns** must be unchanged since previous call.

ifail $= 101$

Constraint: **fcall** $= 0$ or 1.

ifail $= 111$

rcomm has been corrupted between calls.

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.

9.1 Program Text

```
     function g10bb_example

fprintf('g10bb example results\n\n');

% kernel density estimation from 100 values
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];

% Calculate window width from data.
wtype = nag_int(2);

% First Call
fcall = nag_int(1);
ns    = 512;
rcomm = zeros(ns+20,1);

% Perform kernel density estimation
[window, slo, shi, smooth, t, rcomm, ifail] = ...
   g10bb( ...
          x, fcall, rcomm, 'wtype',wtype);

% Display the results
fprintf('Window Width Used = %11.4e\n', window);
fprintf('Interval = (%11.4e,%11.4e)\n\n', slo, shi);
fprintf('First %2d output values:\n\n',20);
fprintf('    Time point      Density estimate\n');
fprintf('    ----------      ----------------\n');
fprintf(' %13.4f     %13.4e\n', [t(1:20), smooth(1:20)]')

fig1 = figure;
plot(t,smooth);
title('Gaussian Kernel Density Estimation');
xlabel('t');
ylabel('Density Estimate');
wind_leg = sprintf('window = %7.4f',window);
legend(wind_leg);
legend('boxoff');
```

9.2 Program Results

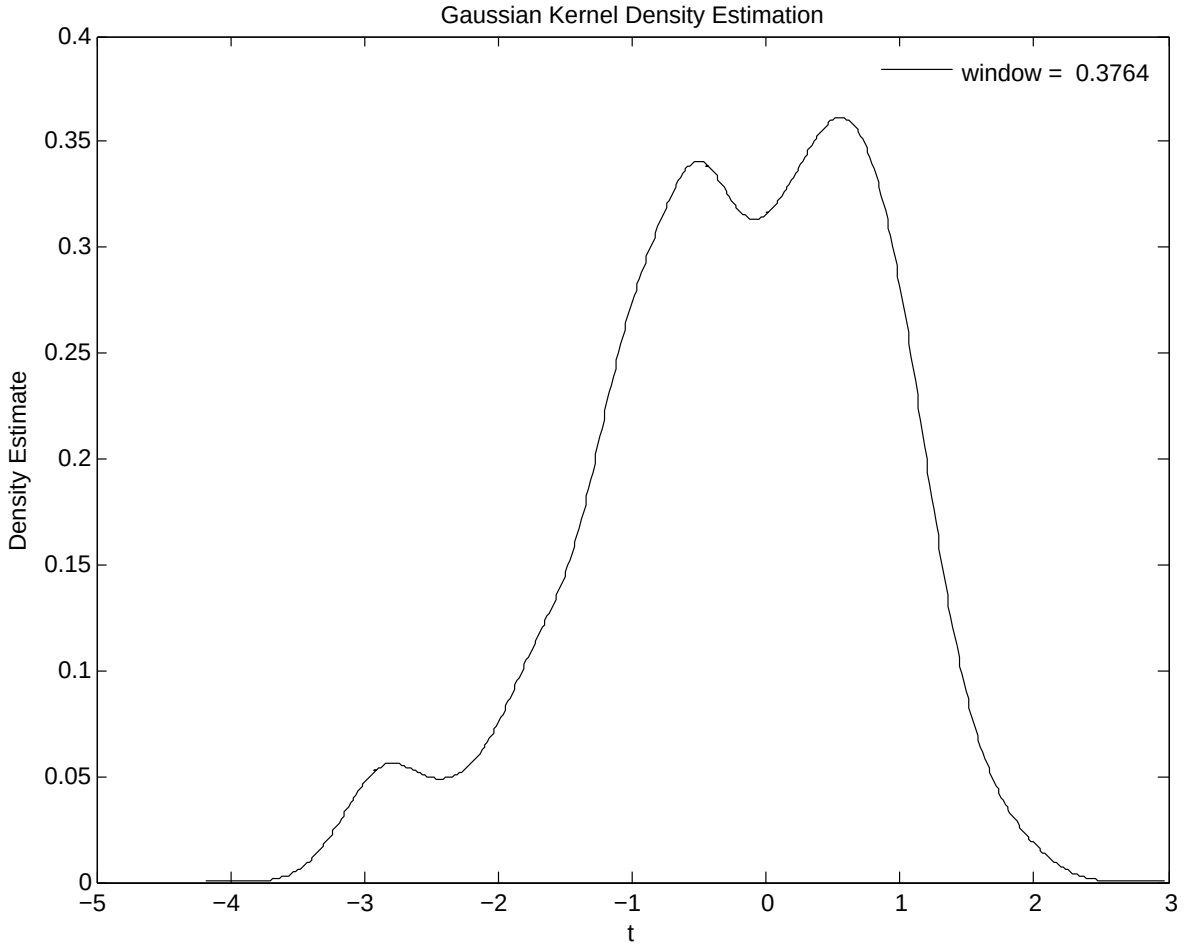
```
     g10bb example results

Window Width Used =  3.7638e-01
Interval = (-4.1882e+00, 2.9822e+00)

First 20 output values:

    Time point      Density estimate
    ----------      ----------------
       -4.1811        3.8281e-06
```

```
-4.1671         4.0305e-06
-4.1531         4.4233e-06
-4.1391         5.0212e-06
-4.1251         5.8461e-06
-4.1111         6.9279e-06
-4.0971         8.3048e-06
-4.0831         1.0025e-05
-4.0691         1.2145e-05
-4.0551         1.4736e-05
-4.0411         1.7881e-05
-4.0271         2.1677e-05
-4.0131         2.6239e-05
-3.9991         3.1700e-05
-3.9851         3.8214e-05
-3.9711         4.5960e-05
-3.9571         5.5141e-05
-3.9431         6.5990e-05
-3.9291         7.8775e-05
-3.9151         9.3796e-05
```



This plot shows the estimated density function for the example data for several window widths.