ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XVII, 3 (1982)

ALGORITHM 84

ANNA BARTKOWIAK (Wrocław)

THE CHOICE OF REPRESENTATIVE VARIABLES BY STEPWISE REGRESSION

1. Procedure declaration. The procedure *idepstep* performs a stepwise search for representative variables satisfying the minimax criterium of residual sums of squares given by formulae (1)-(4). It allows some variables to be introduced obligatorily into the set of representatives R. The search can be performed upwards — introducing new variables into the already established set, and backwards — eliminating the less representative variables from the set.

Data:

p — number of variables;

$a[1:p\times(p+1)\div 2]$ — the lower triangle of the matrix of corrected sums of squares and products, stored row-wise;

$nr[1:p]$ — primary indices of the considered variables;

$ind[0:p+2]$ — integer array whose elements have the following meaning: $ind[0]$ declares the number of variables to be introduced obligatorily into the set of representatives; if $ind[0]\neq 0$, then the values of $ind[1], \ldots, ind[p]$ should be set equal to 1 for these variables and equal to 0 otherwise; the value of $ind[p+1]$ should be set equal to the maximum number of representatives to be selected stepwise upwards, and the value of $ind[p+2]$ should be set equal to the minimum number of representatives to be selected stepwise backwards;

eps — a small number indicating the machine accuracy (e.g. $eps = {}_{10}{-11}$).

Results:

$ind[0:p]$ — indicator array designating the variables actually present in the representative set R: if the i-th variable is in R, then $ind[i]=1$; $ind[0]$ declares for which variable not being in R the residual sum of squares is the largest;

```
procedure idepstep(p,a,nr,ind,eps,onestep,outrepr);
 value p,eps;
 real eps;
 integer p;
 array a;
 integer array nr,ind;
 procedure onestep,outrepr;
 begin
  real flag,min,x,y,z;
  integer h,i,imin,imax,i1,i2,k,l,l1,m,m1,m2;
  array residuals[1:p];
  Boolean done;
  procedure findnext;
   begin
    imin:=ind[0]:=0; min:=₁₀23;
    comment a big number;
    for m:=1 step 1 until p do
     if ind[m]=i1
      then
      begin
       m1:=m×(m-1)+2; x:=a[m1+m];
       if flag×x>eps
        then
        begin
         x:=1.0/x;
         y:=if flag=1.0 then .0 else -x;
         h:=if flag=1.0 then 0 else m;
         k:=0;
         for i:=1 step 1 until p do
          begin
```

```
            if ind[i]=0∧i≠m
            then
            begin
             m2:=if i<m then m1+i else k+m;
             z:=a[k+i]-a[m2]↑2×x;
             if z>y
              then
              begin
               y:=z;
               h:=i
              end z>y
            end ind[i]=0;
           k:=k+i
          end i
        end flag×x>eps
        else h:=0;
       if h>0∧y<min
        then
        begin
         min:=y; imin:=m;
         imax:=h
        end h>0 ∧ y<min
      end m;
    if imin=0
     then done:=true
     else
     begin
      ind[0]:=imax;
      onestep(imin,flag,p,a);
      ind[imin]:=i2; l:=l+flag;
```

```
      k:=0;
      for i:=1 step 1 until p do
       begin
        k:=k+i; residuals[i]:=a[k]
       end i;
      outrepr(p,nr,imin,flag,ind,residuals)
     end imin>0
   end findnext;
 l:=ind[0];
 if l≠0
   then
   begin
    l:=ind[0]:=0;
    for m:=1 step 1 until p do
     if ind[m]=1
      then
      begin
       x:=a[m×(m+1)÷2];
       if x>eps
        then
        begin
         onestep(m,1.0,p,a);
         l:=l+1;
         k:=0;
         for i:=1 step 1 until p do
          begin
           k:=k+i;
           residuals[i]:=a[k]
          end i;
         outrepr(p,nr,m,2.0,ind,residuals)
```

```
          end x>eps
         else ind[m]:=0
       end m;
     outrepr(p,nr,m,3.0,ind,residuals)
    end l≠0;
   l1:=ind[p+1];
   if l<l1
    then
    begin
     done:=false;
     flag:=1.0;
     i1:=0; i2:=1;
addition:
     findnext;
     if l<l1∧¬done then go to addition
    end l<l1;
   l1:=ind[p+2];
   if l>l1
    then
    begin
     done:=false;
     flag:=-1.0;
     i1:=1; i2:=0;
elimination:
     findnext;
     if l>l1∧¬done  then go to elimination
   end l>l1
  end idepstep
```

$a[1:p\times(p+1)\div 2]$ — transformed input matrix a; the diagonal elements for the variables not being in the set of representatives R are equal to the residual sums of squares of these variables regressed on the variables being in R.

Other results, especially those concerning the choice at each step, may be obtained by the procedure *outrepr* described in the sequel.

Other parameters:

onestep — identifier of the procedure calculating one step of the modified Gauss-Jordan procedure, headed as follows: **procedure** *onestep*(q, v, p, c); **value** q, v, p; **real** v; **integer** p, q; **array** c; where q is the no. of the variable to be pivoted in when $v = 1.0$, and pivoted out when $v = -1.0$, p is the total number of variables, c is the lower triangle of the pivoted matrix given row-wise; the procedure performs the transformations given by (5) and (6);

outrepr — identifier of the procedure printing results of the search of representatives; the procedure should be headed as follows: **procedure** *outrepr*$(p, nr, m, flag, ind, residuals)$; **value** p, *flag*, m; **real** *flag*; **integer** p, m; **array** *residuals*; **integer array** *nr*, *ind*; where p is the total number of variables, *nr* — the order numbers of the p variables under consideration in the primary data set, m — the number of the variable last introduced into (eliminated from) the set of representatives R, *flag* — an indicator variable with the following values: -1.0 if the last step was elimination, $+1.0$ if the last step was introduction, $+2.0$ if the last step was obligatory introduction, $ind[0]$ indicates for which variable not belonging to R the residual sum of squares is the largest, $ind[1:p]$ indicates the variables actually being in R, *residuals* contains the diagonal elements of the transformed matrix c.

2. Method used. Let A be the matrix of adjusted sums of squares and products. Let $l_0 = ind[0]$, $l_1 = ind[p+1]$, $l_2 = ind[p+2]$ be given integers. The procedure first introduces l_0 variables into the set R.

Next, if $l_1 > l_0$, the search for representative variables starts. Let $\bar{R}$ be the set of remaining variables, not being in the set R. We put each variable $i_r \in \bar{R}$ to trial assuming it provisionally being adjoined to the representative set R, and seeking the greatest residual sum of squares (unexplained part of variance) of the remaining variables of $\bar{R}$, regressed on the variables being in R. Let $SS_{i_{k0}}^{(i_r)}$ denote the maximum of them:

$$SS_{i_{k0}}^{(i_r)} = \max_{i_k\in\bar{R},\, i_k\neq i_r} SS_{i_k}^{(i_r)}. \tag{1}$$

The variable i_0, for which this maximal unexplained variance is the smallest, i.e.

$$SS_{i_{k0}}^{(i_{r0})} = \min_{i_r \in \bar{R}} SS_{i_{k0}}^{(i_r)}, \tag{2}$$

is chosen to be introduced into the set R of representative variables.

The elimination step is performed similarly. We move temporarily each variable $i_r \in R$ from R to $\bar{R}$ and seek the greatest residual sum of squares of the variables from $\bar{R}$. If

$$SS_{i_{k0}}^{(i_r)} = \max_{i_r \in \bar{R}} SS_{i_k}^{(i_r)} \tag{3}$$

is established for each $i_r \in R$, we remove finally from R that variable i_0 for which

$$SS_{i_{k0}}^{(i_0)} = \min_{i_r \in R} SS_{i_{k0}}^{(i_r)}. \tag{4}$$

If a variable $q = i_0$ to be introduced into R is chosen, we perform a full pivot transformation of the actual matrix $A = \{a_{ij}\}$ by the use of the following formulae given by Jennrich [1]:

$$\begin{gathered} a'_{qq} = -1/a_{qq}, \quad a'_{iq} = a_{iq}/a_{qq}, \quad a'_{qj} = a_{qj}/a_{qq}, \\ a'_{ij} = a_{ij} - a_{iq}a_{qj}/a_{qq}, \quad i \neq q,\ j \neq q. \end{gathered} \tag{5}$$

Similarly, if a variable $q = i_0$ to be eliminated from R is chosen, perform a full inverse pivot transformation of the actual matrix $A = \{a_{ij}\}$ by the use of the following formulae:

$$\begin{gathered} \tilde{a}_{qq} = -1/a_{qq}, \quad \tilde{a}_{iq} = -a_{iq}/a_{qq}, \quad \tilde{a}_{qj} = -a_{qj}/a_{qq}, \\ \tilde{a}_{ij} = a_{ij} - a_{iq}a_{qj}/a_{qq}, \quad i \neq q,\ j \neq q. \end{gathered} \tag{6}$$

If the pivot element a_{qq} is less than *eps*, the declared real, then none of these transformations is performed.

3. Test example. Let $p = 6$, and

$$a[1:15] = [\begin{array}{llllll} 1.0 & & & & & \\ .5353 & 1.0 & & & & \\ .6346 & .7075 & 1.0 & & & \\ .7037 & .6785 & .8902 & 1.0 & & \\ .3851 & .0971 & .0853 & .2263 & 1.0 & \\ -.5425 & -.4333 & -.5389 & -.5830 & -.5606 & 1.0], \end{array}$$

$$nr[1:p] = [1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6],$$

$$ind[0:8] = [2\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 5\ \ 2], \qquad eps = {}_{10}{-10}.$$

```
procedure onestep(q,v,p,c);
 value q,v,p;
 real v;
 integer p,q;
 array c;
 begin
  real x;
  integer i,j,k;
  array d[1:p];
  k:=q×(q-1)÷2;
  for i:=1 step 1 until q do
   begin
    k:=k+1;
    d[i]:=c[k];
    c[k]:=.0
   end i;
  x:=1.0/d[q];
  d[q]:=-v;
  k:=k+q;
  for i:=q+1 step 1 until p do
   begin
    d[i]:=c[k];
    c[k]:=.0;
    k:=k+i
   end i;
  k:=0;
  for i:=1 step 1 until p do
   begin
    v:=d[i]×x;
    for j:=1 step 1 until i do
```

```
        begin
          k:=k+1;
          c[k]:=c[k]-v×d[j]
        end j
      end i
    end onestep
```

Using the procedures *onestep* and *outrepr*, given as an appendix, we obtain the following results.

(A) Results obtained by the procedure *outrepr*, called 8 times during the run of *idepstep*, are presented in Table 1.

TABLE 1

Step after which the procedure was called	No. of variable	Set of representatives	Residual sums of squares
1st	3, chosen obligatorily		
2nd	5, chosen obligatorily	{3, 5}	[1] .48694; [2] .49808; [4] .18477; [6] .44280
3rd	2, chosen	{2, 3, 5}	[1] .47593; [4] .18104; [6] .44062
4th	4, chosen	{2, 3, 4, 5}	[1] .43857; [6] .43783
5th	1, chosen	{1, 2, 3, 4, 5}	[6] .43739
6th	3, eliminated	{1, 2, 4, 5}	[3] .17473; [6] .45790
7th	4, eliminated	{1, 2, 5}	[3] .39500; [4] .37713; [6] .50999
8th	1, eliminated	{2, 5}	[1] .60143; [3] .49917; [4] .51366; [6] .54082

(B) Results obtained as output parameters of *idepstep* are the following:

$$ind[0:p] = [1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0],$$

$a[1:15]$ = [.60142716
.50264594 −1.0095182
.25029220 .70587263 .49916551
.28655160 .66277519 .40747269 .51365899
.33629308 .09802421 .01675977 .16194453 −1.0095182
−.13617761 −.38247184 −.22364987 −.20503341 −.52346198 .54082216]

```
procedure outrepr(p,nr,m,flag,ind,residuals);
 value p,flag,m;
 real flag;
 integer p,m;
 array residuals;
 integer array nr,ind;
 begin
  integer i;
  format('??variable␣no␣123␣␣');
  if flag≠3.0 then print(nr[m]);
  if flag=1.0 then print('chosen')
   else if flag=-1.0
     then print('eliminated')
     else if flag=2.0
       then print('chosen␣obligatorily');
  if flag≠2.0
   then
   begin
    format('1234');
    print('?The␣set␣of␣representants:?');
    m:=0;
    for i:=1 step 1 until p do
     if ind[i]=1
      then
      begin
       m:=m+1; if m=21
        then m:=line(1);
       print(nr[i])
      end i;
```

```
print('?Residual␣sums␣of␣squares:?');
format('?[1234]␣1234567.12345␣');
m:=ind[0];
for i:=1 step 1 until p do
 if ind[i]=0
  then
  begin
   print(nr[i],residuals[i]);
   if i=m then print('(maxres)')
  end i
 end flag←2.0;
line(2)
end outrepr
```

Reference

[1] R. I. Jennrich, *Stepwise regression*, p. 58-75 in: K. Enslein, A. Ralston and H. Wilf (eds.), *Statistical methods for digital computers*, Vol. 3, Wiley, New York 1977.

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF WROCŁAW
51-151 WROCŁAW

Received on 30. 12. 1978

ALGORYTM 84

ANNA BARTKOWIAK (Wrocław)

WYBIERANIE REPREZENTACJI CECH METODĄ KROKOWEJ ANALIZY REGRESJI

STRESZCZENIE

Procedura *idepstep* wybiera reprezentację cech w następujący sposób:

Załóżmy, że zbiór reprezentantów R zawiera r zmiennych (dopuszcza się możliwość, że $r = 0$).

Każdą z pozostałych zmiennych nie należących do R włączamy chwilowo do tego zbioru, po czym obliczamy regresję wielokrotną pozostałych zmiennych od zmiennych tworzących aktualny zbiór R. Rozważając kolejne zmienne nie należące do R znajdujemy tę, dla której regresja jest najgorsza, a więc zmienność resztowa jest największa.

Kontynuując to postępowanie znajdujemy zmienną, której włączenie do zbioru reprezentantów powoduje, że maksymalna zmienność resztowa (obliczona przy chwilowym włączeniu tej zmiennej do zbioru R) jest najmniejsza w porównaniu ze zmiennościami resztowymi otrzymywanymi przy włączaniu innych zmiennych.

Procedura ***idepstep*** dopuszcza możliwość startowania od pustego zbioru R lub też na życzenie wprowadza podane zmienne do zbioru R, po czym rozpoczyna się właściwe działanie procedury. Powiększanie zbioru reprezentantów R odbywa się na zasadzie krokowej — aż do osiągnięcia liczby reprezentantów równej wartości $ind[p+1]$. W dalszym ciągu następuje eliminacja zmiennych ze zbioru R, aż liczebność tego zbioru zostanie zredukowana do liczby reprezentantów równej wartości elementu $ind[p+2]$.

Dane:

p — liczba rozważanych zmiennych;

$a[1 : p \times (p+1) \div 2]$ — tablica zawierająca dolny trójkąt macierzy korelacyjnej (lub macierzy poprawionych iloczynów), zapamiętana wierszami;

$nr[1 : p]$ — numery rozważanych zmiennych w pierwotnej numeracji danych (wykorzystywane przy drukowaniu wyników za pomocą procedury *outrepr*);

$ind[0 : p+2]$ — tablica, której elementy mają następujące znaczenie: $ind[0]$ — liczba zmiennych, które mają być wprowadzone do zbioru reprezentantów R obowiązkowo; jeśli $ind[0] \neq 0$, to kolejne wartości tej tablicy powinny być określone dla $i = 1, 2, \ldots, p$ jak następuje: $ind[i] = 1$, jeśli zmienna o numerze i ma być wprowadzona obowiązkowo do zbioru R, oraz $ind[i] = 0$ w przeciwnym razie; $ind[p+1]$ — maksymalna liczba reprezentantów; $ind[p+2]$ — minimalna liczba reprezentantów;

eps — mała liczba, oznaczająca zero maszynowe.

Wyniki:

$ind[0]$ — numer zmiennej najgorzej reprezentowanej przez zbiór R;

$ind[1 : p]$ — tablica wskazująca, które zmienne znajdują się aktualnie w zbiorze R; jeśli $ind[i] = 1$ $(i = 1, 2, \ldots, p)$, to zmienna o numerze i znajduje się w zbiorze R, $ind[i] = 0$ w przeciwnym wypadku;

$a[1 : p \times (p+1) \div 2]$ — tablica zawierająca dolny trójkąt przetransformowanej macierzy a.

Inne parametry:

onestep — nazwa procedury pomocniczej wykonującej na tablicy a zmodyfikowaną transformację Gaussa-Jordana według sposobu opisanego w [1], przykład realizacji takiej procedury jest podany;

outrepr — nazwa procedury drukującej wyniki pośrednie; przykład realizacji takiej procedury w języku Algol 1204 jest podany.