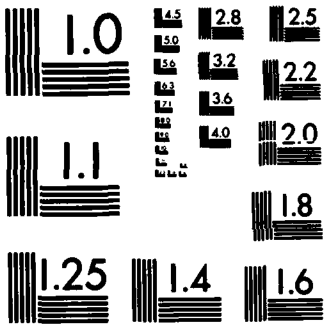
AD-A123 510   A FORTRAN PROGRAM FOR ISOTONIC REGRESSION IN TWO INDEPENDENT VARIABLES(U) IOWA UNIV IOWA CITY DEPT OF STATISTICS AND ACTURIAL SCIENCE G BRIL ET AL. JUL 82   1/1

UNCLASSIFIED   TR-86 N00014-80-C-0321   F/G 12/1   NL

END
FILMED
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

(12)

A FORTRAN PROGRAM FOR ISOTONIC REGRESSION IN TWO INDEPENDENT VARIABLES

Gordon Bril, Richard Dykstra, Carolyn Pillers and Tim Robertson

Department of Statistics and Actuarial Science
The University of Iowa

The University of Iowa

Department of Statistics and Actuarial Science

Technical Report No. 86

July, 1982

This work was supported by ONR Contract N00014-80-C-0321.

A FORTRAN PROGRAM FOR ISOTONIC REGRESSION IN TWO INDEPENDENT VARIABLES

Gordon Bril, Richard Dykstra, Carolyn Pillers and Tim Robertson

The University of Iowa

Key words: Order restricted inference; isotonic regression.

LANGUAGE

Fortran.

DESCRIPTION AND PURPOSES

There are a number of algorithms for calculating the least squares isotonic regression function and six of these algorithms are discussed in Section 2.3 of Barlow, Bartholomew, Bremner and Brunk (1972). The most widely used of these algorithms is the "pool adjacent violators algorithm" which is applicable only in the case of a simple linear ordering or an amalgamation of simple linear orderings. Cran (1980, Algorithm AS 149) gives a routine for computing the isotonic regression for the case of a simple linear ordering using the "Up and Down Blocks" algorithm. In many applications of isotonic regression we have more than one independent variable and the regression function is restricted to be monotone in each independent variable. Several of the algorithms given in Barlow et al. are applicable to this restriction but those which are reasonable to program often lead to excessive computation time. Dykstra and Robertson (1982) give an algorithm for computing the least squares regression function which is increasing in each of several variables. This algorithm uses successive one-dimensional smoothings and is very efficient, relative to other available algorithms. A Fortran IV program to implement this algorithm for two independent variables is given below.

STRUCTURE

SUBROUTINE SMOOTH (NROW, NCOL, X, W, NCYCLE, G, ICYCLE, IFAULT)

Formal Parameters

NROW	Integer	input: the number of rows
NCOL	Integer	input: the number of columns
X	Real Array (NROW,NCOL)	input: the original values
W	Real Array (NROW,NCOL)	input: the original weights, all zero or positive
NCYCLE	Integer	input: the maximum number of cycles if the array does not converge
GCHECK	Real Array (NROW,NCOL)	workspace
WT	Real Array (NROW,NCOL)	workspace
G	Real Array (NROW,NCOL)	output: the smoothed values if convergence was attained, or the values after the maximum number of iterations
ICYCLE	Integer	output: the actual number of cycles the smoothing process required

IFAULT Integer output: a fault indicator, equal to

1 if convergence was not attained in the specified number of cycles

2 if any element of W is negative

3 if rows and/or columns exceed 20

4 if a zero weight was replaced by 0.00001

0 otherwise

Auxiliary Algorithm

SMOOTH uses the subroutine PAV (K,X,IORDER,W,FINALX) to apply the pool adjacent violators algorithm for the isotonic regression on rows and columns.

Constants

The constant EPS is used to determine if convergence has been attained. If each entry of the array after the current smoothing does not differ from the corresponding entry from the previous smoothing by more than EPS, the array is considered to be smoothed.

Restrictions

The X and W arrays cannot have more than 20 rows and/or columns. However, if one wished to smooth a larger array, the changes to the subroutine would be minimal.

The W array cannot contain any negative values. If zero values are encountered, they are replaced by 10^{-5} for the amalgamation.

Time and Accuracy

For the 9×9 array in Table 4 in Dykstra and Robertson (1982), convergence was attained in 47 iterations at a CPU time of .26 seconds.

The constant EPS specifies the change from one cycle to the next for convergence. It has been given an initial value of 10^{-5}; however if greater or lesser accuracy is required, the value may be changed.

Related Algorithms

Cran (1980) gives an algorithm (AS 149) for amalgamation of the means in the case of simple ordering, which can be used as an alternative to SUBROUTINE PAV.

REFERENCES

Dykstra, Richard L. and Robertson, Tim (1982). An algorithm for isotonic regression for two or more independent variables. Ann. Statist. 10 (to appear).

Cran, G.W. (1980). Amalgamation of means in the case of simple ordering. Applied Statistics, 29, 209-211.

Barlow, R.E., Bartholomew, D.J., Bremner, J.M. and Brunk, H.D. (1972). Statistical Inference Under Order Restrictions. John Wiley and Sons, New York.

```
C
C         DRIVER PROGRAM FOR SMOOTH SUBROUTINE.
C
      DIMENSION X(20,20), WT(20,20), G(20,20)
      READ( 5, 10 ) NROW, NCOL, NCYCLE
10    FORMAT( 2I2, I5 )
      DO 30 I = 1, NROW
      READ( 5, 20 ) ( X(I,J), J = 1, NCOL )
20    FORMAT( 10F8.4 )
30    CONTINUE
      DO 40 I = 1, NROW
      READ( 5, 20 ) ( WT(I,J), J = 1, NCOL )
40    CONTINUE
C
C         ECHO CHECK THE INPUT ARRAYS.
C
      WRITE( 6, 50 )
50    FORMAT( 1H1 )
      WRITE( 6, 60 )
60    FORMAT( 4X, 34HTHE ORIGINAL MATRIX TO BE SMOOTHED,// )
      DO 80 I = 1, NROW
      WRITE( 6, 70 ) ( X(I,J), J = 1, NCOL )
70    FORMAT( 4X, 10F12.6 )
80    CONTINUE
      WRITE( 6, 90 )
90    FORMAT( ///4X, 28HTHE ASSOCIATED WEIGHT MATRIX, // )
      DO 100 I = 1, NROW
      WRITE( 6, 70 ) ( WT(I,J), J = 1, NCCL )
100   CONTINUE
C
C         SMOOTH THE X ARRAY.
C
      CALL SMOOTH( NROW, NCOL, X, WT, NCYCLE, ICYCLE, IFAULT, G )
C
C         CHECK FOR FAULT ERRORS IN THE SUBROUTINE.
C
      IF ( IFAULT .EQ. 2 ) GOTO 160
      IF ( IFAULT .EQ. 3 ) GOTO 180
      IF ( IFAULT .EQ. 4 ) WRITE( 6, 110 )
110   FORMAT( //4X, 31HZERO WEIGHTS HAVE BEEN REPLACED,/,
     *4X, 24HBY 0.00001 FOR SMOOTHING )
C
C         OUTPUT THE SMOOTHED MATRIX.
C
      WRITE( 6, 50 )
      WRITE( 6, 120 ) ICYCLE
120   FORMAT( //4X, 35HTHE NUMBER OF CYCLES COMPLETED WAS, I5 )
      IF ( IFAULT .EQ. 1 ) WRITE( 6, 130 )
130   FORMAT( 4X, 43HTHE ARRAY DID NOT CONVERGE IN THE SPECIFIED, )
     */4X, 16HNUMBER OF CYCLES )
      WRITE( 6, 140 )
140   FORMAT( //4X, 19HTHE SMOOTHED MATRIX, // )
      DO 150 I = 1, NROW
      WRITE( 6, 70 ) ( G(I,J), J = 1, NCOL )
150   CONTINUE
      STOP
160   WRITE( 6, 170 )
170   FORMAT( //4X, 31HAT LEAST ONE WEIGHT IS NEGATIVE )
      STOP
180   WRITE( 6, 190 )
```

```
190   FORMAT( //4X, 29HROWS AND/OR COLUMNS EXCEED 20 )
      STOP
      END
```

```
      SUBROUTINE SMOOTH( NROW, NCOL, X, W, NCYCLE, G, ICYCLE, IFAULT )
C
C        SUBROUTINE TO ORDER A 2 DIMENSIONAL ARRAY USING
C        AN ALGORITHM OF DYKSTRA AND ROBERTSON( 1982 ).
C        THE ORDERING IS DONE SO THAT THE REGRESSION
C        FUNCTION IS INCREASING IN EACH INDEPENDENT VARIABLE.
C
      DIMENSION X(20,20), WT(20,20), G(20,20), C(20,20)
      DIMENSION R(20,20), ROWS(20), COLS(20), WEIGHT(20)
      DIMENSION ORDER(20), GCHECK(20,20), W(20,20)
      DATA EPS/ 1.0E-06/
      IFAULT = 0
C        CHECK THAT ROWS AND COLUMNS DO NOT EXCEED 20.
C
      IF ( NROW .GT. 20 .OR. NCOL .GT. 20 ) GOTO 120
C
C        CHECK THAT WEIGHTS ARE POSITIVE, OR ZERO.
C
      DO 5 I = 1, NROW
      DO 5 J = 1, NCOL
      IF ( W( I, J ) .LT. 0.0 ) GOTO 110
      WT(I,J) = W(I,J)
      IF( WT(I,J) .GE. EPS ) GOTO 5
      WT(I,J) = 0.00001
      IFAULT = 4
    5 CONTINUE
C
C        INITIALIZE R AND C TO ZERO, AND SET UP WORKSPACE.
C
      DO 20 I = 1, NROW
      DO 10 J = 1, NCOL
      G( I, J ) = X( I, J )
      GCHECK( I, J ) = X( I, J )
      C( I, J ) = 0.0
      R( I, J ) = 0.0
   10 CONTINUE
   20 CONTINUE
C
C        INITIALIZE COUNTER FOR NUMBER OF CYCLES.
C
      ICOUNT = 0
C
C        SMOOTH OVER ROWS.
C
   25 JCOUNT = 0
      DO 50 I = 1, NROW
      DO 30 J = 1, NCOL
      ROWS( J ) = G( I, J ) - R( I, J )
      WEIGHT( J ) = WT( I, J )
   30 CONTINUE
C
      CALL PAV( NCOL, ROWS, 1, WEIGHT, ORDER )
C
      KCOUNT = 0
      DO 40 J = 1, NCOL
      R( I, J ) = ORDER( J ) - ROWS( J )
      G( I, J ) = ORDER( J )
      IF ( ABS( G( I, J ) - GCHECK( I, J ) ) .LT. EPS )
     * KCOUNT = KCOUNT + 1
      GCHECK( I, J ) = G( I, J )
```

```
40    CONTINUE
C
C        DETERMINE IF THERE IS NO CHANGE IN THE ITH ROW
C        FROM THE PREVIOUS ITERATION.
C
      IF ( KCOUNT .EQ. NCOL ) JCOUNT = JCOUNT + 1
50    CONTINUE
      ICOUNT = ICOUNT + 1
      IF ( ICOUNT .EQ. 1 ) GOTO 55
C
C        DETERMINE IF THERE HAS BEEN NO CHANGE IN ALL
C        ROWS FROM THE PREVIOUS ITERATION.
C
      IF ( JCOUNT .EQ. NROW ) GOTO 90
C
C
C        SMOOTH OVER COLUMNS.
C
55    LCOUNT = 0
      DO 80 J = 1, NCOL
      DO 60 I = 1, NROW
      COLS( I ) = G( I, J ) - C( I, J )
      WEIGHT( I ) = WT( I, J )
60    CONTINUE
C
      CALL PAV( NROW, COLS, 1, WEIGHT, ORDER )
C
      MCOUNT = 0
      DO 70 I = 1, NROW
      C( I, J ) = ORDER( I ) - COLS( I )
      G( I, J ) = ORDER( I )
      IF ( ABS( G( I, J ) - GCHECK( I, J ) ) .LT. EPS )
     * MCOUNT = MCOUNT + 1
      GCHECK( I, J ) = G( I, J )
70    CONTINUE
C
C        DETERMINE IF THERE IS NO CHANGE IN THE ITH COLUMN
C        FROM THE PREVIOUS ITERATION.
C
      IF ( MCOUNT .EQ. NROW ) LCOUNT = LCOUNT + 1
80    CONTINUE
      ICOUNT = ICOUNT + 1
C
C        DETERMINE IF THERE HAS BEEN NO CHANGE IN ALL COLUMNS
C        FROM THE PREVIOUS ITERATION.
C
      IF ( LCOUNT .EQ. NCOL ) GOTO 90
C
C        CHECK IF NUMBER OF CYCLES HAS BEEN REACHED.
C
      IF ( NCYCLE .EQ. ICOUNT ) GOTO 100
      GOTO 25
90    ICYCLE = ICOUNT
      RETURN
100   ICYCLE = ICOUNT
      IFAULT = 1
      RETURN
110   IFAULT = 2
      RETURN
120   IFAULT = 3
```

```
      RETURN
      END
      SUBROUTINE PAV( K, X, IORDER, W, FINALX )
C
C        SUBROUTINE TO APPLY POOL ADJACENT VIOLATORS THEOREM.
C
      DIMENSION FX(20), WT(20), PW(20), X(20), W(20),FINALX(20)
      DIMENSION W1(20)
      DATA EPS / 1.0E-06 /
C
C        SET UP WORKSPACE.
C
      DO 10 I = 1, K
      FX(I) = X(I)
      IF( IORDER .EQ. 0 ) FX(I) = -FX(I)
      WT(I) = W(I)
      PW(I) = WT(I) * FX(I)
      W1(I) = W(I)
10    CONTINUE
      IBEL = K - 1
20    I = 0
30    I = I + 1
35    IF(  I .GT. IBEL ) GOTO 50
      I1 = I + 1
C
C        DETERMINE IF POOLING IS REQUIRED.
C
      IF( ( FX(I) - FX(I1) ) .LE. EPS ) GOTO 30
C
C        POOL THE ADJACENT VALUES.
C
      PW(I) = PW(I) + PW(I1)
      W1(I) = W1(I) + W1(I1)
      FX(I) = PW(I) / W1(I)
      IP = I + 1
      IF( IP .GT. IBEL ) GOTO 45
      DO 40 J = IP, IBEL
      J1 = J + 1
      PW(J) = PW(J1)
      W1(J) = W1(J1)
      FX(J) = FX(J1)
40    CONTINUE
45    IBEL = IBEL - 1
      GOTO 35
50    IBEL1 = IBEL
      ICOUNT = 0
      IF( IBEL1 .LE. 0 ) GOTO 70
C
C        DETERMINE IF ALL VALUES ARE ORDERED.
C
      DO 60 L = 1, IBEL1
      L1 = L + 1
      IF ( ( FX(L) - FX(L1) ) .LE. EPS ) ICOUNT = ICOUNT + 1
60    CONTINUE
      IF( ICOUNT .NE. IBEL1 ) GOTO 20
C
C        RECOVER FINAL ORDERED VALUES.
C
70    FINALX(1) = FX(1)
      I = 1
```

```
      J = 1
      L = 2
80    IF( L .GT. K ) GOTO 100
      WEIGHT = WT(L) + WT(I)
      IF( WEIGHT .GT. W1(J) ) GOTO 90
      FINALX(L) = FX(J)
      WT(I) = WT(I) + WT(L)
      L = L + 1
      GOTO 80
90    J1 = J + 1
      FINALX(L) = FX(J1)
      I = L
      J = J + 1
      L = L + 1
      GOTO 80
100   IF( IORDER .EQ. 1 ) RETURN
      DO 110 I = 1,K
      FINALX(I) = -FINALX(I)
110   CONTINUE
      RETURN
      END
```

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 11	2. GOVT ACCESSION NO. AD-A113 520	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Fortran Program for Isotonic Regression in Two Independent Variables.		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gordon Bril, Richard Dykstra, Carolyn Pillers, and Tim Robertson		8. CONTRACT OR GRANT NUMBER(s) N000-80-C-0321
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Statistics and Actuarial Science The University of Iowa		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, VA		12. REPORT DATE July 1981
		13. NUMBER OF PAGES 10
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Order restricted inference, isotonic regression.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A Fortran program to implement the algorithm developed by Dykstra and Robertson (Ann. Statist. (1982)) for computing the regression which is required to be increasing in each of two independent variables.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

2-8

DT