```
      IF (I .NE. J) X(J) = X(J) - P * S
      I = I + 1
      J = J - 1
      IF (I .LE. J) GOTO 10
      RETURN
      END
C
      SUBROUTINE RESIDU (A, B, N1)
C
C        ALGORITHM AS 175.4 APPL. STATIST. (1982) VOL.31, NO.1
C
C        REPLACES A BY A + B * BSTAR
C
      REAL A(N1), B(N1), SDSDOT
      DO 10 I = 1, N1
   10 A(I) = SDSDOT(N1 - I + 1, A(I), B(1), 1, B(I), 1)
      RETURN
      END
```

Algorithm AS 176

Kernel Density Estimation using the Fast Fourier Transform

By B. W. SILVERMAN

School of Mathematics, University of Bath, Bath BA2 7AY, UK

[Received May 1980. Revised July 1981]

Keywords: PROBABILITY DENSITY ESTIMATION; KERNEL; FAST FOURIER TRANSFORM; CONVOLUTION

LANGUAGE

Fortran 66

DESCRIPTION AND PURPOSE

Suppose $X_1, \ldots, X_n$ are real observations from a probability density f. The kernel estimate f_n of f is defined by

$$f_n(x) = n^{-1} h^{-1} \sum_{j=1}^{n} K\{h^{-1}(x - X_j)\}, \qquad (1)$$

where K is the kernel function and h is the smoothing parameter or window width. The kernel estimator was introduced by Rosenblatt (1956) and has been the subject of much attention; for surveys see Rosenblatt (1971) and Fryer (1977). Density estimates are important for data analysis and presentation, among other applications; see, for example, Boneva *et al.* (1971) and Silverman (1978).

Both theory and practice (see Epanechnikov, 1969) suggest that the choice of kernel is not crucial to the statistical performance of the method and therefore it is quite reasonable to choose a kernel for computational efficiency. The kernel we shall use is the standard Gaussian density.

The character of the estimate is mainly governed by the choice of window width, which determines how much the data are smoothed to obtain the estimate. In this algorithm the choice of window width is left to the user. For exploratory purposes it is useful to examine estimates with several different window widths since these will highlight different features of the data; see Silverman (1981). If a single estimate is required then there are several methods available for choosing the window width; see Fryer (1977), Silverman (1978) and Habbema *et al.* (1974). The optimum window width depends on the unknown density being estimated, but it is worth noting that, if the data come from a normal distribution with standard deviation σ, then the choice

$$h = 1{\cdot}06\sigma n^{-1/5} \tag{2}$$

will, to a high degree of accuracy, minimize the integrated mean square error; see Deheuvels (1977). Thus one possible quick choice of h is (2) with an estimate of σ obtained from the data, though this may oversmooth multimodal populations somewhat. Another frequently used method is to choose subjectively between estimates drawn with different window widths. Under some circumstances, it may be advantageous to transform skew samples to approximate symmetry before applying this technique; see Fryer (1977).

The algorithm described here is designed for calculating $f_n(x)$ at a grid of points, for example in order to plot the estimate. It is highly inefficient to use (1) directly for this purpose; a method using Fourier transforms is far more efficient.

Numerical Method

The numerical method used is first to discretize the data to a very fine grid, and then to use the Fast Fourier transform to convolve the data with the kernel to obtain the estimate.

Take Fourier transforms in (1), using $\sim$ to denote Fourier transform, to obtain

$$\tilde{f}_n(s) = (2\pi)^{\frac{1}{2}} \tilde{K}(hs)\, u(s),$$

where $u(s)$ is the Fourier transform of the data

$$u(s) = (2\pi)^{-\frac{1}{2}} n^{-1} \sum_{j=1}^{n} \exp(isX_j).$$

Next, substitute the Fourier transform of the Gaussian kernel to obtain

$$\tilde{f}_n(s) = \exp(-\tfrac{1}{2}h^2 s^2)\, u(s). \tag{3}$$

A discrete approximation to $u(s)$ is found by constructing a histogram on a grid of 2^k cells and then applying the Fast Fourier transform; since all the transforms handled are those of real functions, the approach of Gentleman and Sande (1966) as implemented by Monro (1976) is used to save storage and time in the calculation. Next the discrete Fourier transform of f_n is found from (3) and finally f_n is found by inverse transformation. Because of rounding and approximation errors, this calculation may lead to some (numerically very small) negative values of f_n. These are set to zero. If estimates for several window widths are required, the discrete approximation to u need only be calculated once.

The algorithm avoids exponential underflow in the calculation of (3) by setting $\tilde{f}_n(s)$ equal to zero if $\frac{1}{2}h^2 s^2$ is larger than the constant *BIG*. Since the discrete Fourier transform imposes "wrap around" edge conditions on the convolution, it is important to do the calculations on an interval which is somewhat larger than the interval of interest; enlarging the interval by $3h$ at each end should be ample for this purpose. Of course, if circular edge conditions are required, for example when considering directional data (cf. Boneva *et al.*, 1971), then no such enlargement should be used.

Structure

SUBROUTINE DENEST(DT, NDT, DLO, DHI, WINDOW, FT, SMOOTH, NFT, ICAL, IFAULT)

Formal parameters

DT	Real array (*NDT*)	input: contains the raw data values
NDT	Integer	input: the sample size
DLO	Real	input: the lower limit of the interval on which the estimate is calculated
DHI	Real	input: the upper limit of the interval on which the estimate is calculated

			must be strictly greater than *DLO*
WINDOW	Real	input:	the window width; must be strictly positive
FT	Real array (*NFT*)	input:	if $ICAL \neq 0$, the Fourier transform of the data as previously output
		output:	the Fourier transform of the data
SMOOTH	Real array (*NFT*)	output:	the values of the density estimate. *SMOOTH* (*I*) contains the value of the estimate at $DLO+(I-\frac{1}{2})(DHI-DLO)/NFT$
NFT	Integer	input:	number of points at which estimate is calculated *NFT* must be a positive integer power of 2 within the range $2^{**}KFTLO \leqslant NFT \leqslant 2^{**}KFTHI$
ICAL	Integer	input:	control parameter 0: compute *FT* and *SMOOTH* from *DT* 1: use previously computed array *FT*
IFAULT	Integer	output:	fault indicator, equal to: 1 if illegal value of *NFT* provided; 2 if *WINDOW* is not positive; 3 if $DLO \geqslant DHI$; 0 otherwise
Constants			
BIG	Real		Chosen so that *EXP*(*BIG*) is a very large number within the range of the machine
KFTLO	Integer		Logarithm to base 2 of lowest acceptable value of *NFT*; must be at least the smallest integer for which the auxiliary routines *FORRT* and *REVRT* will accept 2***KFTLO* values
KFTHI	Integer		Logarithm to base 2 of highest acceptable value of *NFT*; the integer 2***KFTHI* must be within the machine range and must be an acceptable number of values for the auxiliary routines *FORRT* and *REVRT*

Auxiliary routines

Subroutines *FORRT* and *REVRT* which perform forward and reverse discrete Fast Fourier transformation of real data are required. These should be the routines of Monro (1976) or routines which do the equivalent calculation.

The routine *FORRT* takes an array $X(M)$ of $M = 2^k$ real values and returns their discrete Fourier transform $\{Y\}$ stored with the real parts of Y_0 to $Y_{M/2}$ in locations $X(1)$ to $X(M/2+1)$ and the imaginary parts of Y_1 to $Y_{M/2-1}$ stored $M/2$ locations above their corresponding real parts.

Restrictions and Remarks

Because of restrictions in the Fourier transform routines *NFT* must be chosen to be equal to 2^k with k an integer, $3 \leqslant k \leqslant 21$ if the routines of Monro (1976) are used and if 2^{21} is representable on the machine. Since (see Accuracy below) there is little benefit in using very large values of *NFT* and very small values may lead to unacceptable discretization errors, the present version will only accept *NFT* equal to 2^k with $5 \leqslant k \leqslant 11$, but this range may be extended by altering the constants *KFTLO* and *KFTHI*.

The constants *DLO* and *DHI* must be chosen with care; see the remarks in NUMERICAL METHOD above.

The routine may be called successively with *ICAL* set non-zero after the first call in order to obtain various estimates from the same data. If this is done, the contents of *FT* and the values of *NFT*, *DLO* and *DHI* must not be altered between calls, though of course different values of *WINDOW* may be used.

If the Fourier transform of the data is *not* required for successive calls, then the routine may be called with *SMOOTH* and *FT* equal to the same actual array, provided that this violation of the Fortran 66 standard is acceptable. In this case the array *FT* will not contain the Fourier transform of the data on output.

PRECISION

A double precision version may be obtained by declaring all real variables to be double precision, by using double precision in the data statements and by using appropriate versions of the auxiliary routines *FORRT* and *REVRT* and of the functions *FLOAT*, *INT*, *ATAN*, *SQRT* and *EXP* where they appear.

TIMING

Once the discretization step has been performed, the running time of the algorithm is entirely independent of *NDT*, but depends only on *NFT*. For all except very large sample sizes, *NDT*, the discretization step will be extremely fast. The main burden is likely to be in the calls to the Fast Fourier transform routines; if *ICAL* is zero there will be one call each to *FORRT* and *REVRT* while if *ICAL* is non zero only *REVRT* is called. The effect of *WINDOW* on execution time is extremely marginal but because of the fine detail of the algorithm larger values will lead to very slightly faster calculation. Table 1 gives timings on a Honeywell twin processor level 68DPS machine for calls to *DENEST* with $ICAL = 0$ and $ICAL = 1$ using Monro's (1976) Fourier transform routines. For comparison, times using a routine calculating direct from (1) are included. The timings refer to samples generated from a standard normal density with $DLO = -4$, $DHI = 4$ and $WINDOW = 0{\cdot}5$. It is clear that the time increases approximately linearly with *NFT* and that the improvement over direct calculation from (1) is dramatic.

TABLE 1
Timings in seconds for first and subsequent calls to DENEST, and for calculation by direct application of the definition

NFT	*NDT*	*ICAL* = 0	*ICAL* = 1	*Direct*
64	20	0·031	0·015	0·14
	100	0·033	0·015	0·68
	1000	0·059	0·015	6·8
256	20	0·11	0·05	0·54
	100	0·11	0·05	2·7
	1000	0·13	0·05	27
1024	20	0·45	0·21	2·2
	100	0·46	0·21	11
	1000	0·47	0·21	110

ACCURACY

The accuracy of the routine is limited by two factors, the errors introduced by discretizing the data and those introduced by the use of values of the continuous Fourier transform of the kernel as its discrete Fourier transform, thus ignoring wrap-around and discretization effects. In order to minimize wrap-around effects, the user is advised to extend the range of calculation as described in *Numerical Method* above. An indication of the size of errors likely to be introduced is given in Table 2. Here the data are drawn from a standard normal distribution, and the calculations are performed with $DLO = -8$ and $DHI = 8$. The errors given are the maximum over the interval $(-4, 4)$ of the difference between the values output by the algorithm and the exact values of the estimate obtained by direct evaluation of (1). They do not refer to the difference between the estimated and true densities, which is much larger in all cases.

TABLE 2
Maximum calculation errors for estimates of a standard normal density based on 100 independent observations

NFT	*Window*	*Error*
64	0·2	$2{\cdot}5\times10^{-2}$
	0·6	$3{\cdot}8\times10^{-3}$
	1·2	$1{\cdot}4\times10^{-3}$
128	0·2	$8{\cdot}6\times10^{-3}$
	0·6	$1{\cdot}8\times10^{-3}$
	1·2	$5{\cdot}3\times10^{-4}$
256	0·2	$6{\cdot}7\times10^{-3}$
	0·6	$7{\cdot}8\times10^{-4}$
	1·2	$2{\cdot}5\times10^{-4}$
512	0·2	$3{\cdot}8\times10^{-3}$
	0·6	$1{\cdot}2\times10^{-4}$
	1·2	$2{\cdot}3\times10^{-5}$

REFERENCES

BONEVA, L. I., KENDALL, D. G. and STEFANOV, I. (1971). Spline transformations (with Discussion). *J. R. Statist. Soc.* B, **33**, 1–70.

DEHEUVELS, P. (1977). Estimation non paramétrique de la densité par histogrammes généralisés. *Rev. Stat. Appl.*, **25** (3), 5–43.

EPANECHNIKOV, V. A. (1969). Nonparametric estimation of a multivariate probability density. *Theory Prob. Applic.*, **14**, 153–158.

FRYER, M. J. (1977). A review of some non-parametric methods of density estimation. *J. Inst. Maths. Applics*, **20**, 335–354.

GENTLEMAN, W. M. and SANDE, G. (1966). Fast Fourier transforms—for fun and profit. In *AFIPS Proceedings of the Fall Joint Computer Conference*, **19**, 563–578.

HABBEMA, J. D. F., HERMANS, J. and VAN DER BROEK, K. (1974). A stepwise discriminant analysis program using density estimation. *COMPSTAT 1974*, Proceedings in Computational Statistics, pp. 101–110. Wien: Physica Verlag.

MONRO, D. M. (1976). Algorithm AS 97. Real discrete fast Fourier transform. *Appl. Statist.*, **25**, 166–172.

ROSENBLATT, M. (1956). Remarks on some non-parametric estimates of a density function. *Ann. Math. Statist.*, **27**, 832–837.

—— (1971). Curve estimates. *Ann. Math. Statist.*, **42**, 1815–1842.

SILVERMAN, B. W. (1978). Choosing a window width when estimating a density. *Biometrika*, **65**, 1–11.

—— (1981). Density estimation for univariate and bivariate data. In *Interpreting Multivariate Data* (V. Barnett, ed.), Chapter 3. Chichester: Wiley.

```
      SUBROUTINE DENEST(DT, NDT, DLO, DHI, WINDOW, FT, SMOOTH,
     *  NFT, ICAL, IFAULT)
      DIMENSION DT(NDT), FT(NFT), SMOOTH(NFT)
C
C        ALGORITHM AS 176  APPL. STATIST. (1982) VOL.31, NO.1
C
C        FIND DENSITY ESTIMATE BY KERNEL METHOD USING GAUSSIAN
C        KERNEL.  THE INTERVAL ON WHICH THE ESTIMATE IS EVALUATED
C        HAS END POINTS DLO AND DHI.  IF ICAL IS NOT ZERO
C        THEN IT IS ASSUMED THAT THE ROUTINE HAS BEEN
C        CALLED BEFORE WITH THE SAME DATA AND END POINTS
C        AND THAT THE ARRAY FT HAS NOT BEEN ALTERED.
C
      DATA ZERO, ONE, THIR2 /0.0E0, 1.0E0, 32.0E0/
      DATA BIG, KFTLO, KFTHI /30.0E0, 5, 11/
C
C        THE CONSTANT BIG IS SET SO THAT EXP(-BIG) CAN BE CALCULATED
C        WITHOUT CAUSING UNDERFLOW PROBLEMS AND CAN BE CONSIDERED
C        TO BE ZERO.
C
C        INITIALIZE AND CHECK FOR VALID PARAMETER VALUES
C
      IF (WINDOW .LE. ZERO) GOTO 92
      IF (DLO .GE. DHI) GOTO 93
      II = 2 ** KFTLO
      DO 1 K = KFTLO, KFTHI
      IF (II .EQ. NFT) GOTO 2
      II = II + II
    1 CONTINUE
      IFAULT = 1
      RETURN
    2 STEP = (DHI - DLO) / FLOAT(NFT)
      AINC = ONE / (FLOAT(NDT) * STEP)
      NFT2 = NFT / 2
      HW = WINDOW / STEP
      FAC1 = THIR2 * (ATAN(ONE) * HW / FLOAT(NFT)) ** 2
      IF (ICAL .NE. 0) GOTO 10
C
C        DISCRETIZE THE DATA
C
      DLO1 = DLO - STEP
      DO 3 J = 1, NFT
    3 FT(J) = ZERO
      DO 4 I = 1, NDT
      JJ = (DT(I) - DLO1) / STEP
      IF (JJ .GE. 1 .AND. JJ .LE. NFT) FT(JJ) = FT(JJ) + AINC
    4 CONTINUE
C
C        TRANSFORM TO FIND FT
C
      CALL FORRT(FT, NFT)
C
C        FIND TRANSFORM OF DENSITY ESTIMATE
C
   10 JHI = SQRT(BIG / FAC1)
      JMAX = MIN0(NFT2 - 1, JHI)
      SMOOTH(1) = FT(1)
      RJ = ZERO
      DO 11 J = 1, JMAX
      RJ = RJ + ONE
      FAC = EXP(-FAC1 * RJ * RJ)
      J1 = J + 1
      J2 = J1 + NFT2
      SMOOTH(J1) = FAC * FT(J1)
      SMOOTH(J2) = FAC * FT(J2)
   11 CONTINUE
C
C        COPE WITH UNDERFLOW BY SETTING TAIL OF TRANSFORM TO ZERO
C
      IF (JHI + 1 - NFT2) 21, 23, 20
   20 SMOOTH(NFT2 + 1) = EXP(-FAC1 * FLOAT(NFT2) ** 2) * FT(NFT2 + 1)
      GOTO 24
   21 J2LO = JHI + 2
      DO 22 J1 = J2LO, NFT2
      J2 = J1 + NFT2
      SMOOTH(J1) = ZERO
      SMOOTH(J2) = ZERO
   22 CONTINUE
   23 SMOOTH(NFT2 + 1) = ZERO
C
C        INVERT FOURIER TRANSFORM OF SMOOTH TO GET ESTIMATE
C        AND ELIMINATE NEGATIVE DENSITY VALUES
C
   24 CALL REVRT(SMOOTH, NFT)
      DO 25 J = 1, NFT
   25 IF (SMOOTH(J) .LT. ZERO) SMOOTH(J) = ZERO
      IFAULT = 0
      RETURN
   92 IFAULT = 2
      RETURN
   93 IFAULT = 3
      RETURN
      END
```

Remark AS R41

A Remark on Algorithm AS 126: Probability Integral of the Normal Range

By Mohamed el Lozy

Department of Nutrition, Harvard School of Public Health, Cambridge, Mass., USA

[Received October 1979. Revised October 1981]

A modification of AS 126, using a 24 rather than 16 point Gauss–Legendre formula (coded in double precision), was used to generate the values of $P(t \mid n)$ for $t = 0{\cdot}5\,(0{\cdot}5)\,11$ and $n = 2\,(2)\,100$ and the results compared with Harter's (1970) eight place tables. Almost perfect agreement was found, none of the differences amounting to more than two units in the eighth place. Single and double precision versions of Algorithm AS 126 were then compared with this algorithm for $t = 0\,(0{\cdot}05)\,10{\cdot}95$, $n = 2\,(2)\,100$. In spite of the short word length of the PDP 11/70 used (a 24 bit fraction) the single and double precision results were very similar. In single precision the maximum error was less than half a unit in the fifth decimal place for $n \leqslant 42$, in double precision for $n \leqslant 48$. The maximum error increased linearly with n, and for $n = 100$ was slightly more than one unit in the fifth place. Both accuracy and range of applicability of AS 126 are thus greater than originally reported.

Conversion of the algorithm to use the 24 point Gauss–Legendre formula requires redimensioning the G and H arrays to 12, modifying the *DATA* statements using half the values given for the abscissas (G) and weights (H) in Table 25.4 of Abramowitz and Stegun (1964), and changing the *DO* statement to read *DO* 10 $I = 1, 12$. In addition, on a short word length computer, the program (and *ALNORM*) should be converted to double precision. The double precision 24 point version takes about three times as long as the single precision 16 point version.

Acknowledgement

This work was supported by the Fund for Research and Teaching, Department of Nutrition, Harvard School of Public Health.

References

Abramowitz, M. and Stegun, I. A. (1964). *Handbook of Mathematical Functions*. Washington, D.C.: National Bureau of Standards. (Applied Mathematics Series, 55.)

Barnard, J. (1978). Algorithm AS 126. Probability integral of the normal range. *Appl. Statist.*, **27**, 197–198.

Harter, H. L. (1970). *Order Statistics and Their Use in Testing and Estimation*, Vol. 1. Washington, D.C., United States Government Printing Office.