NAG Toolbox for MATLAB

g10ba

1 Purpose

g10ba performs kernel density estimation using a Gaussian kernel.

2 Syntax

```
[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:

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

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:

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

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 *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) – double array**

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

2: **window – double scalar**

h, the window width.

Constraint: **window** > 0.0.

3: **slo – double scalar**

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 – double scalar**

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 scalar**

Must be set to **false** if the values of Y_l are to be calculated by g10ba and to **true** if they have been computed by a previous call to 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 g10ba with **usefft** $=$ **false**.

6: **fft(ns) – double array**

If **usefft** $=$ **true**, then **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 – int32 scalar**

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

n, the number of observations in the sample.

Constraint: **n** > 0.

2: **ns – int32 scalar**

Default: The dimension of the arrays **smooth**, **t**, **fft**. (An error is raised if these dimensions are not equal.)

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 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **smooth(ns) – double array**

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

2: **t(ns) – double array**

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

3: **fft(ns) – double array**

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

4: **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** ≤ 0,
or **ns** < 2,
or **shi** $\leq$ **slo**,
or **window** ≤ 0.0.

ifail $= 2$

On entry, g10ba has been called with **usefft** $=$ **true** but the function has not been called previously with **usefft** $=$ **false**,
or g10ba has been called with **usefft** $=$ **true** but some of the arguments **n**, **slo**, **shi**, **ns** have been changed since the previous call to 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 (see c06ea).

ifail $= 4$

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

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

```
window = 0.1;
slo = -4;
shi = 4;
usefft = false;
fft = zeros(100,1);
[x] = g05fd(0, 1, int32(1000));
[smooth, t, fftOut, ifail] = g10ba(x, window, slo, shi, usefft, fft)
```

```
smooth =
     array elided
t =
     array elided
fftOut =
     array elided
ifail =
           0
```