ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
20,2 (1990), pp. 227–238

A. BARTKOWIAK (Wrocław)

EXPERIENCE IN COMPUTING OPTIMAL REGRESSION BY BRANCH AND BOUND

0. For given predictor variables $x_1, x_2, \ldots, x_p$ and a predicted variable y we consider an algorithm for search of the optimal subset of size k. The criterion of optimality is: the optimal subset should yield the largest determination coefficient $R^2_{y,x_{i_1},\ldots,x_{i_k}}$ being the square of the multiple correlation coefficient between the variable y and the variables $x_{i_1}, \ldots, x_{i_k}$. Instead of performing an all-subset search we can use a branch and bound algorithm which often permits us to omit the evaluations of the $R^2_{y,x_{i_1},\ldots,x_{i_k}}$ statistic for a considerable number of subsets.

We describe an algorithm for finding the best subset using a branch and bound method.

Executing the calculations for pseudo-randomly generated data with an assumed structure we indicate the circumstances when, applying the branch and bound algorithm, we might save much time of computing.

1. Preliminary definitions, notation and statements. Suppose we have p predictor variables $x_1, x_2, \ldots, x_p$ from which we want to predict the variable y called in the sequel the *predicted variable.*

The goodness of prediction of the variable y by the variables $x_1, x_2, \ldots, x_p$ is measured by the square of the multiple correlation coefficient $R^2_{y(1,2,\ldots,p)}$, called also the *coefficient of determination.*

Suppose we have n realizations of the considered predictor and predicted variables. The i-th realization $(i = 1, 2, \ldots, n)$ is denoted by $x_{i1}, x_{i2}, \ldots, x_{ip}, y_i$.

Let SS_y denote the adjusted sum of squares of observed values of the predicted variable y:

$$\mathrm{SS}_y = \sum_{i=1}^{n} (y_i - \bar{y})^2. \tag{1}$$

Using the method of least squares we can establish a linear function

$$\hat{y} = b_0 + b_1 x_1 + \ldots + b_p x_p$$

which, when applied to our data, gives the minimal value of the residual sum of squares:

$$\mathrm{SS}(1, 2, \ldots, p) = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_{i1} - \ldots - b_p x_{ip})^2. \tag{2}$$

The multiple correlation coefficient $R^2_{y(1,2,\ldots,p)}$ can be calculated as

$$R^2_{y(1,2,\ldots,p)} = 1 - \mathrm{SS}(1, 2, \ldots, p)/\mathrm{SS}_y. \tag{3}$$

The smaller the residual sum of squares, the better the approximation of y by $x_1, x_2, \ldots, x_p$.

The starting point for all calculations is the adjusted cross-product matrix $A = (a_{ij})$, $i = 1, 2, \ldots, p+1$, $j = 1, 2, \ldots, i$, with elements defined as follows:

$$a_{rs} = \begin{cases} \sum_{l=1}^{n} (x_{lr} - \bar{x}_{.r})(x_{ls} - \bar{x}_{.s}), & r, s \neq p+1, \\ \sum_{l=1}^{n} (y_l - \bar{y})(x_{ls} - \bar{x}_{.s}), & r = p+1,\ s \neq p+1, \\ \sum_{l=1}^{n} (y_l - \bar{y})^2, & r = s = p+1. \end{cases} \tag{4}$$

To evaluate the residual sum of squares we do not need necessarily evaluate the regression coefficients $b_0, b_1, \ldots, b_p$. Using the abbreviated Gauss–Jordan pivoting [5], we can obtain directly the residual sum of squares. In the sequel we use this algorithm. It is approximately three times faster than the full Gauss–Jordan algorithm [1]. It can be used also for stepwise calculations with steps performed upwards and backwards – with the restriction that the variables should be removed from the regression set in the reverse order as they entered it.

In the sequel we use the following property of the residual sum of squares:

PROPERTY 1. *Suppose we have already calculated the residual sum of squares for a set of h variables $x_1, x_2, \ldots, x_h$. After augmenting this set to a set of m variables $x_1, x_2, \ldots, x_m$, $m > h$, we obtain the residual sum of squares for m variables $x_1, x_2, \ldots, x_m$. The residual sum of squares obtained for regression with a greater number of variables cannot be larger than that obtained for a subset of those variables:*

$$\mathrm{SS}(1, 2, \ldots, h) \geqslant \mathrm{SS}(1, 2, \ldots, m), \quad m > h. \tag{5}$$

This property is essential for further considerations.

Now we determine a special notation for the residual sum of squares obtained when introducing all but one out of p variables into the regression set:

$$Q(-r) = \mathrm{SS}(1, 2, \ldots, r-1, r+1, \ldots, p), \quad r = 1, 2, \ldots, p. \tag{6}$$

Thus $Q(-r)$ is the residual sum of squares after introducing all but the r-th variable into the regression set.

2. The algorithm. To find the best subset of size k we should evaluate $\binom{p}{k}$ subsets and for each of them evaluate the residual sum of squares. Generating the subsets in a special way (the next subset is obtained from the preceding one by the exchange of one variable), we can save a considerable amount of calculations when pivoting out only one variable from the regression set and pivoting in another variable. Say, we seek for the best k variables out of p. It is easy to see that all subsets of size k out of p can be divided into $k+1$ subgroups (called in the sequel also *branches*):

The $(k+1)$-st branch comprises only one set: $1, 2, \ldots, k$.

The j-th branch $(1 \leqslant j \leqslant k)$ comprises sets with the following structure: first the integers $1, 2, \ldots, j-1$, and next $k-j+1$ integers chosen from the set $\{j+1, \ldots, p\}$.

The first branch comprises all k-tuples which can be chosen from the integers $2, 3, \ldots, p$.

It follows that the number of subsets to be investigated in the j-th branch, $1 \leqslant j \leqslant k+1$, is $\binom{p-j}{k-j+1}$. An example for $p = 7$, $k = 3$ is given in Table 1.

TABLE 1. The branches for subsets of size $k = 3$ which can be chosen from $p = 7$ variables

Branch No.	1	2	3	4
Number of subsets in the branch	$\binom{6}{3}$	$\binom{5}{2}$	$\binom{4}{1}$	$\binom{3}{0}$
The subsets	234	**134**	**124**	**123**
	235	**135**	**125**	
	236	**136**	**126**	
	237	**137**	**127**	
	245	**145**		
	246	**146**		
	247	**147**		
	256	**156**		
	257	**157**		
	267	**167**		
	345			
	346			
	347			
	356			
	357			
	367			
	456			
	457			
	467			
	567			

Our clue is that the numeration of the variables is not arbitrary. It is chosen in a special way. We proceed as follows. First for $j = 1, 2, \ldots, p$ we evaluate the residual sums of squares $Q(-1), Q(-2), \ldots, Q(-p)$ defined by (6). Next we order them in descending magnitude. The resulting sequence is

$$Q(-j_1) \geqslant Q(-j_2) \geqslant \ldots \geqslant Q(-j_p). \tag{7}$$

We relabel the variables according to (7).

From now up to the end of the paper we assume that the numeration (order) of the variables is such that

$$Q(-1) \geqslant Q(-2) \geqslant \ldots \geqslant Q(-p). \tag{8}$$

Judging from (8) we could say that variable No. 1 has the greatest importance when approximating y by $x_1, x_2, \ldots, x_p$: excluding variable No. 1 we obtain the largest residual sum of squares, hence the worst approximation. Conversely, excluding variable No. p, we obtain the smallest residual sum of squares, which means that the $(p-1)$-tuples of variables without x_p are better than the $(p-1)$-tuples with x_p. It follows that in some way the variable x_p is the worst for evaluating y from the remaining $p-1$ predictor variables.

Continuing this argumentation we assume that the order of variables in (8) is close to their importance for predicting y.

The quadratic forms $Q(-1), Q(-2), \ldots, Q(-p)$ impose definite bounds for residual sums of squares evaluated for subsets of variables which can be obtained when considering all predictor variables but the j-th one $(j = 1, 2, \ldots, p)$.

Specifically, no subset chosen from the variables $2, 3, \ldots, p$, when introduced into the regression set, can give the residual sum of squares smaller than $Q(-1)$. Removing some further variables from the set $\{2, 3, \ldots, p\}$ we can possibly increase the residual sum of squares, but never decrease it.

We start the calculations with the evaluation of the residual sum of squares $\mathrm{SS}(1, 2, \ldots, k)$. This is simultaneously the residual sum of squares for the branch No. $k+1$. We take this residual sum of squares as the current minimum SS_0.

Suppose that for some j_0, $1 \leqslant j_0 \leqslant k$, after considering all subsets belonging to the branches $j_0, j_0+1, \ldots, k+1$, the current minimum is SS_0. Suppose further that this SS_0 is smaller than $Q(-(j_0-1))$.

Using Property 1 we are sure that it is useless to investigate the subsets belonging to the branches $j_0-1, j_0-2, \ldots, 1$. Any subset from these branches will give a residual sum of squares larger than the already obtained value SS_0. So we have only to investigate the subsets from the branches $k, k-1, \ldots, j_0$.

Now suppose we are investigating the h-th branch $(j_0 \leqslant h \leqslant k)$. The structure of subsets belonging to this branch is such that at the first place we have the integers $1, 2, \ldots, h-1$, and at the remaining $k-h+1$ places the integers chosen from the set $\{h+1, h+2, \ldots, p\}$. Together there are $\binom{p-h}{k-h+1}$

such subsets. We evaluate the residual sum of squares for all these subsets sequentially and retain the indices and the residual sum of squares for that subset which has the residual sum of squares smaller than the current SS_0. If we find such a subset, we relabel the optimal (found up to this moment) set and the current value SS_0. All subsets from the h-th branch considered, we check the inequality $(h > 1)$

$$SS_0 \leqslant Q(-(h-1)). \tag{9}$$

If this holds, we finish our calculations (we have found the optimal subset), otherwise we proceed evaluating further subsets and further branches.

3. An example of calculations. We use here a part of data presented in the paper by Liebhart et al. [6]. The predicted variable y is TLC (total lung capacity). The predictor variables are: age of the patient (x_1), height of the patient (x_2), and some simple spirometric values such as

$$\text{VC } (x_3),\ \text{FEV}_1\ (x_4),\ \text{FEF } (x_5),\ \text{MMRF/MMFT } (x_6),\ \text{FEF/VC } (x_7).$$

We consider data obtained for the control group (adults with no diagnosed respiratory disease) comprising $n = 28$ individuals (in fact, the control group considered in [6] was enlarged by adding 11 new individuals). The adjusted cross-product matrix is given in Table 2.

We want to find $k = 3$ predictor variables which give the largest multiple correlation coefficient R^2 with the predicted variable y.

According to the algorithm described in Section 2, we evaluate the

TABLE 2. Adjusted cross-product matrix for the considered example of predicting TLC (y) from age (x_1), height (x_2) and some spirometric values ($x_3, \ldots, x_7$)

Variable	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y
x_1	3281.4							
x_2	−483.5	1830.4						
x_3	−39744.2	154320.7	19088867.8					
x_4	−52788.5	122171.4	13821485.7	11675571.4				
x_5	82.2	14158.0	1567321.1	1444394.2	369238.3			
x_6	−530168.4	1027861.5	117016567.7	134574345.8	30001623.5	4244512806.2		
x_7	742.6	441.8	13948.8	76491.5	55134.8	4550028.0	12580.9	
y	−27964.9	227342.6	24047117.4	18275169.2	2346617.5	158672548.7	102292.8	35227468.6
Means	41.1429	163.6429	3851.0714	2927.1429	287.8286	19817.1888	74.6105	5414.9249

residual sums of squares $Q(-1), Q(-2), \ldots, Q(-7)$. Dividing each of them by

$$\mathrm{SS}_y = \sum_{i=1}^{n} (y_i - \bar{y})^2,$$

we obtain the standardized residual sums of squares $\tilde{Q}(-1)$, $\tilde{Q}(-2), \ldots, \tilde{Q}(-7)$. Ordering them from the largest to the smallest one, we obtain the sequence

$$\tilde{Q}(-2) \geqslant \tilde{Q}(-3) \geqslant \tilde{Q}(-1) \geqslant \tilde{Q}(-6) \geqslant \tilde{Q}(-4) \geqslant \tilde{Q}(-5) \geqslant \tilde{Q}(-7). \tag{10}$$

The appropriate values of these standardized residuals are

$$\tilde{Q}(-2) = 0.097711, \quad \tilde{Q}(-3) = 0.071971, \quad \tilde{Q}(-1) = 0.071132,$$
$$\tilde{Q}(-6) = 0.069851, \quad \tilde{Q}(-4) = 0.067922, \quad \tilde{Q}(-5) = 0.067922,$$
$$\tilde{Q}(-7) = 0.064342.$$

The new ordering of variables according to (10) is

new order	1	2	3	4	5	6	7
old order	2	3	1	6	4	5	7

The search for the optimal subset is presented in Table 3.

The starting set is the set $\{1, 2, 3\}$ (in the old numeration the set $\{2, 3, 1\}$). The standardized residual sum of squares for this subset equals

$$\widetilde{\mathrm{SS}}_0 = \widetilde{\mathrm{SS}}(1, 2, 3) = 0.0784$$

and is larger than $\tilde{Q}(-3) = 0.071132$.

It follows that we should investigate the branch No. 3. Here we consider four subsets. One of them (the subset $\{1, 2, 6\}$ in the new numeration) gives the standardized residual sum smaller than the actual $\widetilde{\mathrm{SS}}_0$, therefore we retain the subset $\{1, 2, 6\}$ as the actually best set with the relabelled value $\widetilde{\mathrm{SS}}_0 = 0.0773$. This value is still larger than $\tilde{Q}(-2) = 0.071971$, therefore we should investigate the subsets belonging to the branch No. 2. Now we have to investigate 10 subsets. None of them gives a standardized residual sum of squares smaller than $\widetilde{\mathrm{SS}}_0 = 0.0773$. It follows that the subset $\{1, 2, 6\}$ (in the new numeration) remains the actually optimal subset.

Before proceeding to evaluate the subset belonging to the branch No. 1 we check the inequality

$$\widetilde{\mathrm{SS}}_0 < \tilde{Q}(-1).$$

We obtain $0.0773 < 0.097711$, and hence this inequality is satisfied.

It follows that every subset belonging to the branch No. 1 will give a standardized residual sum of squares larger than 0.097711, therefore we cannot obtain here a residual smaller than the actual residual $\widetilde{\mathrm{SS}}_0$. Hence our search is terminated.

In this example, instead of considering 35 subsets, we needed to evaluate only 15 subsets to find the optimal one.

TABLE 3. An example of search for the optimal subset of size $k = 3$ out of $p = 7$ variables

Variables in the subset i_1, i_2, i_3		Standardized residual sum of squares $\widetilde{\mathrm{SS}}(i_1, i_2, i_3)$	Multiple correlation coefficient $R^2_{y(x_{i_1}, x_{i_2}, x_{i_3})}$
old numeration	new numeration		
Branch No. 4 comprising the starting set			
231	123	0.0784	0.9216
Branch No. 3 – has the bound $\tilde{Q}(-3) = 0.071132$			
236	124	0.0869	0.9131
234	125	0.0858	0.9142
235	126	0.0773*	0.9227*
236	127	0.0869	0.9131
Branch No. 2 – has the bound $\tilde{Q}(-2) = 0.071971$			
216	134	0.1804	0.8196
214	135	0.0974	0.9026
215	136	0.1556	0.8444
217	137	0.1857	0.8143
264	145	0.1124	0.8876
265	146	0.1650	0.8450
267	147	0.1905	0.8095
245	156	0.1176	0.8824
247	157	0.1732	0.8268
257	167	0.0973	0.9027
Branch No. 1 – has the bound $\tilde{Q}(1) = 0.097711$			
316	234	0.1343	0.8657
314	235	0.1155	0.8845
315	236	0.1211	0.8789
317	237	0.1213	0.8787
364	245	0.1244	0.8756
365	246	0.1172	0.8828
367	247	0.1187	0.8813
345	256	0.1204	0.8796
347	257	0.1201	0.8799
357	267	0.1236	0.8764
164	345	0.1301	0.8699
165	346	0.5474	0.4526
167	347	0.8146	0.1854
145	356	0.1599	0.8401
147	357	0.1568	0.8432
357	367	0.1236	0.8764
645	456	0.1326	0.8674
647	457	0.1481	0.8519
657	467	0.1625	0.8375
457	567	0.1507	0.8493

4. Experience with some generated data. We carried out two experiments with generated data. In the sequel we describe these experiments and then compare their results.

4.1. *The first experiment.* In the first experiment we generated p pseudo-random values from the uniform distribution (0, 1). Thus we obtained realization of the variables $x_1, x_2, \ldots, x_p$. Next we generated another pseudo-random value u from the uniform distribution $U(0, 1)$. The value for the "dependent" variable y was then calculated as follows:

$$y = x_1 + x_2 + \ldots + x_{p/2} + u\alpha. \tag{11}$$

In our experiment we considered $\alpha = 3$.

Repeating this procedure $n = 50$ times we obtained the data arrays $X[1{:}50, 1{:}p]$ and $Y[1{:}50]$ comprising n artificial realizations of the variables $x_1, x_2, \ldots, x_p, y$. Note that in fact the random variable y depends only on the first $p/2$ variables $x_1, x_2, \ldots, x_{p/2}$.

The covariance matrices calculated for the generated data were the starting point for the all-subset search. The CPU times of computing on the ODRA 1305 computer using the algorithm described in Section 2 are given in Table 4. In the same table, times needed by an algorithm using the full all-subset search, described, e.g., by Bartkowiak [1] are shown in brackets.

TABLE 4. Average CPU times (in minutes) of computing, on the ODRA 1305 computer, the best subset using a branch and bound method and times of computing using an all-subset search (in brackets). p is the number of considered variables, k is the size of the subset

p \ k	1÷4	5÷7	8÷11	12÷15
Experiment 1				
8	0.02 [0.03]	0.02 [0.03]	–	–
12	0.11 [0.11]	0.05 [0.33]	0.05 [0.21]	–
16	0.52 [0.37]	1.31 [4.39]	0.41 [9.13]	0.13 [3.02]
Experiment 2. Helmert matrices				
8	0.03 [0.03]	0.02 [0.03]	–	–
12	0.14 [0.11]	0.32 [0.33]	0.05 [0.21]	–
16	0.61 [0.37]	3.50 [4.39]	2.94 [9.13]	0.15 [3.02]

The CPU times presented in Table 4 are means from calculations for 10 matrices. We assumed consecutively $p = 8, 12, 16$. Next we have been searching for an optimal subset of size k, with k in 3 groups: $1 \leqslant k \leqslant 4$, $5 \leqslant k \leqslant 7$, and $8 \leqslant k \leqslant 12$.

In Table 4 one can see that for $p = 8$ the search of the best subset is very quick and the gain in computing time is small. For $p = 12$ we have no gain when searching a subset of size $1 \leqslant k \leqslant 4$, and a substantial gain when searching a subset of size $5 \leqslant k \leqslant 7$ (0.33 mins. by the traditional method, 0.05 mins. by the new method) and of size $8 \leqslant k \leqslant 11$ (0.21 mins. by the traditional method, 0.05 mins. by the new method).

For $p = 16$, when searching for a subset of size $1 \leqslant k \leqslant 4$, using the new method we need even more time (we have here to evaluate all subsets and perform some additional calculations to evaluate the bounds for the branches and introduce the new order of variables). A substantial gain in CPU time is obtained when searching for subsets of size $k > 5$.

4.2. *The second experiment.* In the second experiment we constructed the covariance matrices using Helmert matrices. We say that H is a *Helmert matrix* if for a given p its elements are determined by the formula ([8], p. 33)

$$H_{p\times p} = \begin{bmatrix} h' \\ H_0 \end{bmatrix} \tag{12}$$

with h' being the first row (of dimension $1 \times p$) of $H_{p\times p}$, $h' = (1/\sqrt{p})1'_p$, where $1'_p = [1, 1, \dots, 1]$ is a vector of p's, and with H_0 being the last $(p-1)$ rows of $H_{p\times p}$, while the r-th row of H_0 takes the form

$$\left[\frac{1}{\sqrt{r(r+1)}}1'_r \quad \frac{-r}{\sqrt{r(r+1)}}0_{(p-r-1)\times 1}\right] \quad \text{for } r = 1, 2, \dots, p-1.$$

It can be shown that H is orthogonal.

Using a Helmert matrix H and a diagonal matrix D,

$$D = \operatorname{diag}(d_1, d_2, \dots, d_p),$$

we construct the matrix C:

$$C = HDH'. \tag{13}$$

It is easy to prove that the matrix C constructed according to formula (13) is symmetric:

$$C' = (HDH')' = HD'H' = HDH = C.$$

One can see an analogy between formula (1) and the spectral decomposition of a square matrix C:

$$C = A\Lambda A, \tag{14}$$

where $A = (a_1, a_2, \dots, a_p)$ are the eigenvectors and

$$\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$$

are the eigenvalues of the matrix C.

Assuming nonnegative values of $D = \operatorname{diag}(d_1, d_2, \dots, d_p)$ and constructing the matrix C according to formula (13) we obtain a matrix which is nonnegative definite.

The elements of D may be given values in descending order. They may be interpreted as eigenvalues of C with $H = (h_1, h_2, \ldots, h_p)$ as eigenvectors.

After normalizing the matrix H (or A) to comprise orthonormal vectors we have the equality (see, e.g., [7])

$$\text{(15a)} \qquad \operatorname{tr}(C) = \sum_{i=1}^{p} \lambda_i$$

or

$$\text{(15b)} \qquad \operatorname{tr}(C) = \sum_{i=1}^{p} d_i.$$

Equality (11) or (12) means that the sum of the diagonal elements of the matrix C can be reproduced using the eigenvalues and eigenvectors of this matrix. Specifically, according to (15a) or (15b), the sum of the diagonal elements $c_{11}, c_{22}, \ldots, c_{pp}$ can be reproduced (explained) by the diagonal elements $\lambda_1, \lambda_2, \ldots, \lambda_p$ or $d_1, d_2, \ldots, d_p$. If the considered variables are interdependent, and the reason for the interdependence is that there exist a few number of factors (principal components) which cause the interdependence among the variables, then a few vectors with their eigenvalues often suffice to reproduce the matrix C quite good.

Assuming different values of $d_1, d_2, \ldots, d_p$ and using appropriate Helmert matrices, we can construct matrices with various interdependence structure.

In our second experiment we assumed $d_i = p-i+1$ for $i = 1, 2, \ldots, p$. Then we calculated the auxiliary matrix $T = HD^{1/2}$.

Next we generated realizations of a p-variate random variable according to the following rule of constructing observations with a given covariance matrix C, presented, e.g., in [9] or [2]:

1° Find a matrix T such that $TT' = C$.

2° Generate p independent pseudo-random values

$$z = (z_1, z_2, \ldots, z_p)$$

from the normal distribution $N(0, 1)$.

3° Compute $x = (x_1, x_2, \ldots, x_p)$ as $x = Tz$.

After obtaining $x_1, x_2, \ldots, x_p$ according to points 1°, 2°, 3° presented above, we constructed y in the usual way using formula (11).

Proceeding in this way we generated 10 groups of data matrices, each of them comprising 50 "observations" $x_1, x_2, \ldots, x_p, y$ constructed using the rules presented above. From these data we calculated the covariance matrices. Next we have been searching for the best subset. The average CPU times needed for finding the optimal subset of size k are given again in Table 4.

One can see that for $p = 12$ a substantial gain is obtained in the last class of the values of k $(8 \leqslant k \leqslant 11)$. For $p = 16$ we obtained a substantial gain when searching for a subset of size $k \geqslant 8$: for $8 \leqslant k \leqslant 11$ the CPU time decreased from 9.13 mins. (the traditional method) to 2.94 mins. (the new method).

Similarly, searching for a subset of size $12 \leqslant k \leqslant 15$ we needed 3.02 mins. using the traditional method and only 0.15 mins. using the new method.

4.3. *Comparison of results.* It happened that the internal interdependence structure between the variables considered in experiment No. 1 and experiment No. 2 is similar. This may be seen by inspecting the diagrams exhibiting the percents of exhaustion of the diagonal diag(C) by successive eigenvalues of this matrix. Fig. 1 shows these percentages for the first matrices from each experiment. The similarity of the interdependence structure is the reason for the similarity of the CPU time obtained for various variants of the calculations.

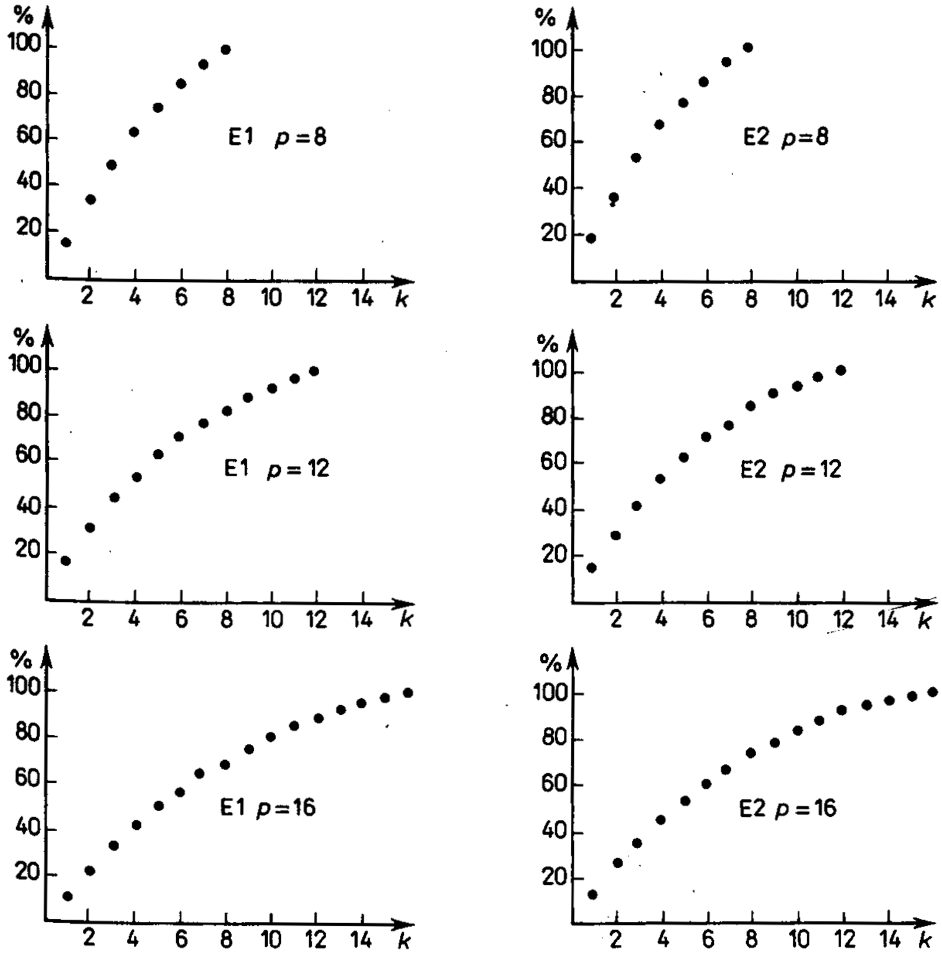

Fig. 1. Percentage of exhaustion of the diagonal of the covariance matrix by the first k eigenvalues

E1 – experiment No. 1, E2 – experiment No. 2, p – number of considered variables

5. Final conclusions and remarks. Using the branch and bound method we can expect a substantial gain in CPU time when the predicted variable y depends on fewer than p variables.

All the considerations concerned with the best subset search in regression analysis can be applied immediately to the best subset search in discriminant analysis with 2 groups of data, provided that an optimal subset is defined as such one that gives the largest Mahalanobis distance between these two groups. An example of the application of this principle in discriminant analysis is shown in [3].

Acknowledgement. We thank Dr. J. Liebhart and Dr. E. Liebhart for allowing us to use their data in the example shown in Section 3.

References

[1] A. Bartkowiak, *SABA, An Algol Package for Statistical Data Analysis on the ODRA 1305 Computer*, Universitas Wratislaviensis, Wrocław 1984.

[2] —and E. Krusińska, *SABA, A Description of Statistical Programs for the ODRA 1305 Computer*, Vol. II (in Polish), Universitas Wratislaviensis, Wrocław 1986.

[3] A. Bartkowiak, S. Łukasik, K. Chwistecki and M. Mrukowicz, *Search for most discriminative features for CHD using a branch and bound method*, paper prepared for the 4th IMEKO conference "Advances in Biomedical Measurements", Bratislava, 20–24 May 1987.

[4] R. R. Hocking, *Selection of the best subset of regression variables*, pp. 39–57 in: K. Enslein, A. Ralston and H. S. Wilf (eds.), *Statistical Methods for Digital Computers*, J. Wiley 1977.

[5] R. I. Jennrich, *Stepwise regression*, pp. 58–75 in: K. Enslein, A. Ralston and H. S. Wilf (eds.), *Statistical Methods for Digital Computers*, J. Wiley 1977.

[6] J. Liebhart, Z. Karkowski, E. Krusińska, E. Liebhart and J. Małolepszy, *Determination of total lung capacity (TLC) using data from forced expiratory flow volume curves and simple measurements of the chest* (in Polish), Pneumonologia Polska 53 (1985), pp. 520–526.

[7] D. F. Morrison, *Multivariate Statistical Methods*, McGraw-Hill, New York 1967.

[8] S. R. Searle, *Linear Models*, J. Wiley, New York 1971.

[9] R. Zieliński, *Generators of Random Numbers* (in Polish), Wydawnictwa Naukowo-Techniczne, Warszawa 1979.

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF WROCŁAW
51-151 WROCŁAW

Received on 1987.06.05