INTERDEPENDENCE EXAMINATIONS
BY ANALYSIS OF REGRESSION

1. Procedure declaration. Given the number of variables $n$ and the optional number of representatives $r$, the procedure idep performs a systematic examination of all subsets to select that one which, while considering the regression of the omitted variables upon the selected subset, yields the minimax of residual variances.

Data:

- $n$ — number of variables;
- $r$ — optional number of representatives (the size of the subset to be chosen);
- $c[1 : q \times (q+1) \div 2]$ — lower triangle of the correlation matrix (with diagonals), $q \geq n$ depends upon the Boolean variables $sq$ and $prod$ as explained below;
- $sq$ — Boolean variable assuming the value true if the regression with squares is considered; thus $q = 2n$;
- $prod$ — Boolean variable assuming the value true if the regression with product terms is considered; thus $q = 2n + n(n-1)/2$;
- $combi$ — procedure identifier of the procedure yielding subsequent combinations of $r$ out of $n$ objects, headed as follows:

  \begin{verbatim}
  procedure combi(n, r, ind, first); value n, r;
  integer n, r; integer array ind; Boolean first;
  \end{verbatim}

The procedure COMBI of Mifsud [2] can be applied here.

Results:

- $optind[1 : r]$ — integer array enumerating the selected combination of variables (the representative subset of variables);
- $res[0 : n]$ — residual sums of squares for the $n$ variables under consideration; $res[0]$ stands for the maximum of all residuals; the residuals for the selected set of variables are assumed to be equal to zero.
procedure idep(n,r,c,sq,prod,combi,optind,res);
value n,r,sq,prod;
integer n,r;
array c,res;
integer array optind;
Boolean sq,prod;
procedure combi;
begin
integer dep,h,i,i1,i2,i3,j,k,p,pr,q,qr;
real x,y,min,max;
Boolean first;
h:=r;
p:=n;
if sq
then
begin
p:=p+m;
h:=h+r
end sq;
if prod
then
begin
p:=p+n*(n-1)+2;
h:=h+r*(r-1)+2
end prod;
if r<n
then
begin
integer array ind,ii[1:p];
array a[1:h*(h+1)+2],b[1:(n-r)*p],dt[1:p];
end
max:=1.0;
pr:=r;
if sq
then pr:=pr+r;
if prod
then pr:=pr×(r-1)+2;
qr:=pr×(pr+1)+2;
first:=true;
newcomb:
combi(n,r,ind,first);
if first
then go to fin;
for i:=1 step 1 until p do
ii[i]:=0;
for i:=1 step 1 until r do
ii[ind[i]]:=1;
if sq
then
for i:=1 step 1 until r do
ii[n+ind[i]]:=1;
if prod
then
for i:=1 step 1 until r-1 do
begin
j:=ind[i];
i1:=n-j;
i1:=p-i1×(i1+1)+2-j;
for j:=i+1 step 1 until r do
ii[i1+ind[j]]:=1
end prod;
\( k:=i1:=0; \)
\[
\text{for } i:=1 \text{ step } 1 \text{ until } p \text{ do }
\]
\[
\begin{align*}
\text{begin } & \\
\text{if } ii[i]=1 & \\
\text{then } & \\
\text{for } j:=1 \text{ step } 1 \text{ until } i \text{ do } & \\
\text{if } ii[j]=1 & \\
\text{then } & \\
\text{begin } & \\
\quad k:=k+1; & \\
\quad a[k]:=c[i1+j] & \\
\text{end } & \}
\]
\[
ii[j]:=1; & \\
ii:=i1+1 & \\
\text{end } & \}
\]
\[
\text{end } i; & \\
\min:=0; \}
\]
\[
q:=0; & \\
\text{for } dep:=1 \text{ step } 1 \text{ until } n \text{ do } & \\
\text{if } ii[dep]=0 & \\
\text{then } & \\
\text{begin } & \\
\quad j:=dep=(dep-1)+2; & \\
\quad \text{for } i:=1 \text{ step } 1 \text{ until } dep \text{ do } & \\
\quad dt[i]:=c[j+i]; & \\
\quad \text{for } j:=dep+1 \text{ step } 1 \text{ until } p \text{ do } & \\
\quad dt[j]:=c[j\times(j-1)+2+dep]; & \\
\quad \text{for } i:=1 \text{ step } 1 \text{ until } p \text{ do } & \\
\quad \text{if } ii[i]=1 & \\
\quad \text{then } & \\
\quad \text{begin } & \\
\quad q:=q+1; & \\
\quad \text{end } & \}
\]
\[ b[q] := dt[i] \]
    \[ \text{end i;} \]
    \[ q := q+1; \]
    \[ b[q] := 1.0 \]
    \[ \text{end dep;} \]
    \[ k := pr+1; \]
    \[ i1 := 0; \]
    \[ \text{for q := 1 step 1 until pr do} \]
    \[ \text{begin} \]
    \[ i1 := i1 + q; \]
    \[ x := a[i1]; \]
    \[ \text{if } x > 0 \]
    \[ \text{then} \]
    \[ \text{begin} \]
    \[ x := -1.0 / x; \]
    \[ i2 := i1 + q; \]
    \[ \text{for i := q+1 step 1 until pr do} \]
    \[ \text{begin} \]
    \[ y := dt[i] = a[i2]; \]
    \[ y := y \times x; \]
    \[ \text{for j := q+1 step 1 until i do} \]
    \[ \text{begin} \]
    \[ i2 := i2 + 1; \]
    \[ a[i2] := a[i2] + y \times dt[j] \]
    \[ \text{end j;} \]
    \[ i2 := i2 + q \]
    \[ \text{end i;} \]
    \[ i := 0; \]
    \[ \text{for i := 1 step 1 until n do} \]
    \[ \text{if } i1[1] = 0 \]
then
begin
i3:=i3+q;
y:=dt[k]:=b[i3];
y:=y*x;
for j:=q+1 step 1 until k do
begin
i3:=i3+1;
b[i3]:=b[i3]+y*dt[j]
end j
end i
end x>.0
end q;
i3:=k;
for i:=n-r step -1 until 1 do
begin
x:=b[i3];
if x<min
then min:=x;
i3:=i3+k
end i;
if min>max
then
begin
res[0]:=max:=min;
for i:=1 step 1 until r do
optind[i]:=ind[i];
i3:=0;
for i:=1 step 1 until n do
if i[i]=1
Algorithm 41

then res[1]:=0
else
begin
i3:=i3+k;
res[1]:=b[i3]
end
if min>max;
then go to newcomb:
end idep

2. Method used. The essential features of the method used are described in the paper of Beale [1]. Suppose that the variables $x_1, x_2, \ldots, x_n$ are considered. The first subset of indices obtained by combi is $1, 2, \ldots, r$; thus the variables indexed as $r+1, \ldots, n$ are dropped. For each of the dropped variables we calculate the residual sum of squares assuming the regression relationship of the following forms ($l = r+1, \ldots, n$):

(a) if $sq = \text{false}$ and $prod = \text{false}$, then
$$x_l = b_0 + b_1 x_1 + \ldots + b_r x_r;$$

(b) if $sq = \text{true}$ and $prod = \text{false}$, then
$$x_l = b_0 + b_1 x_1 + \ldots + b_r x_r + b_{r+1} x_{r+1}^2 + \ldots + b_{2r} x_{2r}^2;$$

(c) if $sq = \text{true}$ and $prod = \text{true}$, then
$$x_l = b_0 + b_1 x_1 + \ldots + b_{2r} x_{2r}^2 + b_{2r+1} x_{2r+1} x_{2r+2} + \ldots + b_{2r+r(r-1)/2} x_{r-1} x_r.$$

We calculate the residual sum of squares for $l = r+1, \ldots, n$ and mark the maximum value maxres(1, ..., r). Next we continue examining further subsets $(i_1, \ldots, i_r)$ yielded by subsequent calls of combi. We choose as a representative set that one which gives the minimum of the maxres values maxres($i_1, \ldots, i_r$) calculated for each subset.

3. Certification.

Example 1. Calling idep with the values
$$n = 3, \quad r = 2,$$
$$e[1:6] = [1.0000, .4899, 1.0000, .4899, -.5000, 1.0000],$$
$$sq = \text{false}, \quad prod = \text{false},$$
we get the following results:

\[ \text{optind}[1 : 2] = [1, 2], \quad \text{res}[\theta : 3] = [0.0395, 0.0, 0.0, 0.0395]. \]

**Example 2.** Calling `idep` with the values

\[ n = 3, \quad r = 1, \]
\[ c[1 : 15] = [1.0000, 0.4899, 1.0000, 0.4899, -0.5000, 1.0000, 0.9883, 0.4721, 0.4721, 1.0000, 0.5094, 0.9878, -0.4939, 0.5065, 1.0000, 0.5094, -0.4939, 0.9878, 0.5065, -0.4797, 1.0000], \]
\[ sq = \text{true}, \quad prod = \text{false}, \]

we get the following results:

\[ \text{optind}[1 : 1] = [2], \quad \text{res}[\theta : 3] = [0.7500, 0.7223, 0.0000, 0.7500]. \]

**Example 3.** Calling `idep` with the values

\[ n = 3, \quad r = 2, \]
\[ c[1 : 36] = [1.0000, 0.4899, 1.0000, 0.4899, -0.5000, 1.0000, 0.9883, 0.4721, 0.4721, 1.0000, 0.5094, 0.9878, -0.4939, 0.5056, 1.0000, 0.5094, -0.4939, 0.9878, 0.5065, -0.4797, 1.0000, 0.7340, 0.9350, -0.2158, 0.7316, 0.9534, -0.2056, 1.0000, 0.7340, -0.2158, 0.9350, 0.7316, -0.2056, 0.9534, 0.0862, 1.0000, 0.8581, 0.4671, 0.4671, 0.8120, 0.4250, 0.4250, 0.6439, 0.6439, 1.0000], \]
\[ sq = \text{true}, \quad prod = \text{true}, \]

we get following results:

\[ \text{optind}[1 : 2] = [2, 3], \quad \text{res}[\theta : 3] = [0.0, 0.0, 0.0, 0.0]. \]

**References**


**Institute of Informatics**
**University of Wroclaw**
**50-384 Wroclaw**

*Received on 26. 5. 1974*
ANNA BARTKOWIAK (Wrocław)

BADANIE WSPÓŁZALEŻNOŚCI CECH METODĄ ANALIZY REGRESJI

STRESZCZENIE

Procedura idep przeszukuje w sposób systematyczny wszystkie podzbiory \( r \)-elementowe \( (r < n) \) danego zespołu zmiennych \( (x_1, \ldots, x_n) \), wybierając ten podzbiór \( (x_{i_1}, \ldots, x_{i_r}) \), na podstawie którego można wyznaczyć pozostałe zmienne z najmniejszą wariancją resztową. W zależności od zmiennych boolowskich \( sq \) i \( prod \) uwzględnia się również regresję z kwadratami i iloczynami zmiennych (por. wzory (a), (b) i (c)).