A Computer Program for Variable Selection in Discriminant Analysis

by

George P. McCabe, Jr.* and Richard J. Pohl*

Purdue University
Department of Statistics
Division of Mathematical Sciences
Mimeograph Series No. 334
October 10, 1973

*Research supported in part by National Science Foundation Grant GJ30269.

I. Introduction

This document is a supplement to "Computations for Variable Selection in Discriminant Analysis" by George McCabe, Technometrics 17, 103-109, (February 1975). It is assumed that the reader is familiar with the scope and notation of this paper. What follows is a description of the algorithm, the array structures, and the setup of the data.

II. Algorithm

Let $X_{i\ell n}$ denote the observation on the i-th variable for the n-th person in group ℓ where $i=1,\ldots,p, \ell=1,\ldots,k$ and $n=1,\ldots,N_\ell$.

The main program calculates the within groups cross product matrix and the total cross product matrix, denoted by W and T, respectively. Thus,

$$w_{ij} = \sum_{\ell=1}^{k} \sum_{n=1}^{N_\ell} (X_{i\ell n} - X_{i\ell.})(X_{j\ell n} - X_{j\ell.})$$

$$= \sum_{\ell=1}^{k} \sum_{n=1}^{N_\ell} X_{i\ell n} X_{j\ell n} - \sum_{\ell=1}^{k} N_\ell X_{i\ell.} X_{j\ell.}$$

and

$$t_{ij} = \sum_{\ell=1}^{k} \sum_{n=1}^{N_\ell} (X_{i\ell n} - X_{i..})(X_{j\ell n} - X_{j..})$$

$$= \sum_{\ell=1}^{k} \sum_{n=1}^{N_\ell} X_{i\ell n} X_{j\ell n} - NX_{i..} X_{j..}$$

where the dot notation is used to indicate averaging over that index and

$$N = \sum_{\ell=1}^{k} N_\ell.$$

For a subset of variables of size m, $X_{i_1}, X_{i_2}, \ldots, X_{i_m}$, let U be the ratio of the within groups generalized variance to the total generalized variance for that subset. The subroutine DISC computes the U value for each of the 2^p-1 subsets of the variables $X_1, \ldots, X_p$. The algorithm first determines the U value for a single variable X_i and then, in successive steps, computes the U values for all combinations of $X_1, \ldots, X_{i-1}$ with X_i. The algorithm operates efficiently by using previously calculated results. For example, if the U value for X_5, X_3, X_2 has already been calculated then the U value for X_5, X_3, X_2, X_1 can be calculated by U*(within groups variance of $X_1|X_5,X_3,X_2$)/(total variance of $X_1|X_5,X_3,X_2$).

The U value for X_i (call it U_i) is simply w_{ii}/t_{ii} where w_{ii} and t_{ii} were defined previously. By Gaussian elimination, the i^{-th} row is swept out of the W and T matrices leaving matrices of size i-1 (but stored in different arrays than W and T), say W_{i-1} and T_{i-1}. Then the U values for $X_i,X_{i-1};X_i,X_{i-2};\ldots;X_i,X_1$ are $U_i{*}(W_{i-1})_{i-1,i-1}/(T_{i-1})_{i-1,i-1}$, $U_i{*}(W_{i-1})_{i-2,i-2}/(T_{i-1})_{i-2,i-2},\ldots,$ $U_i{*}(W_{i-1})_{1,1}/(T_{i-1})_{1,1}$, respectively. The process continues until the U value for $X_1,X_2,\ldots,X_{i-1},X_i$ has been calculated and then the process starts over again with X_{i+1}. The U values are calculated for the combinations of $X_1,\ldots,X_i$ in the following order; $X_i;X_i,X_1;X_i,X_2;X_i,X_2,X_1;X_i,X_3;X_i,X_3,X_1;X_i,X_3,X_2;X_i,X_3,X_2,X_1;\ldots;X_i\ldots X_2,X_1$; for i=1,...,p.

For these calculations, the algorithm uses two three dimensional arrays of size (K+1)*(K+1)*(K+1). The original matrices W and T are stored in the (K+1) -st block of those matrices and the results of Gaussian elimination sweeps are stored in the lower blocks. It should be noted that a block is not written over until the information that is written in that block will not be needed for later calculation.

In what follows W and T refer to the 3-dimensional arrays. W(I,J,K) refers to the I th block, J th row, and K th column.

Let NT be an integer array of size p+1 defined as

$$NT(i) = \begin{cases} 0 & \text{if variable } i \text{ is \underline{not} in the discriminant equation} \\ 1 & \text{if variable } i \text{ is in the discriminant equation.} \end{cases}$$

Note that there are 2^{p-1} possible combinations of zeros and ones (not counting all zeros). Initially NT(i)=0 i=1,...,p and NT(p+1)=1. The array NT is altered by the following code:

```
      DO 20 I=1,K
      IF(NT(I).EQ.0) GO TO 21
   20 CONTINUE
      GO TO ...
   21 NT(I)=1
      .
      .
      .
```

Then U values will be calculated for the subset $X_1;X_2;X_2,X_1;X_3;X_3,X_1;X_3,X_2;X_3,X_2,X_1;X_4;X_4,X_1;\dots$ in that order. The value of I when the program branches out of the loop is the index of the variable to be added to the discriminant equation.

Let J=min{L: NT(L)=1, L=i+1,...,K+1}. The block J of W(T) is called the source block. If J=K+1 then X_I is the only variable in the discriminant equation and U=W(J,I,I)/T(J,I,I). The matricies W and T are swept along the column I leaving an (I-1)(I-1) matrix which is stored in block I of the W and T arrays. (Of course if (I=1) there is nothing to be swept so the elimination is not performed).

If J $\leq$ K then J represents the index of the last variable added to the equation. The (j-1)x(j-1) matrix in block J and the U value calculated when it was formed are used to calculate the new U value: U = (the old U value) *W(J,I,I)/T(J,I,I). The new U value is stored and if I>1 the matrix in block J is swept along the I th column and the resulting (I-1)(I-1) matrix is stored in block I of the W and T arrays.

At each step the calculated U is compared with the previously calculated U's for that subset size. The ten smallest U's for each subset size are stored along with a record of which variables were in that equation.

To determine the subset size when the variable I is added, the algorithm uses the fact that the subset size after I has been added, is one greater than the subset size before I has been added. Let NX be an array of size K+1. Set NX(K+1)=0. Then the subset size is calculated by NX(I)=NX(J)+1. If J=K+1 then I is the only value in the equation and NX(I)=NX(J)+1=1. If J $\leq$ K then NX(J) is the number of variables in the equation when block J was formed. A linear array SAVE of size K is used to store U values for use in later calculations.

III. Array Structure

Regarding the 3 dimensional arrays, three facts should be noted: 1) the 1st block is never used, 2) the Ith block contains an (I-1)x(I-1) matrix, 3) the matrix stored in each block is symmetric.

Capitalizing on the first point is simple. Let the array be of size KxKxK and instead of using J to index a block, J-1 is used.

To capitalize on points two and three, the three dimensional array structure is abandoned and simple linear arrays are used. The maximum value K may attain has been set at 20. A symmetric matrix can be stored in the following order: A_{11} A_{21} A_{22} A_{31} A_{32} A_{33}, A_{41},.... Call the linear array B. To reference A_{ij}, $(i > j)$ calculate $\ell=i*(i-1)/2+j$. The term a_{ij} is stored in B_ℓ. To avoid calculating i*(i-1)/2 each time an array INDROW is set up with INDROW(I)= I*(I-1)/2. Then, the (i,j)th element (i>j) is in the INDROW(I)+J position of the linear array B. To arrange successive symmetric matricies of increasing order define a linear array, INDBLK(I) = $\sum_{j=1}^{I}$ INDROW(J). Previously W(I,J,K) was referenced. For I>J$\geq$K, L=INDBLK(I-1)+INDROW(J)+K. W(L) then corresponds to W(I,J,K).

Use of these arrays saves several thousand words of storage, and a significant amount of CPU time.

IV. Input

Order of Cards for Input

1) Problem Card

2) Groups Card(s)

3) Format Card

4) Data

Preparation of Input Cards

1) Problem Card.

Col.	Information (all fields right justified)
1-5	Number of Variables (maximum of 20)
6-10	Number of Groups (maximum of 20)

2) Groups Card(s)

Col.	Information (all fields right justified)
1-5	Number of observations for the first group
6-10	" " " " " second group
11-15	" " " " " third group
etc.	

If necessary the number of observations per group is extended to a second card.

3) Format Card

The format card describes the format for the subsequent data cards.

4) Data

The data cards follow the format card.

All card input is read from logical tape 5 (i.e. READ (5,))

All output is written on logical tape 6 (i.e. WRITE (6,))

V. <u>Example</u> (see McCabe, 1975 for details)

Input Description:

1) The problem card indicates that there are

    9 variables

    12 groups

2) The GROUPS card indicates that each group has 4 cards.

3) The format card indicates that the data is in format

    (9F8.3)

Output Description:

A heading tells the subset size for the U values.

The U values are listed with the variables printed to the right of the U value.

VI. Input for example

```
   9     12
   4     4     4     4     4     4     4     4     4     4     4     4
(9F8.3)
5.40    0.188   0.92    215.    16.35   7.65    0.72    1.14    1.09
5.65    0.165   1.04    208.    12.25   5.15    0.71    0.94    1.35
5.14    0.260   0.95    300.    13.02   5.68    0.68    0.60    1.41
5.14    0.169   1.10    248.    11.92   7.88    1.09    1.01    1.64
5.14    0.164   1.12    174.    14.17   8.12    0.70    2.17    1.85
5.10    0.094   1.22    129.    8.55    6.92    0.81    2.67    3.18
4.70    0.100   1.52    117.    8.74    8.16    0.39    3.32    4.16
4.46    0.112   1.47    170.    9.49    9.16    0.70    3.76    5.14
4.37    0.112   1.07    121.    8.85    10.35   0.74    5.74    5.73
4.39    0.058   1.54    115.    4.73    6.91    0.77    5.85    6.45
4.17    0.078   1.26    112.    6.29    7.95    0.26    5.30    8.37
3.89    0.070   1.42    117.    6.61    9.76    0.41    8.30    9.21
3.88    0.077   1.25    127.    6.41    10.96   0.56    9.67    10.64
4.07    0.046   1.54    91.     3.82    6.61    0.50    7.67    10.07
3.88    0.055   1.53    91.     4.98    8.00    0.23    8.78    11.26
3.74    0.053   1.40    79.     5.86    10.14   0.41    11.04   12.15
5.11    0.247   0.94    261.    13.25   7.55    0.61    1.86    2.61
5.46    0.298   0.96    300.    12.30   7.50    0.68    2.00    1.98
5.61    0.145   1.10    242.    9.66    6.76    0.63    1.01    0.76
5.85    0.186   1.20    229.    13.78   7.12    0.62    3.09    2.85
4.57    0.102   1.37    156.    8.58    9.92    0.63    3.67    3.24
5.11    0.097   1.30    139.    8.58    9.69    0.42    4.70    4.63
4.78    0.122   1.30    214.    8.22    7.75    0.32    3.07    3.67
6.67    0.083   1.42    132.    12.68   9.56    0.55    8.30    8.10
3.96    0.059   1.53    98.     4.80    10.00   0.36    6.52    7.72
4.00    0.050   1.50    115.    5.06    8.31    0.28    7.91    9.78
4.12    0.086   1.55    148.    6.16    7.58    0.16    6.39    9.07
4.99    0.048   1.46    97.     7.49    9.38    0.40    9.70    9.13
3.80    0.049   1.48    108.    3.82    8.80    0.24    9.57    11.57
3.96    0.036   1.28    103.    4.78    7.29    0.24    9.67    11.42
3.93    0.048   1.42    109.    4.93    7.47    0.14    9.65    13.32
4.02    0.039   1.51    100.    5.66    8.84    0.37    10.54   11.57
5.24    0.194   1.00    445.    12.27   6.27    0.72    1.02    0.75
5.20    0.256   0.78    380.    11.39   7.55    0.78    1.63    2.20
5.30    0.136   1.00    259.    9.96    8.08    0.45    1.97    2.27
5.67    0.127   1.13    248.    9.12    7.04    0.55    1.43    0.67
4.46    0.087   1.24    276.    7.24    9.40    0.43    4.17    5.08
4.91    0.092   1.47    158.    7.37    10.57   0.59    5.07    6.37
4.79    0.047   1.46    121.    6.99    9.91    0.30    5.15    6.82
5.36    0.095   1.26    195.    8.59    8.66    0.48    4.17    3.65
3.94    0.054   1.60    148.    4.85    9.62    0.18    7.20    10.14
4.52    0.051   1.53    115.    6.34    9.78    0.34    9.52    9.74
4.35    0.032   1.55    82.     5.99    9.73    0.22    7.02    8.60
4.64    0.065   1.46    152.    4.43    10.54   0.22    7.61    9.09
3.82    0.038   1.48    105.    4.65    9.85    0.18    10.15   12.26
4.24    0.035   1.47    100.    4.56    8.95    0.33    10.51   11.29
4.22    0.030   1.56    97.     5.29    8.37    0.14    8.27    9.51
4.41    0.058   1.58    130.    4.58    9.46    0.14    3.28    12.69
```

VII. Output for example

```
                    U RATIO FOR  1 VARIABLES


                     7.2345315E-02    9
                     8.9181211E-02    8
                     1.7266080E-01    5
                     1.8725929E-01    2
                     2.0156292E-01    4
                     2.3980439E-01    3
                     2.5829377E-01    1
                     3.1086599E-01    7
                     4.7315793E-01    6




                    U RATIO FOR  2 VARIABLES


                     3.3930127E-02    4 9
                     3.5562455E-02    5 8
                     3.5687590E-02    1 8
                     3.8081884E-02    2 9
                     3.8384953E-02    3 9
                     4.0338730E-02    5 9
                     4.1230290E-02    4 8
                     4.3164579E-02    6 9
                     4.4323291E-02    1 9
                     4.6908955E-02    3 8




                    U RATIO FOR  3 VARIABLES


                     1.5184513E-02    2 4 9
                     1.5847169E-02    4 5 8
                     1.5945922E-02    1 4 8
                     1.6081403E-02    1 6 8
                     1.8206408E-02    3 5 8
                     1.8609446E-02    2 4 8
                     1.8833618E-02    5 6 8
                     1.8850610E-02    4 5 9
                     1.8960943E-02    1 4 9
                     1.9223622E-02    1 3 8




                    U RATIO FOR  4 VARIABLES
```

```
   7.1751422E-03    1  4  6  8
   7.3213055E-03    1  2  4  8
   7.9626272E-03    1  3  6  8
   7.9760707E-03    1  4  5  8
   8.1480903E-03    1  5  6  8
   8.4735913E-03    4  5  6  8
   8.5122332E-03    1  2  4  9
   8.6691979E-03    1  2  6  8
   8.8027612E-03    1  6  7  8
   8.9712949E-03    2  4  6  9



U RATIO FOR  5 VARIABLES

   3.0771196E-03    1  2  4  6  8
   3.0860162E-03    1  4  5  6  8
   3.5829950E-03    1  4  6  7  8
   3.9116644E-03    1  3  4  6  8
   4.2326824E-03    1  3  5  6  8
   4.4827129E-03    1  2  4  7  8
   4.7653663E-03    1  3  6  7  8
   4.7701877E-03    2  4  5  6  8
   4.8261482E-03    1  2  4  6  9
   4.9250382E-03    1  2  4  5  8



U RATIO FOR  6 VARIABLES

   1.6623158E-03    1  2  4  6  7  8
   1.7070958E-03    1  4  5  6  7  8
   1.9241861E-03    1  2  3  4  6  8
   1.9337476E-03    1  2  4  5  6  8
   1.9638642E-03    1  3  4  5  6  8
   2.2032667E-03    1  3  4  6  7  8
   2.5667227E-03    1  2  4  6  7  9
   2.5880196E-03    1  2  4  6  8  9
   2.5974259E-03    1  4  5  6  7  9
   2.6640775E-03    1  3  5  6  7  8



U RATIO FOR  7 VARIABLES

   1.0768456E-03    1  2  4  5  6  7  8
   1.0974227E-03    1  2  3  4  6  7  8
   1.1278036E-03    1  3  4  5  6  7  8
   1.2175837E-03    1  2  3  4  5  6  8
   1.4130605E-03    1  4  5  6  7  8  9
   1.4391947E-03    1  2  4  6  7  8  9
```

```
1.5438808E-03    1 2 3 4 6 7 9
1.6423574E-03    1 2 3 4 6 8 9
1.6593020E-03    1 2 4 5 6 8 9
1.6635310E-03    1 3 4 5 6 7 9


U RATIO FOR  8 VARIABLES

6.9867247E-04    1 2 3 4 5 6 7 8
8.8611364E-04    1 2 4 5 6 7 8 9
9.0796237E-04    1 2 3 4 6 7 8 9
9.2236832E-04    1 3 4 5 6 7 8 9
1.0070857E-03    1 2 3 4 5 6 7 9
1.0509513E-03    1 2 3 4 5 6 8 9
1.3068148E-03    2 3 4 5 6 7 8 9
1.4530880E-03    1 2 3 5 6 7 8 9
1.5939784E-03    1 2 3 4 5 7 8 9


U RATIO FOR  9 VARIABLES

5.6105433E-04    1 2 3 4 5 6 7 8 9
```

VIII. Program Listing

```
      PROGRAM DISCRM(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION W(1540),T(1540),INDBLK(20),INDROW(20)
      DIMENSION XMSUM(20, 20),XSUM(20),NM(20),X(20)
      DIMENSION FMT(8)
      DATA INDBLK/0,1,4,10,20,35,56,84,120,165,220,286,364,455,560,680,
     1 816,969,1140,1330/
      DATA INDROW/0,1,3,6,10,15,21,28,36,45,55,66,78,91,105,120,136,
     1153,171,190/
      READ(5,600) IP,IG,(NM(I),I=1,IG)
  600 FORMAT(2I5/16I5/4I5)
      DO 5 I=1,1540
      W(I)=0.0
    5 T(I)=0.0
      IA=INDBLK(IP)
      DO 10 I=1,20
      XSUM(I)=0.0
      DO 10 J=1,20
   10 XMSUM(I,J)=0.0
      READ(5,601) FMT
  601 FORMAT(8A10)
      N=0
      DO 100 I=1,IG
      N=N+NM(I)
      NMCASE=NM(I)
      DO  100 J=1,NMCASE
      READ(5,FMT) (X(K),K=1,IP)
      DO 100 K=1,IP
      XSUM(K)=XSUM(K)+X(K)
      XMSUM(I,K)=XMSUM(I,K)+X(K)
      IB=IA+INDROW(K)
      DO 100 L=1,K
      IC=IB+L
  100 W(IC)=W(IC)+X(K)*X(L)
      DO 60 I=1,IP
      IB=IA+INDROW(I)
      DO 60 J=1,I
      IC=IB+J
      T(IC)=W(IC)-XSUM(I)*XSUM(J)/FLOAT(N)
      DO 60 K=1,IG
   60 W(IC)=W(IC)-XMSUM(K,I)*XMSUM(K,J)/FLOAT(NM(K))
      CALL DISC(W,T,IP,INDBLK,INDROW)
      STOP
      END
      SUBROUTINE DISC(W,T,N,INDBLK,INDROW)
      DIMENSION W(1),T(1),INDBLK(1),INDROW(1),SAVE(20),NT(21)
      DIMENSION NX(20),USAVE(200),ISAVE(200)
      NX(N)=-10
      KOUNT=0
      ITEN=10E11
      WRITE(6,700)
  700 FORMAT(1H1)
      NM=N-1
      NA=N+1
      NT(NA)=1
      DO 1 I=1,200
      USAVE(I)=1.0
    1 ISAVE(I)=0
      DO 5 I=1,N
    5 NT(I)=0
   10 DO 20 I=1,NM
      IF(NT(I).EQ.0)GO TO 21
      NT(I)=0
   20 CONTINUE
      I=N
      IF(NT(I).NE.0) GO TO 100
      NT(I)=1
      KOUNT=KOUNT+1
      IM=I-1
      GO TO 23
   21 NT(I)=1
      KOUNT=KOUNT+1
      K=I +1
      IM=I-1
      DO 22 J=K,N
      IF(NT(J).EQ.1)GO TO 30
   22 CONTINUE
   23 JM=N
      IA=INDBLK(JM)
      IC=IA+INDROW(I)+I
      U=W (IC)/T(IC)
      GO TO 29
   30 JM=J-1
      IA=INDBLK(JM)
      IC=IA+INDROW(I)+I
      U=SAVE(JM)*W(IC)/T(IC)
   29 INUM=NX(JM)+10
      IF(USAVE(INUM+10).LE.U) GO TO 70
      DO 50 LOOPA=1,9
      LOOP=11-LOOPA
      ID=INUM+LOOP
      IB=ID-1
      IF(U.GE.USAVE(IB)) GO TO 65
      ISAVE(ID)=ISAVE(IB)
   50 USAVE(ID)=USAVE(IB)
      ID=IB
   65 USAVE(ID)=U
      ISAVE(ID)=KOUNT
   70 IF(I.EQ.1) GO TO 10
      SAVE(IM)=U
      NX(IM)=NX(JM)+10
      IB=INDBLK(IM)
      ID=IA+INDROW(I)
      DO 31 L=1,IM
      IF=ID+L
      A=W(IF)/W(IC)
      B=T(IF)/T(IC)
      LP=INDROW(L+1)
      IH=IA+LP
      II=IB+LP
      DO 31 M=L,IM
      IJ=ID+M
      W(II)=W(IH)-A*W(IJ)
      T(II)=T(IH)-B*T(IJ)
      II=II+M
 31   IH=IH+M
      GO TO 10
100   N10=10*N-9
      DO 120 I=1,N10,10
      IUSE=I-1
      II = IUSE/10+1
      WRITE(6,601) II
601   FORMAT(/////13H U RATIO FOR ,I2,10H VARIABLES//)
      DO 115 J=1,10
      KOUNT=ISAVE(IUSE+J)
      IF(KOUNT.EQ.0) GO TO 120
      L=0
      DO 110 K=1,N
      IF(KOUNT-KOUNT/2*2.EQ.0) GO TO 110
      L=L+1
      NT(L)=K
110   KOUNT = KOUNT/2
115   WRITE(6,600) USAVE(IUSE+J),(NT(K),K=1,L)
600   FORMAT(1X,E14.7,2X,20I2)
120   CONTINUE
      RETURN
      END
```