```
C       S(.) AND C(.) NOW CONTAIN THE SUMS OF SINES AND
C       COSINES, RESPECTIVELY
C
C       NOW USE SPEC(.) FOR THE NORMALISED PERIODOGRAM
C
      CNORM = 2.0 / FN
      DO 7 I = 1, NF
    7 SPEC(I) = CNORM * (C(I) * C(I) + S(I) * S(I))
C
C       NOW OBTAIN THE SMOOTHED SPECTRAL ESTIMATES -
C       RETURN IF WANT ONLY THE PERIODOGRAM
C
      IF (NW .EQ. 1) RETURN
      KU = NF - NW + 1
      L = NW / 2
      DO 9 I = 1, KU
      SUM = 0.0
      J = I + NW - 1
      DO 8 K = I, J
    8 SUM = SUM + SPEC(K)
C
C       STORE THE SMOOTHED ESTIMATES AND THE CORRESPONDING
C       FREQUENCIES IN THE FIRST (NF - NW + 1) ELEMENTS OF
C       SPEC(.) AND FREQ(.)
C
      SPEC(I) = SUM / FW
      M = I + L
      FREQ(I) = FREQ(M)
    9 CONTINUE
      RETURN
   10 IFAULT = 1
      RETURN
   11 IFAULT = 2
      RETURN
   12 IFAULT = 3
      RETURN
   13 IFAULT = 4
      RETURN
   14 IFAULT = 5
      RETURN
   15 IFAULT = 6
      RETURN
      END
```

Algorithm AS 151

Spectral Estimates for Bivariate Counting Processes by Sectioning the Data

By David M. Charnock

Department of Social Sciences, W. A. Institute of Technology, Australia

Keywords: SPECTRUM; COHERENCE; PHASE; BIVARIATE COUNTING PROCESS; TIME-AVERAGE SMOOTHING

Language

ISO Fortran

Description and Purpose

Given a bivariate point process observed for a period of length T, in which N_j events occur in process j ($j = 1, 2$), the purpose of the subroutines *BIVCNT* and *SPLIT* is to compute estimates of the auto-spectra of the two processes and the squared coherence and phase spectra between the two processes (see, for example, Brillinger, 1972).

The major problem usually encountered in calculating such estimates is that the amount of computing time required is excessive, especially for processes containing large numbers of events. This is because the number of operations involved in calculating the sums of sines and cosines at all frequencies for process j is proportional to N_j^2. One well-known way to reduce the required computing time is to split the period of observation into several sections, compute estimates for each section and obtain smoothed estimates by averaging over the sections (see, for example, Lewis, 1970). If k sections are used, the number of operations for process j is then proportional to N_j^2/k.

This can give large savings in computation time and is the procedure implemented in *BIVCNT* and *SPLIT*. Subroutine *SPLIT* takes the event times in a univariate point process, divides the total period of observation into nonoverlapping sections of equal length and computes the event times relative to the sections. Subroutine *BIVCNT* calls *SPLIT* for both marginal processes in a bivariate point process and uses the resulting event times to compute smoothed estimates of the auto-spectra, coherence and phase spectra.

More precisely, suppose that k sections are used, that the event times relative to the sections in process j are $t'_j(r)$, $(j = 1, 2; r = 1, 2, ..., N_j)$, that the index number of the last event in section l of process j is $b_j(l)$ and that $n_j(l) = b_j(l) - b_j(l-1)$, i.e. $n_j(l)$ is the number of events in section l of process j $(j = 1, 2;\ l = 1, 2, ..., k)$.

Writing $\omega_p = 2\pi pk/T$, let

$$C_{jl}(\omega_p) = \sum_{r=b_j(l-1)+1}^{b_j(l)} \cos(\omega_p t'_j(r)),$$

$$S_{jl}(\omega_p) = \sum_{r=b_j(l-1)+1}^{b_j(l)} \sin(\omega_p t'_j(r)),$$

$$A(\omega_p) = \sum_{l=1}^{k} \{C_{1l}(\omega_p)\, C_{2l}(\omega_p) + S_{1l}(\omega_p)\, S_{2l}(\omega_p)\}/(n_1(l)\, n_2(l))^{\frac{1}{2}}$$

and

$$B(\omega_p) = \sum_{l=1}^{k} \{C_{2l}(\omega_p)\, S_{1l}(\omega_p) - C_{1l}(\omega_p)\, S_{2l}(\omega_p)\}/(n_1(l)\, n_2(l))^{\frac{1}{2}}.$$

By calling the subroutine *SCOUNT* for each section, *BIVCNT* computes the auto-spectral estimates

$$\hat{g}_j(\omega_p) = \frac{2}{k} \sum_{l=1}^{k} \frac{C^2{}_{jl}(\omega_p) + S^2{}_{jl}(\omega_p)}{n_j(l)}, \quad j = 1, 2,$$

the squared coherence estimate

$$\hat{\kappa}_{12}^2(\omega_p) = 4(A^2(\omega_p) + B^2(\omega_p))/(k^2\, \hat{g}_1(\omega_p)\hat{g}_2(\omega_p))$$

and the phase estimate

$$\hat{\Phi}_{12}(\omega_p) = \arctan(B(\omega_p)/A(\omega_p)),$$

each for $p = 1, 2, ..., NF$.

STRUCTURE

SUBROUTINE BIVCNT(N1, N2, TZERO, NSECT, T1, T2, NF, TTEMP, SPC1, SPC2, C1, S1, C2, S2, NT1, NT2, NN1, NN2, SPEC1, SPEC2, COHERE, PHASE, FREQ, IFAULT)

Formal parameters

N1	Integer	input : the number of events in process 1
N2	Integer	input : the number of events in process 2

TZERO	Real	input :	the length of the period of observation
NSECT	Integer	input :	the number of sections used
*T*1	Real array (*N*1)	input :	the event times in process 1
		output :	the event times in process 1 relative to the sections
*T*2	Real array (*N*2)	input :	the event times in process 2
		output :	the event times in process 2 relative to the sections
NF	Integer	input :	the number of frequencies at which the estimates are computed
TTEMP	Real array (*NF*)	workspace :	used in the calls to *SPLIT* and *SCOUNT*
*SPC*1	Real array (*NF*)	workspace :	used in the calls to *SCOUNT*
*SPC*2	Real array (*NF*)	workspace :	used in the calls to *SCOUNT*
*C*1	Real array (*NF*)	workspace :	used in the calls to *SCOUNT*
*S*1	Real array (*NF*)	workspace :	used in the calls to *SCOUNT*
*C*2	Real array (*NF*)	workspace :	used in the calls to *SCOUNT*
*S*2	Real array (*NF*)	workspace :	used in the calls to *SCOUNT*
*NT*1	Integer array (*NSECT*)	output :	the index numbers of the last event in each section of process 1
*NT*2	Integer array (*NSECT*)	output :	the index numbers of the last event in each section of process 2
*NN*1	Integer array (*NSECT*)	output :	the numbers of events in each section of process 1
*NN*2	Integer array (*NSECT*)	output :	the numbers of events in each section of process 2
*SPEC*1	Real array (*NF*)	output :	the smoothed estimates of the auto-spectrum of process 1
*SPEC*2	Real array (*NF*)	output :	the smoothed estimates of the auto-spectrum of process 2
COHERE	Real array (*NF*)	output :	the smoothed estimates of the squared coherence spectrum
PHASE	Real array (*NF*)	output :	the smoothed estimates of the phase spectrum
FREQ	Real array (*NF*)	output :	the frequencies at which the estimates are computed
IFAULT	Integer	output :	a fault indicator, equal to: 7 if $TZERO \leqslant 0$; 8 if $N1$ or $N2 \leqslant 0$; 9 if $T1\ (N1)$ or $T2\ (N2) > TZERO$; 10 if $NSECT \leqslant 0$; 11 if $NF < NSECT$; 12 if the number of events in any section of either process is greater than *NF*; values between 1 and 6 can result from errors in *SCOUNT*; 0 otherwise

SUBROUTINE SPLIT(*NSECT*, *NEVENT*, *TZERO*, *T*, *NT*, *TSECT*, *IFAULT*)

Formal parameters

NSECT	Integer	input :	the number of sections used
NEVENT	Integer	input :	the total number of events
TZERO	Real	input :	the length of the period of observation

T	Real array ($NEVENT$)	input : the event occurrence times output : the event times relative to the sections
NT	Integer array ($NSECT$)	output : the index numbers of the last event in each section
$TSECT$	Real array ($NSECT$)	output : the times at which the sections end
$IFAULT$	Integer	output : a fault indicator, equal to : 13 if $NSECT \leqslant 1$ or $> NEVENT$; 14 if $TZERO \leqslant 0$; 15 if $T(NEVENT) > TZERO$; 16 if any section contains no events

Auxiliary algorithm

BIVCNT uses the subroutine *SCOUNT* of algorithm AS 150 (Charnock, 1980).

Time and Accuracy

As already observed, the time required increases in proportion to the square of the numbers of events in the two processes but is inversely proportional to the number of sections used. It will also depend on the value chosen for the constant *NRECUR* in the auxiliary algorithm *SCOUNT*. Table 1 gives timings (on a DEC–10 computer) for *BIVCNT* with $NRECUR = 100$ when both processes contain the same number of events. With large values (around 100) of *NRECUR* the phase estimates at frequencies with very low coherency values have been found to be accurate to only one significant figure. However, this is not a serious problem because the phase estimates are of little use when the coherency is very low.

Table 1
Timings (in seconds) for BIVCNT on a DEC–10

	Number of events in each process			
Number of sections	208	1000	2000	4000
1	6·35	125·43	494·86	
5	1·61	26·17	100·39	396·24
10	1·44	13·72	51·66	199·85
20		7·48	26·89	101·13

Restriction

The size of the array *TTEMP* has been set equal to *NF*. This is a fairly arbitrary value : in fact, *TTEMP* must be large enough to store the event times section by section for each process. Since the user will normally set *NF* to a value somewhat greater than max ($N1$, $N2$)/$NSECT$, i.e. the mean number of events per section in the process with the larger number of events, this value for the size of *TTEMP* should be large enough for most analyses. If, however, the error condition $IFAULT = 12$ occurs then *TTEMP* should be *DIMENSION*ed to a larger size than *NF*.

Acknowledgements

I am grateful to the editor and referee for several suggestions which helped to improve the quality of the coding.

References

Brillinger, D. R. (1972). The spectral analysis of stationary interval functions. *Proc. Sixth Berkeley Symp. Math. Stat. and Prob.*, **1**, 485–513.

Charnock, D. M. (1980). Algorithm AS 150. Spectrum estimate for a counting process. *Appl. Statist.*, **29**, 211–214.

Lewis, P. A. W. (1970). Remarks on the theory, computation and application of the spectral analysis of series of events. *J. Sound and Vib.*, **12**, 353–375.

```
      SUBROUTINE BIVCNT(N1, N2, TZERO, NSECT, T1, T2, NF, TTEMP, SPC1,
     *  SPC2, C1, S1, C2, S2, NT1, NT2, NN1, NN2, SPEC1, SPEC2, COHERE,
     *  PHASE, FREQ, IFAULT)
C
C        ALGORITHM AS 151 APPL. STATIST. (1980) VOL.29, NO.2
C
C        CALCULATES SMOOTHED SPECTRAL ESTIMATES FOR A BIVARIATE
C        POINT PROCESS BY SPLITTING THE PERIOD OF OBSERVATION
C        INTO NONOVERLAPPING SECTIONS OF EQUAL LENGTH
C
      DIMENSION T1(N1), T2(N2), FREQ(NF), SPEC1(NF), SPEC2(NF),
     *  COHERE(NF), PHASE(NF), C1(NF), S1(NF), C2(NF), S2(NF),
     *  TTEMP(NF), SPC1(NF), SPC2(NF)
      INTEGER NT1(NSECT), NT2(NSECT), NN1(NSECT), NN2(NSECT)
C
C        TEST FOR PARAMETER ERRORS
C
      IF (TZERO .LE. 0.0) GOTO 9
      IF (N1 .LE. 0 .OR. N2 .LE. 0) GOTO 10
      IF (T1(N1) .GT. TZERO .OR. T2(N2) .GT. TZERO) GOTO 11
      IF (NSECT .LE. 0) GOTO 12
      IF (NF .LT. NSECT) GOTO 13
      IFAULT = 0
C
C        INITIALISE ARRAYS FOR ACCUMULATING SPECTRAL ESTIMATES
C
      DO 1 I = 1, NF
      SPEC1(I) = 0.0
      SPEC2(I) = 0.0
      COHERE(I) = 0.0
      PHASE(I) = 0.0
    1 CONTINUE
C
C        UNLESS NSECT = 1, CALL SPLIT TO SECTION THE PERIOD OF
C        OBSERVATION AND CALCULATE THE EVENT TIMES RELATIVE TO
C        THE SECTION ORIGINS
C
      NT1(1) = N1
      NT2(1) = N2
      NN1(1) = N1
      NN2(1) = N2
      IF (NSECT .EQ. 1) GOTO 3
      CALL SPLIT(NSECT, N1, TZERO, T1, NT1, TTEMP, IFAULT)
      IF (IFAULT .NE. 0) RETURN
      CALL SPLIT(NSECT, N2, TZERO, T2, NT2, TTEMP, IFAULT)
      IF (IFAULT .NE. 0) RETURN
C
C        CALCULATE NO. OF EVENTS IN EACH SECTION OF BOTH SERIES
C
      NN1(1) = NT1(1)
      NN2(1) = NT2(1)
C
C        TEST WHETHER THE WORKSPACE ARRAY TTEMP(.) IS LARGE ENOUGH
C        TO STORE THE EVENT TIMES IN SECTION 1 OF EACH SERIES
C
      IF (NN1(1) .GT. NF .OR. NN2(1) .GT. NF) GOTO 14
      DO 2 I = 2, NSECT
      NN1(I) = NT1(I) - NT1(I - 1)
      NN2(I) = NT2(I) - NT2(I - 1)
C
C        DO THE SAME FOR THE OTHER SECTIONS
C
      IF (NN1(I) .GT. NF .OR. NN2(I) .GT. NF) GOTO 14
    2 CONTINUE
C
C        CALL SCOUNT SECTION BY SECTION, USING TTEMP(.) TO STORE
C        THE EVENT TIMES TEMPORARILY
C
    3 FN = 1.0 / FLOAT(NSECT)
      TS = TZERO * FN
```

```
      KU1 = 0
      KU2 = 0
      DO 7 J = 1, NSECT
      KL1 = KU1 + 1
      KU1 = NT1(J)
      KL2 = KU2 + 1
      KU2 = NT2(J)
      DO 4 I = KL1, KU1
      K = I - KL1 + 1
      TTEMP(K) = T1(I)
    4 CONTINUE
C
C        CALL SCOUNT WITH NW = 1 TO OBTAIN THE NORMALISED PERIODOGRAM
C        AND THE SUM OF SINES AND COSINES FOR SECTION J OF SERIES 1
C
      CALL SCOUNT(NN1(J), TS, NF, 1, TTEMP, SPC1, FREQ, C1, S1, IFAULT)
      IF (IFAULT .NE. 0) RETURN
      DO 5 I = KL2, KU2
      K = I - KL2 + 1
      TTEMP(K) = T2(I)
    5 CONTINUE
C
C        DO THE SAME FOR SERIES 2
C
      CALL SCOUNT(NN2(J), TS, NF, 1, TTEMP, SPC2, FREQ, C2, S2, IFAULT)
      IF (IFAULT .NE. 0) RETURN
      SQNN = 1.0 / SQRT(FLOAT(NN1(J) * NN2(J)))
C
C        ACCUMULATE (OVER THE NSECT SECTIONS) THE SPECTRAL ESTIMATES
C        AT EACH FREQUENCY
C
      DO 6 I = 1, NF
      SPEC1(I) = SPEC1(I) + SPC1(I)
      SPEC2(I) = SPEC2(I) + SPC2(I)
      COHERE(I) = COHERE(I) + (C1(I) * C2(I) + S1(I) * S2(I)) * SQNN
      PHASE(I) = PHASE(I) + (C2(I) * S1(I) - C1(I) * S2(I)) * SQNN
    6 CONTINUE
    7 CONTINUE
C
C        NOW FIND THE SMOOTHED ESTIMATES
C
      DO 8 I = 1, NF
      SPEC1(I) = SPEC1(I) * FN
      SPEC2(I) = SPEC2(I) * FN
      TEMP = COHERE(I)
      COHERE(I) = 4.0 * FN * FN * (TEMP * TEMP + PHASE(I) * PHASE(I))
     *  / (SPEC1(I) * SPEC2(I))
      PHASE(I) = ATAN2(PHASE(I), TEMP)
    8 CONTINUE
      RETURN
    9 IFAULT = 7
      RETURN
   10 IFAULT = 8
      RETURN
   11 IFAULT = 9
      RETURN
   12 IFAULT = 10
      RETURN
   13 IFAULT = 11
      RETURN
   14 IFAULT = 12
      RETURN
      END
C
      SUBROUTINE SPLIT(NSECT, NEVENT, TZERO, T, NT, TSECT, IFAULT)
C
C        ALGORITHM AS 151.1 APPL. STATIST. (1980) VOL.29, NO.2
C
C        SPLITS A PERIOD OF OBSERVATION ON A POINT PROCESS INTO
C        NONOVERLAPPING SECTIONS OF EQUAL LENGTH AND COMPUTES
C        THE EVENT TIMES RELATIVE TO THE SECTION ORIGINS
C
```

```
C
      DIMENSION T(NEVENT), TSECT(NSECT)
      INTEGER NT(NSECT)
      IFAULT = 0
C
C        TEST FOR PARAMETER ERRORS
C
      IF (NSECT .LE. 1 .OR. NSECT .GT. NEVENT) GOTO 7
      IF (TZERO .LE. 0.0) GOTO 8
      IF (T(NEVENT) .GT. TZERO) GOTO 9
C
      INITIA = NEVENT / NSECT
      JJ = NSECT - 1
      FN = TZERO / FLOAT(NSECT)
      DO 4 I = 1, JJ
C
C        COMPUTE THE TIMES AT WHICH THE SECTIONS END
C
      TSECT(I) = FLOAT(I) * FN
C
C        AS A STARTING POINT, ASSUME THAT ALL SECTIONS
C        CONTAIN THE SAME NUMBER OF EVENTS
C
      NT(I) = I * INITIA
      INDEX = NT(I)
C
C        TEST WHETHER THE FINISHING POINT FOR SECTION I
C        IS TOO SMALL, JUST RIGHT OR TOO HIGH
C
      IF (T(INDEX) - TSECT(I)) 1, 4, 3
    1 NT(I) = NT(I) + 1
      INDEX = NT(I)
      IF (INDEX .GT. NEVENT) GOTO 2
      IF (T(INDEX) - TSECT(I)) 1, 4, 2
    2 NT(I) = NT(I) - 1
      GOTO 4
    3 NT(I) = NT(I) - 1
      INDEX = NT(I)
      IF (INDEX .LT. 1) GOTO 4
      IF (T(INDEX) .GT. TSECT(I)) GOTO 3
    4 CONTINUE
      NT(NSECT) = NEVENT
C
C        NT(I) IS NOW EQUAL TO THE TOTAL NUMBER OF EVENTS IN
C        THE FIRST I SECTIONS
C
C        NOW COMPUTE THE EVENT TIMES RELATIVE TO THE SECTION
C        ORIGINS - FIRST TEST WHETHER SECTION 1 CONTAINS NO EVENTS
C
      IF (NT(1) .EQ. 0) GOTO 10
      DO 6 I = 2, NSECT
      KK = I - 1
      LK = NT(KK) + 1
      LU = NT(I)
C
C        TEST WHETHER THE SECTION CONTAINS NO EVENTS
C
      IF (LK .GT. LU) GOTO 10
      DO 5 J = LK, LU
    5 T(J) = T(J) - TSECT(KK)
    6 CONTINUE
      RETURN
    7 IFAULT = 13
      RETURN
    8 IFAULT = 14
      RETURN
    9 IFAULT = 15
      RETURN
   10 IFAULT = 16
      RETURN
      END
```