ZASTSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XVI, 2 (1978)

ALGORITHM 55

ANNA BARTKOWIAK (Wrocław)

MULTIPLE REGRESSION WITH STEPWISE SELECTION OF VARIABLES

1. Procedure declaration. Let c be the covariance matrix of the variables $x_1, x_2, \ldots, x_p, y$. We want to examine the regression of the last (criterion) variable y upon the p predictor variables $x_1, x_2, \ldots, x_p$. The calculations may be run in a number of ways:

1° Simple way. The variables $x_1, x_2, \ldots, x_p$ are introduced into the regression set in the order in which they appear in the matrix c.

2° Stepwise selection. Let us assume that there are already k variables introduced into the regression set. We select that variable which adds the greatest amount to the (multiple) correlation coefficient of the variable y with the variables $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$. The variable to be chosen has to be significant at some significance level α.

3° A prescribed number of variables is introduced in a simple way, and then we start the selection procedure.

4° Having selected a declared number of variables we may change the significance level by diminishing the value of α, and eliminate those variables which do not agree with the new significance level. Then, if the required size of the regression set is not completed yet, we may anew start the selection procedure seeking to add to the regression set further variables corresponding to the new significance level.

Procedure *maxstepregr* is also suitable for discriminant analysis.

After each step we may print out the multiple correlation coefficient, the regression coefficients, the residual variance and various test statistics such as the t-statistic for the significance of the regression coefficient and F-test for the significance of the multiple correlation coefficient.

Data:

regr — Boolean variable indicating the goal of the calculations; if *regr* ≡ **true**, the goal is regression analysis, and if *regr* ≡ **false**, the goal is discriminant analysis;

simple — Boolean variable conditioning the way of the run of the procedure; if *simple* ≡ **true**, the variables are introduced into the regression set according to their order in the matrix *c*;

vr1, vb1, vb2 — Boolean variables, not used directly in *maxstepregr*, conditioning the printout of the results; they are used in procedure *printregr* supplied by the user;

n — total number of observations (used in *printregr* in the case of discriminant analysis);

mn — some arbitrary constant not used directly in *maxstepregr* but in *printregr* (e.g., *mn* $= (n_1+n_2)/n_1n_2$ in the case of discriminant analysis);

p — number of predictor variables;

c[*1*: $(p+1)(p+4)\div 2$] — lower triangle of the adjusted product matrix in the following order:

c_{11}
c_{21} c_{22}
. . . . . .
c_{p1} c_{p2} ... c_{pp}
c_{y1} c_{y2} ... c_{yp} c_{yy}
$\bar{x}_1$ $\bar{x}_2$ ... $\bar{x}_p$ $\bar{y}$

the last row contains the means of the variables under investigation, not directly used in *maxstepregr* but needed in *printregr*;

nr[*1*: *p*] — primary indices of the predictor variables under consideration (their order numbers in the original data basis);

init — initial number of variables to be chosen without selection (to be set only if a stepwise run is required);

l — required final size of the regression set containing the "best" predictor variables (mostly correlated with *y*);

alfa — initial significance level;

alfa1 — final significance level;

step — step by which *alfa* should be diminished to attain *alfa1*;

big — large positive number playing the role of infinity;

eps — small positive number: if the pivot element of the considered variable is less than *eps*, the pivoting transformations are not performed;

Ftest — identifier of the real procedure calculating the probabilities for Snedecor's F-distribution, headed as follows: **real procedure** *Ftest*(*fc*, *df1*, *df2*, *maxn*); **value** *fc*, *df1*, *df2*, *maxn*; **real** *fc*; **integer** *df1*, *df2*, *maxn*; where *fc* is the calculated value of the F-test function, *df1*, *df2* are the numbers of degrees of freedom in the numerator and denominator, respectively, of the formula for F, and *maxn* denotes that normal approximation is used if *df1* or *df2* exceeds *maxn*; the body of the procedure may be found in [4].

Results:

ind[*1*: *p*] — indicator array designating the variables actually present in the regression set:

$$ind[i] = \begin{cases} 1 & \text{if variable no. } i \text{ is present,} \\ 0 & \text{if variable no. } i \text{ is not present;} \end{cases}$$

c — transformed input matrix *c*.
Let $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ denote the variables actually present in the regression set. The rows and columns of the matrix *c* corresponding to those variables are proportional (up to the scale changing transformation; see Section 2) to the inverse of the matrix *c* corresponding to these variables.

Other results may be obtained by procedure *printregr* whose body should be supplied by the user. Procedure *printregr* should be headed as follows:

procedure *printregr*(*regr*, *stage*, *vr1*, *vb1*, *vb2*, *p*, *lr*, *n*, *mn*, *c*, *sd*, *nr*, *ind*);
value *regr*, *stage*, *p*, *lr*, *n*, *mn*;
integer *p*, *lr*, *n*;
real *mn*;
boolean *regr*, *stage*, *vr1*, *vb1*, *vb2*;
integer array *nr*, *ind*;
array *c*, *sd*;

Data:

regr — as in Data for *maxstepregr*;
stage — indicates whether the results printed are some intermediary or final results, issued at the end of the run *maxstepregr*;
vr1, *vb1*, *vb2* — auxiliary variables specifying the results printed: *vr1* ≡ **true** causes the printout of the multiple regression coefficient after each step, *vb1* ≡ **true** causes the printout

```
procedure maxstepregr(regr,simple,vr1,vb1,vb2,n,mn,p,c,nr,
 init,l,alfa,alfa1,step,big,eps,Ftest,printregr);
 value regr,simple,vr1,vb1,vb2,n,mn,p,init,l,alfa,alfa1,
 step,big,eps;
 Boolean regr,simple,vr1,vb1,vb2;
 integer n,p,init,l;
 real mn,alfa,alfa1,step,big,eps;
 integer array nr;
 array c;
 procedure Ftest,printregr;
 begin
  integer i,j,k,lr,p1,pk,pq,q,r;
  real max,x,y,z,t;
  integer array ind[1:p];
  array d,dt,sd[1:p+1];
  procedure printout(stage);
   Boolean stage;
   printregr(regr,stage,vr1,vb1,vb2,p,lr,n,mn,c,sd,nr,ind);
  procedure onestep(q,v);
   value q,v;
   integer q;
   real v;
   begin
    integer k,r;
    r:=q×(q-1)÷2;
    x:=-1.0/c[r+q];
    for k:=1 step 1 until q do
     d[k]:=c[r+k];
    for i:=q+1 step 1 until p1 do
     d[i]:=c[i×(i-1)÷2+q];
```

```
   for i:=1 step 1 until p1 do
    dt[i]:=d[i]×x×v;
  dt[q]:=x;
  for k:=1 step 1 until q do
   c[r+k]:=dt[k];
  for i:=q+1 step 1 until p1 do
   c[i×(i-1)÷2+q]:=dt[i];
  k:=0;
  for i:=1 step 1 until p1 do
   begin
    if i≠q
     then
     begin
      x:=d[i];
      for j:=1 step 1 until i do
       if j≠q
        then c[k+j]:=c[k+j]+x×dt[j]×v;
     end i≠q;
    k:=k+i
   end i
 end onestep;
p1:=p+1;
pk:=p×(p+1)÷2;
pq:=pk+p1;
if simple
 then init:=p;
k:=0;
for i:=1 step 1 until p do
 ind[i]:=0;
for i:=1 step 1 until p1 do
```

```
 begin
  comment normalization of the input matrix;
  x:=sd[i]:=1.0/sqrt(c[k+i]);
  for j:=1 step 1 until i do
   c[k+j]:=c[k+j]×x×sd[j];
  k:=k+i
 end i;
t:=1.0/sd[p1];
lr:=k:=0;
for i:=1 step 1 until p do
 sd[i]:=sd[i]×t;
for r:=1 step 1 until init do
 begin
  k:=k+r;
  if c[k]>eps
   then
   begin
    onestep(r,1.0);
    ind[r]:=1;
    lr:=lr+1;
    printout(true)
   end
   else outinteger(1,nr[r]);
  comment warning on the dependence of variable no.nr[r];
 end r;
if init>0
 then printout(false);
if ¬simple
 then
 begin
```

```
    comment forward selection;
forw:
    max:=.0;
    q:=0;
    if lr≥1
     then go to back;
    for r:=1 sten 1 until p do
     if ind[r]=0
      then
      begin
       y:=c[r×(r+1)÷2];
       if y>eps
        then
        begin
         x:=c[pk+r];
         x:=x×x/y;
         if x>max
          then
          begin
           max:=x;
           q:=r
          end x>max
        end y>eps
      end r;
    if q=0
     then go to kmsr;
    if Ftest(max×(n-lr-2)/(c[pq]-max),1,n-lr-2,80)<alfa
     then
     begin
      onestep(q,1.0);
```

```
        ind[q]:=1;
        lr:=lr+1;
        printout(true)
       end test<alfa
       else go to kmsr;
      k:=lr;
      comment now back elimination;
back:
      q:=0;
      y:=big;
      for r:=1 step 1 until p do
       if ind[r]=1
        then
        begin
         x:=-c[pk+r]↑2/c[r×(r+1)÷2];
         if x<y
          then
          begin
           y:=x;
           q:=r
          end x<y
        end r;
      if q=0
       then go to kmsr;
      if Ftest(y×(n-lr-1)/c[pq],1,n-lr-1,80)>alfa
       then
       begin
        ind[q]:=0;
        lr:=lr-1;
        onestep(q,-1.0);
```

```
                go to back
               end F>alfa;
              if k>lr
               then printout(true);
              if lr<l
               then go to forw;
        kmsr:
              if alfa>alfa1
               then
               begin
                k:=lr;
                alfa:=alfa-step;
                outreal(1,alfa);
                if alfa<alfa1
                 then alfa:=alfa1;
                go to back
               end alfa>alfa1;
              printout(false)
             end ¬simple
            end maxstepregr
```

of the regression equation after each step performed, *vb2* ≡ **true** causes the printout of the regression equation only at the run of *maxstepregr*;

p — number of predictor variables dealt with (see Data in *maxstepregr*);

lr — number of variables actually present in the regression set;

n — total number of observations;

mn — some constant (see Data for *maxstepregr*);

c — transformed matrix c (see Results of *maxstepregr*);

$sd[1: p+1]$ — reciprocals of standard deviations (if the matrix c is the covariance matrix) or the reciprocals of square roots of the adjusted sums of squares (if the matrix c contains the adjusted sums of squares and products) of the variables $x_1, x_2, \ldots, x_p, y$ (see formula (6));

```
procedure printregr(regr,stage,vr1,vb1,vb2,p,lr,n,mn,c,da,
 nr,ind);
 value regr,p,lr,n,mn;
 integer p,lr,n;
 real mn;
 Boolean regr,stage,vr1,vb1,vb2;
 integer array nr,ind;
 array c,da;
 begin
  integer i,k,pk,pq;
  real x,y,z,t;
  array b[0:p];
  comment global variables
Ftest,normal calculating probabilities
dif[nr[1]],dif[nr[2]],...,dif[nr[p]]-differences
between means of observed variables;
  procedure druk26(x);
   value x;
   real x;
   begin
    if x<.05
     then outchar(26)
     else outchar(64);
    if x<.01
     then outchar(26)
     else outchar(64);
    if x<.001
     then outchar(26)
     else outchar(64)
   end druk26;
```

```
procedure printrr;
 begin
  x:=c[pq];
  x:=(1.0-x)/x;
  if ¬regr
   then
   begin
    format('??dd=123␣456␣789.123␣␣␣1.1234??');
    y:=x×(n-2)×mn;
    print(y,'p(2/1)=p(1/2)=',normal(.5×y,true))
   end not regr;
  format('
␣lz=12␣␣rr=1.1234␣␣␣Fo=12345.123␣␣␣p(F␣gt␣Fo)=1.1234␣␣');
  y:=x×(n-lr-1)/lr;
  z:=Ftest(y,lr,n-lr-1,80);
  print(lr,1-c[pq],y,z);
  druk26(z);
  format('␣␣se=123␣456.1234');
  print(sqrt(c[pq]/(n-lr-1))/da[p+1]);
  line(2)
 end printrr;
procedure printb;
 begin
  for i:=p step -1 until 1 do
   if ind[i]≠0
    then b[i]:=-c[pk+i]×da[i];
  if key(6)∧¬regr
   then
   begin
    z:=.0;
```

```
      comment
  scaling the coefficients of the discriminant function;
      for i:=p step -1 until 1 do
       if ind[i]≠0
        then z:=z+b[i]×dif[nr[i]];
       z:=(n-2)×mn/(1.0-z)
      end key(6)
      else z:=1.0;
     format('?␣␣␣b[12]=-123␣456.123␣456␣␣␣t=-123456.123?');
     x:=.0;
     t:=1.0/c[pq];
     for i:=p step -1 until 1 do
      begin
       if ind[i]≠0
        then
        begin
         b[i]:=y:=b[i]×z;
         x:=x+y×c[pq+i];
         print(nr[i],y);
         y:=-c[pk+i]×sqrt((-n+lr+1)×t/c[i×(i+1)÷2]);
         if abs(y)>999999.999
          then y:=999999.999;
         print(y)
        end ind[i]≠0
        else b[i]:=.0
      end i;
     format('?␣␣␣b[12]=-123␣456.123␣456??');
     y:=b[0]:=c[pq+p+1]-x;
     print(i,y);
     if key(23)
```

```
      then
      begin
       comment
calculations of the discriminant scores for both
groups of data, not described here;
      end key(23);
    end printb;
  pk:=p×(p+1)÷2;
  pq:=pk+p+1;
  if stage
   then
   begin
    if vr1
     then printrr;
    if vb1
     then printb
   end stage
   else
   begin
    if ¬vr1
     then printrr;
    if vb2∧¬vb1
     then printb
   end ¬stage
 end printregr
```

$nr[1:p]$ — order numbers of the variables $x_1, x_2, \ldots, x_p$ in the primary data basis (see Data for *maxstepregr*);

$ind[1:p]$ — indicator array designating those variables which are actually present in the regression set (see also Results of *maxstepregr*).

The way in which these data may be used to obtain the regression equations, test statistics, residual variances, etc., is explained in Section 2.

2. Method used. At the begin of the run of the procedure the input matrix c is normalized to the correlation matrix to attain a higher stability of the algorithm subsequently used (see [2]). This being done, the proper calculations concerning regression analysis are started.

The procedure uses the modified Gauss-Jordan algorithm [1] which permits stepwise introduction or elimination of variables. The algorithm is such that, for each introduced or eliminated variable, one transformation of the matrix c is performed. The formulae for the transformations are as follows. For introducing the variable no. r into the regression set we have to perform the following transformation (the prime $'$ denotes the element after transformation):

$$\begin{aligned} c'_{rr} &= -1/c_{rr}, \\ c'_{ir} &= c'_{ri} = c_{ir}c'_{rr} \\ c'_{ij} &= c'_{ji} = c_{ij} + c_{ir}c'_{rj} \end{aligned} \qquad (i,j = 1,2,\ldots,q = p+1, i,j \neq r, r \leqslant p). \tag{1}$$

To eliminate variable no. r from the regression set we have to perform the following transformation:

$$\begin{aligned} c'_{rr} &= -1/c_{rr}, \\ c'_{ir} &= c'_{ri} = -c_{ir}c'_{rr} \\ c'_{ij} &= c'_{ji} = c_{ij} - c_{ir}c'_{rj} \end{aligned} \qquad (i,j = 1,2,\ldots,q = p+1,\ i,j \neq r,\ r \leqslant p).$$

The variables actually being in the regression set can be identified by the indicator array *ind*.

After each step, partial results may be printed out by a call of *printregr*.

Typical results can be obtained as follows (r_{ij} stands for the actual value of the input element c_{ij} which was normalized and transformed, s_i and s_y denote square roots of the diagonals c_{ii} and c_{qq}).

To calculate the regression coefficient b_i of y on the i-th variable (if it is present in the regression set) we use the formula

$$b_i = -r_{iq}\frac{s_y}{s_i}. \tag{2}$$

To evaluate the standard deviation (error) for the computed value b_i we use the formula

$$s(b_i) = \frac{\sqrt{-r_{ii}}}{s_i}\, s_e, \tag{3}$$

where

$$s_e = s_y \sqrt{\frac{r_{yy}}{n-k-1}} \tag{4}$$

is an estimator of the standard deviation of the residuals,

$$s_e = \sqrt{\frac{1}{n-k-1}\sum_{i=1}^{n}(y_i - b_0 - b_1 x_{i_1} - \ldots - b_k x_{i_k})^2},$$

and k stands for the number of predictor variables actually present in the regression set.

If all p variables are introduced into the regression set, to get the inverse of the input matrix c we have to use the following formula:

$$c^{ij} = -\frac{r_{ij}}{s_i s_j} \qquad (i, j = 1, 2, \ldots, p). \tag{5}$$

It is to be emphasized that normalizing the input matrix c we made the substitution

$$sd[i] = \sqrt{\frac{c_{vv}}{c_{ii}}} \qquad (i = 1, 2, \ldots, p), \tag{6}$$

$$sd[p+1] = \frac{1}{\sqrt{c_{vv}}},$$

where *sd* is a formal parameter of procedure *printregr*.

If the input matrix c is the adjusted sum of squares and product matrix whose elements are defined by

$$c_{ij} = \sum_{l=1}^{n}(x_{li} - \bar{x}_i)(x_{lj} - \bar{x}_j) \qquad (i, j = 1, 2, \ldots, p, q = p+1), \tag{7}$$

then formulae (2)-(4) and (5) can be rewritten with the use of the array *sd*:

$$b_i = -r_{iq} sd[i], \qquad s(b_i) = sd[i]\sqrt{\frac{-r_{ii} r_{vv}}{n-k-1}},$$

$$s_e = \frac{1}{sd[p+1]}\sqrt{\frac{r_{vv}}{n-k-1}}, \qquad c^{ij} = -r_{ij} sd[i] sd[j] (sd[p+1])^2. \tag{8}$$

If the run of *maxstepregr* is with selection of variables, then each variable introduced into the regression set results from the following two actions:

1° From all variables not being in the regression set that one is typified to be selected which reduces mostly the value of r_{qq}.

2° The typified variable has to be accepted at the significance level *alfa* by the test procedure

$$\Omega\colon\ y = b_0 + b_1 x_{i_1} + \ldots + b_k x_{i_k} + b_{k+1} x_{i_{k+1}} + e,$$

$$H_0\colon\ b_{k+1} = 0,$$

$$\omega\colon\ y = b_0 + b_1 x_{i_1} + \ldots + b_k x_{i_k} + e,$$

$$F_c = \frac{SS_\omega - SS_\Omega}{1} : \frac{SS_\Omega}{n-k-2},$$

where $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ stand for variables already present in the regression set, $x_{i_{k+1}}$ is the typified variable, SS_ω and SS_Ω denote the residual sums of squares corresponding to the models ω and Ω specified above.

The typified variable $x_{i_{k+1}}$ enters the regression set if

$$\mathrm{P}(F > F_c \mid H_0) < \textit{alfa}.$$

As we see, the test procedure used here does not take into account that the variables are selected in a specific manner but acts as if they were some declared variables, and so the test used is not utterly correct.

Elimination of variables is performed by two similar actions:

1° From all variables $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ being present in the regression set that one is typified which, when eliminated, augments the residual variance at least. Say, it is the variable x_{i_k}.

2° The typified variable x_{i_k} has to be rejected at the significance level *alfa* by the following test procedure:

$$\Omega\colon\ y = b_0 + b_1 x_{i_1} + \ldots + b_k x_{i_k} + e,$$

$$H_0\colon\ b_k = 0,$$

$$\omega\colon\ y = b_0 + b_1 x_{i_1} + \ldots + b_{k-1} x_{i_{k-1}} + e,$$

$$F_c = \frac{SS_\omega - SS_\Omega}{1} : \frac{SS_\Omega}{n-k-1}.$$

The variable x_{i_k} is eliminated from the regression set if

$$\mathrm{P}(F > F_c \mid H_0) < \textit{alfa}.$$

As previously, this test handles the variable x_{i_k} like a given one and not a selected one and is not entirely correct.

3. Test example. Let the following data be given:

no.	x_1	x_2	x_3	y
1	5	6	5	0.6
2	4	4	5	0.6
3	6	6	3	0.6
4	5	4	7	0.6

5	0	0	0	−0.4
6	−1	0	−2	−0.4
7	−2	−1	0	−0.4
8	0	−2	0	−0.4
9	2	3	0	−0.4
10	1	0	2	−0.4

For these data, by formula (7) we calculate the totals of x_1, x_2, x_3, y and the matrix of sums of adjusted squares and products from which we need the lower triangle only:

```
c = [72
     70   78
     62   56   76
     12   12   12   2.4
      2.0  2.0  2.0  0.0].
```

We wish to find the regression equation describing the regression of the variable y on the variables x_1 and x_2. To do this we have to call *maxstepregr* with the following values:

regr = **true**,
simple = **true**,
vr1, *vb1*, *vb2* — as specified in procedure *printregr*,
n = 10,
mn — for instant arbitrary,
p = 3,
c[*1*: *14*] = [72 70 78 62 56 76 12 12 12 2.4 2.0 2.0
2.0 0.0],
nr[*1*: *3*] = [1 2 3],
init — arbitrary,
l — arbitrary,
alfa, *alfa1*, *step*, *big* — arbitrary,
eps = $_{10}-10$,
Ftest — library procedure body (see [4]).

Results:

```
ind[1 : 3] = [1  1  0],
c[1 : 14]  = [−7.8436
               7.3265  −7.8436
              −1.2452   0.4358  0.2733
              −0.7344  −0.1911  0.1340  0.1620
               2.0000   2.0000  2.0000  0.0000].
```

Other results, especially intermediate results after introduction of x_1 into the regression set, can be obtained by *printregr* which is conditioned by the Algol compiler, used in the dealt case, especially by its editing procedures.

In *printregr* the output matrix c can be denormalized, and by formulae (2)-(4) or (8) we can get, among others, the following results:

1° After the first stage of calculations, it is after introduction of x_1 into the regression set, we obtain

the coefficients of the regression equation

$$y = b_0 + b_1 x_1 + e$$

with $b_0 = -0.333333$ and $b_1 = 0.166667$ (b_0 follows from the condition $\bar{y} = b_0 + b_1 \bar{x}_1$);

the square of the correlation coefficient $R^2_{y(1)} = 0.8333$;

the standard deviation of the residuals $s_e = 0.2236$.

2° After the second stage of calculations, with both x_1 and x_2 in the regression set, we obtain

the coefficients of the regression equation

$$y = b_0 + b_1 x_1 + b_2 x_2 + e$$

with $b_0 = -0.335196$, $b_1 = 0.134078$, and $b_2 = 0.033520$ (b_0 follows from the condition $\bar{y} = b_0 + b_1 \bar{x}_1 + b_2 \bar{x}_2$);

the square of the multiple correlation coefficient $R^2_{y(1,2)} = 0.8380$;

the standard deviation of the residuals $s_e = 0.2357$.

Now let us suppose that the data dealt with comprise two sets. The first four rows are sampled from one group, and the second six rows are sampled from another group of subjects, and we wish to calculate a linear function of x_1 and x_2 discriminating the two groups under consideration. For this purpose we define an auxiliary variable y as follows (see [3]):

$$y_i = \begin{cases} \dfrac{n_2}{n_1 + n_2} & \text{if row } i \text{ belongs to I group of subjects,} \\ \dfrac{-n_1}{n_1 + n_2} & \text{if row } i \text{ belongs to II group of subjects.} \end{cases}$$

The variable y so defined takes in our example exactly the values y as they are given.

To obtain the discriminant function we start now the regression procedure with the same input values as in the regression case, except the values *regr* and *mn* which now are set: *regr* = **false** and *mn* = 10/24. In the last row of the matrix c we put now the arithmetic mean of the means of both groups except $\bar{y} = 0$ so that the matrix c has the following values:

$$c = [72$$
$$70 \quad 78$$
$$62 \quad 56 \quad 76$$
$$12 \quad 12 \quad 12 \quad 2.4$$
$$2.5 \quad 2.5 \quad 2.5 \quad 0.0].$$

We emphasize that the last row is not directly used in *maxstepregr*, but is needed only for the calculations of the free term in the regression equation.

After the call of *maxstepregr* we obtain the values b_1 and b_2 as in the regression case. The constant b_0 is derived from the condition that the discriminant plane should pass through the arithmetic mean of mean values for each group and take then the value zero,

$$0 = b_0 + b_1 \bar{x}_1^0 + b_2 \bar{x}_2^0,$$

where $\bar{x}_1^0 = (\bar{x}_1^{\mathrm{I}} + \bar{x}_1^{\mathrm{II}})/2$ and $\bar{x}_2^0 = (\bar{x}_2^{\mathrm{I}} + \bar{x}_2^{\mathrm{II}})/2$, $\bar{x}_1^{\mathrm{I}}, \bar{x}_2^{\mathrm{I}}$ being the means for group I and $\bar{x}_1^{\mathrm{II}}, \bar{x}_2^{\mathrm{II}}$ for group II.

Let us emphasize that the constant b_0 is different for the regression and discriminant cases.

In our example the best linear discriminant function based on x_1 and x_2 is

$$z = -0.418994 + 0.134078 x_1 + 0.033520 x_2. \tag{9}$$

Positive values of z indicate similarity to group I, and negative values of z indicate similarity to group II under consideration.

The power of the discriminant function can be measured by the Mahalanobis distance D^2 which follows directly from the (squared) multiple correlation coefficient,

$$D^2(x_1, x_2) = \frac{R^2}{1-R^2} \frac{(n_1+n_2)(n_1+n_2-2)}{n_1 n_2},$$

or from the output value of the element c_{qq} and the value of *mn* declared while entering *maxstepregr* as $mn = (n_1+n_2)/n_1 n_2$:

$$D^2(x_1, x_2) = \frac{1-c_{qq}}{c_{qq}} mn(n-2).$$

In our example we get $D^2(x_1, x_2) = 17.241$.

The discriminant function obtained by the regression method can be transformed by multiplication by a constant to another form satisfying directly Fisher's formula for the linear discriminant function,

$$\tilde{b}' = S^{-1} d', \tag{10}$$

where $\tilde{\boldsymbol{b}}' = \{\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_p\}$, $\boldsymbol{d}' = \{x_1^{\mathrm{I}} - x_1^{\mathrm{II}}, x_2^{\mathrm{I}} - x_2^{\mathrm{II}}, \ldots, x_p^{\mathrm{I}} - x_p^{\mathrm{II}}\}$, and $\boldsymbol{S}$ is the within covariance matrix.

It can be shown that the transforming formula is of the form

$$\tilde{\boldsymbol{b}} = \frac{\boldsymbol{b}}{\mathscr{R}(1-A)}, \tag{11}$$

where

$$\mathscr{R} = \frac{n_1 n_2}{(n_1+n_2)(n_1+n_2-2)}, \qquad A = \boldsymbol{b}'\boldsymbol{d},$$

$\boldsymbol{b}' = \{b_1, b_2, \ldots, b_p\}$ denoting the regression coefficients obtained by the regression method.

In our example, Fisher's linear discriminant function takes the form

$$z = -8.620\,690 + 2.758\,621 x_1 + 0.689\,655 x_2. \tag{12}$$

Fisher's discriminant function has the property that the difference between the means of the discriminant scores calculated for the first and second sets of data equals exactly the Mahalanobis distance D^2 between both groups under consideration. This fact is shown in Table 1.

TABLE 1. Discriminant scores z for the example

No. of subject	Group I				Group II					
	1	2	3	4	5	6	7	8	9	10
z by regression method (9)	0.453	0.251	0.587	0.385	−0.419	−0.553	−0.721	−0.486	−0.050	−0.285
Mean score	$\bar{z} = 0.4190$				$\bar{z} = -0.4190$					
z by Fisher's method (12)	9.310	5.172	12.069	7.931	−8.621	−11.379	−14.827	−10.000	−1.034	−5.862
Mean score	$\bar{z} = 8.6207$				$\bar{z} = -8.6207$					

4. Additional remarks.

(a) Introducing the variable no. r into the regression set ($r = 1, 2, \ldots, p$) the algorithm performs the pivoting operation comprised in formula (1):

$$c'_{rr} = 1/c_{rr}. \tag{13}$$

This operation is executed on the correlation matrix of residuals. It may happen that variable no. r is linearly dependent on the variables being actually in the regression set. In such a case the actual value c_{rr} should be equal to zero up to rounding errors, and taking its reciprocal is not executable. In this situation the advice is to omit the variable x_r.

To face this possibility during the run of *maxstepregr*, before performing step (13) we verify whether the following inequality is satisfied:

$$c_{rr} > eps.$$

The pivoting step (13) is executed only if this inequality is true. If not, the order number $nr[r]$ is printed to warn the user of the linear dependence of the variable x_r. The printing instruction comprised in *maxstepregr* is a formal Algol 60 language instruction (*outinteger*(*1*, *nr*[*r*])), and should be changed to a suitable editing procedure specified in the Algol compiler in use.

(b) During a run with variable selection the value of *alfa* is diminished by the quantity *step* with printed signals written as the formal Algol 60 language instruction *outreal*(*1*, *alfa*). This instruction should be changed to a suitable printout procedure specified in the Algol compiler in use.

5. Procedure *printregr*. We enclose an example of procedure *printregr* written in the Algol 1204 language. It may be used for both regression analysis and discriminant analysis between two groups. In evaluating significances of test statistics the procedure uses real functions *Ftest* and *normal*, calculating the tails in Snedecor's F and the normal probability distributions. Performing the scaling of the discriminant functions as shown in (10) and (11), the procedure uses differences between the means of the considered characteristics. These differences are held in the global variables *dif*[*nr*[*1*]], *dif*[*nr*[*2*]], ..., *dif*[*nr*[*p*]].

References

[1] E. M. L. Beale, M. D. Kendall and D. W. Mann, *The discarding of variables in multivariate analysis*, Biometrika 54 (1967), p. 357-366.

[2] G. H. Golub, *Matrix decompositions and statistical calculations*, in: *Statistical computation*, Academic Press, New York 1969.

[3] P. A. Lachenbruch, *Discriminant analysis*, Macmillan, New York 1975.

[4] J. Morris, *Algorithm 346, Ftest probabilities*, Comm. ACM 12 (1969), p. 184-185.

[5] G. Peters and J. H. Wikinson, *On the stability of the Gauss-Jordan elimination with pivoting*, ibidem 18 (1975), p. 20-24.

INSTITUTE OF INFORMATICS
UNIVERSITY OF WROCŁAW
50-384 WROCŁAW

Received on 20. 6. 1976

ALGORYTM 55

ANNA BARTKOWIAK (Wrocław)

REGRESJA WIELOKROTNA Z KROKOWYM WYBOREM ZMIENNYCH

STRESZCZENIE

Rozważamy regresję zmiennej y od zmiennych $x_1, x_2, \ldots, x_p$. Regresję tę możemy obliczać kilkoma sposobami:

1° Regresja prosta. Zmienne $x_1, x_2, \ldots, x_p$ są wprowadzane do równania regresji w takiej kolejności, w jakiej są zadeklarowane, tzn. najpierw x_1, potem x_2 itd., na końcu x_p.

2° Z wybieraniem zmiennych. Wprowadza się kolejno (po jednej) do równania regresji te zmienne, które łącznie ze zmiennymi już wybranymi dają największy współczynnik korelacji wielokrotnej.

3° Sposób mieszany. Najpierw wprowadza się do równania regresji zadeklarowaną liczbę zmiennych, po czym dobiera się następne według kryterium maximum współczynnika korelacji wielokrotnej.

4° Na różnych poziomach istotności. Najpierw wprowadza się do równania regresji l zmiennych na słabym poziomie istotności (np. $\alpha = 0{,}30$), gdzie l jest liczbą deklarowaną przy wejściu do procedury, następnie zaostrza się kryterium istotności (np. obniżając wielkość α do 0,05), usuwa się zmienne nieistotne przy nowym poziomie istotności, po czym dobiera się nowe zmienne (jeśli jest to możliwe), aż łączna liczba zmiennych w równaniu regresji osiągnie zadeklarowaną liczbę l.

Procedura może być również użyta przy analizie dyskryminacyjnej między dwiema grupami danych (por. [3]).

Procedura posługuje się zmodyfikowanym algorytmem Gaussa-Jordana, opisanym wzorem (1), zaczerpniętym z pracy [1]. Stabilność tego algorytmu opisana jest w [5].

Właściwe obliczenia wykonuje się na macierzy korelacji. Obliczenia są wykonywane tylko dla tych zmiennych, dla których współczynnik korelacji wielokrotnej ze zmiennymi już wprowadzonymi do równania regresji jest mniejszy niż *eps*.

Dane:

regr — zmienna boolowska wskazująca na cel obliczeń; przyjmuje ona wartość **true**, gdy celem obliczeń jest analiza regresji, wartość **false**, gdy celem obliczeń jest analiza dyskryminacyjna; nie wpływa na tok obliczeń w *maxstepregr*, jest jedynie zmienną formalną w procedurze *printregr* drukującej wyniki obliczeń;

simple — zmienna boolowska warunkująca wariant obliczeń; jeśli *simple* ≡ **true**, obliczenia przebiegają według wariantu 1°; jeśli *simple* ≡ **false**, obliczenia przebiegają według pozostałych wariantów;

vr1, vb1, vb2 — zmienne boolowskie, nie używane bezpośrednio przy obliczeniach *maxstepregr*, lecz będące parametrami formalnymi procedury *printregr* drukującej wyniki; za pomocą tych zmiennych można określić rodzaj i liczbę drukowanych wyników;

n — liczba obserwacji, nie używana bezpośrednio w obliczeniach procedury, lecz będąca parametrem formalnym procedury *printregr* drukującej wyniki;

mn — pewna stała (mnożnik), nie występująca bezpośrednio w obliczeniach wykonywanych przez procedurę *maxstepregr*, lecz będąca parametrem

formalnym procedury *printregr*; potrzebna przy drukowaniu wyników dla celów analizy dyskryminacji, powinna być wtedy określona jako $mn = (n_1 + n_2)/n_1 n_2$;

p — liczba zmiennych objaśniających $x_1, x_2, \ldots, x_p$;

c — dolny trójkąt macierzy sum poprawionych iloczynów (lub kowariancji) zmiennych $x_1, x_2, \ldots, x_p, y$ oraz (nie używane w *maxstepregr*, lecz występujące w *printregr*) średnie zmiennych $x_1, x_2, \ldots, x_p, y$ (w przypadku analizy dyskryminacyjnej średnie arytmetyczne średnich obu rozważanych grup danych oraz $\bar{y} = 0$);

nr — tablica oryginalnych numerów rozważanych zmiennych $x_1, x_2, \ldots, x_p$;

init — liczba zmiennych, które mają być wprowadzone do równania regresji według wariantu 1° (bez wybierania); używane tylko wtedy, gdy *simple* ≡ **false**; gdy *simple* ≡ **true**, następuje automatyczne podstawienie *init* := *p*;

l — końcowa liczba zmiennych w równaniu regresji; używane tylko wtedy, gdy *simple* ≡ **false**;

alfa — początkowy poziom istotności (używane tylko wtedy, gdy *simple* ≡ **false**);

alfa1 — końcowy poziom istotności (używane tylko wtedy, gdy *simple* ≡ **false**);

step — określa, jakim krokiem ma być zmniejszana wielkość *alfa* do wielkości *alfa1*;

big — duża liczba dodatnia, grająca rolę nieskończoności;

eps — mała liczba dodatnia: jeśli element rozwiązujący r_{kk} rozważanej zmiennej x_k jest mniejszy od *eps*, transformacje nie zostają przeprowadzone;

Ftest — nazwa procedury obliczającej prawdopodobieństwa w rozkładzie *F* Snedecora.

Wyniki:

ind — tablica całkowita, wskazująca na numery zmiennych wprowadzonych do równania regresji;

c — transformowana macierz wejściowa *c*, z której za pomocą wzorów (1)-(4) i (5) lub (6)-(8) można odtworzyć współczynniki równania regresji, macierz odwrotną, zmienność resztową i inne podobne wskaźniki rozważane w analizie regresji. Wygodniej jednak otrzymywać wyniki poprzez procedurę *printregr*, wywoływaną każdorazowo po zmianie liczby zmiennych znajdujących się w równaniu regresji. Tekst tej procedury zależy od instrukcji wyjścia, specyficznych dla poszczególnych maszyn cyfrowych, i powinien być napisany przez użytkownika procedury. Przykład takiej procedury został podany na str. 302.