Aplikace matematiky

Vilém Novák

An algorithm for reduction of complexity of relations in a system of variables

Aplikace matematiky, Vol. 22 (1977), No. 6, 418–424

Persistent URL: http://dml.cz/dmlcz/103718

AN ALGORITHM FOR REDUCTION OF COMPLEXITY OF RELATIONS IN A SYSTEM OF VARIABLES

VILÉM NOVÁK

(Received July 8, 1976)

1. INTRODUCTION

Assume we are given a great number of parameters describing some real process. The parameters can change their values (in time) and so we can interpret them as variables. Our aim is to decide, whether some relations exist among the parameters and to express these relations in the form of concrete dependences generally of one variable on one or more others (e.g. by means of regression analysis). In the case of many independent variables in these dependences we meet great computational difficulties even when using contemporary computers. Therefore it may be useful to carry out a sort of *pre-research*, the aim of which is the reduction of all independent variables in each dependence studied. The following algorithm was developed for this purpose. It can be of use in many concrete problems, especially for the study of economic systems.

2. THE PROBLEM OF THE DEPENDENCE

Assume we are given n variables (parameters) $\mathscr{X} = \{X_1, X_2, \ldots, X_n\}$ (n is equal 15 to 20 at least). Let us consider two variables X_i and X_j, $1 \leqq i, j \leqq n$. What does it mean: *one variable depends on another one?* We shall distinguish three cases:

1. If the variables are *random*, then the change of one, e.g. X_j, influences the change of X_i with a certain probability. Stochastic dependence may exist between them in this case. Using the regression analysis we interlay a regression line through the set of points with coordinates X_i and X_j. This line is considered a functional representation of the dependence of the variable X_i on X_j with the corresponding degree of significance. We substitute stochastic dependence by functional dependence.

2. If the variables are *non-random*, then functional dependence may exist between them. It can be approximated e.g. by an interpolation polynomial.

3. Knowing a concrete meaning of each individual variable we can infer the dependence also *intuitively*. The intuition, however, follows from the experience and knowledge of the process studied and it is contiguous to the idea that the change of one variable influences the change of the other.

In the following text we shall understand one of the above mentioned types of dependences if we write that the variables are dependent. The decision whether the dependence exists or not, is made on the basis of an intellectual experiment.
We make the following presumptions:

Presumption 1. We can determine whether the variable X_i depends on X_j or not (in one of the above three meanings).

The fact that the variable X_i depends on X_j is denoted by

$$X_i \, \mathfrak{f} \, X_j \, .$$

Presumption 2. The dependence $\mathfrak{f}$ between the variables X_i, X_j is a relation. We assume that the relation $\mathfrak{f}$ has the following properties:

(1) *Reflexivity:*

$$\forall_{X_i \in X} (X_i \, \mathfrak{f} \, X_i)$$

(2) *Transitivity*: If $X_i \, \mathfrak{f} \, X_j$ and $X_j \, \mathfrak{f} \, X_k$, then $X_i \, \mathfrak{f} \, X_k$.

(3) *Antisymmetry*: If $X_i \, \mathfrak{f} \, X_j$ and $X_j \, \mathfrak{f} \, X_i$, then $X_i \, \mathfrak{b} \, X_j$. The relation $\mathfrak{b}$ means the variables X_i and X_j are "bilaterally dependent" (interchangeable), so $\mathfrak{b}$ has the properties of an equivalence.

We can remark that the relation $\mathfrak{f}$ is an ordering in the set $\mathscr{X}$. The set $\mathscr{X}$ is *a partially ordered set.*

During the pre-research we consider succesively each pair of the variables X_i, $X_j \in \mathscr{X}$. At the start, we have not usually a sufficient amount of information about the nature of possible relations among the variables studied. So we do not know whether Presumptions 1, 2 are fulfilled. During the pre-research it is useful to accept the hypothesis they are fulfilled. The conclusion and eventual corrections can be made, of course, when the research proper and its detailed analysis are finished.

The main idea of the proposed algorithm is based on the following lemma:

Lemma 1. *If*

$$X_i \, \mathfrak{f} \, X_l$$

and

$$X_l \, \mathfrak{f} \, X_{j1} \, , \quad X_l \, \mathfrak{f} \, X_{j_2}, \ldots, X_l \, \mathfrak{f} \, X_{j_k} \, , \quad k < n \, ,$$

then

$$X_i \, \mathfrak{f} \, X_{j_1} \, , \quad X_i \, \mathfrak{f} \, X_{j_2}, \ldots, X_i \, \mathfrak{f} \, X_{j_k} \, .$$

The proof is evident.

Corollary 1. *Let*

$$X_i \mathfrak{f} X_{j_1}, \quad X_i \mathfrak{f} X_{j_2}, \ldots, X_i \mathfrak{f} X_{jk}$$

and

$$X_i \mathfrak{b} X_l$$

Then

$$X_l \mathfrak{f} X_{j_1}, \quad X_l \mathfrak{f} X_{j_2}, \ldots, X_l \mathfrak{f} X_{j_k}.$$

We shall write

$$X_i \mathfrak{f} (X_{j_1}, X_{j_2}, \ldots, X_{j_k})$$

instead of

$$X_i \mathfrak{f} X_{j_1}, \quad X_i \mathfrak{f} X_{j_2}, \ldots . X_i \mathfrak{f} X_{j_k}.$$

Our algorithm is based on the following observation: The fact that $X_l \mathfrak{f} (X_{j_1}, X_{j_2}, \ldots$ $\ldots, X_{j_k})$ and $X_i \mathfrak{f} (X_l, X_{j_1}, X_{j_2}, \ldots, X_{j_k})$ is fully represented by the dependences

$$X_l \mathfrak{f} (X_{j_1}, X_{j_2}, \ldots, X_{j_k})$$

and

$$X_i \mathfrak{f} (X_l).$$

This follows immediately from Lemma 1.

The algorithm uses the zero-unit matrix of the order $n \times n$. We construct it in the following way: Inscribe

a) unit, if $X_i \mathfrak{f} X_j$ holds,

b) zero, if $X_i \mathfrak{f} X_j$ does not hold

into the square (i, j), $i, j = 1, 2, \ldots, n$. We call this matrix the *matrix of cross relations.* It is our aim now to simplify this matrix in order that the number of units in every row might be as small as possible.

3. THE DESCRIPTION OF THE ALGORITHM

3.1. *Verbal description*

1. Inscribe unit into the square (i, j), if $X_i \mathfrak{f} X_j$. Inscribe zero in the opposite case (into the main diagonal inscribe units because of reflexivity of the relation $\mathfrak{f}$).

2. Make column S, elements S_i of which contain the row-sums. Number S_i means that the variable X_i is a function of $S_i - 1$ other variables. If $S_i = 1$, then the variable X_i is "pure independent". Mark it and cancel the i-th row.

3. Make row Z of column-sums. Number Z_j means that the variable X_j is in $Z_j - 1$ relations. If $Z_j = 1$, then the variable X_j is "pure dependent".

4. Find all units placed symmetrically round the main diagonal. The corresponding variables X_i, X_j are bilaterally dependent. Mark them. The rows i, j (as well as the columns i, j) must coincide.

5. Find a row i where S_i is minimal. In the case there are more S_i of the same value, take S_i with minimal i.

6. The corresponding variable X_i is a function of all variables X_{j_k}, where the square (i, j_k) contains a unit (we omit, of course, the square (i, i)). Write this dependence and cancel the i-th row.

7. Find all rows l with a unit in the square (l, i).

8. Cancel units from k squares of the row (l, j_k) (subscripts j_k are the same as in item 6). If there is zero in a certain square (l, j_k) and this unit was not canceled before, a mistake is signalized, because the variable X_l depends on all the variables X_{j_k} on which X_i depends.

10. If there are some uncanceled rows, then go back to the item 5.

3.2. *The program*

The algorithm was programmed in Algol 60. The reading row by row of the matrix of cross relations is supposed. The procedure *print* is determined for output of string (in string brackets) or number. The number m is the number of bilaterally dependent variables:

$$m \leqq \binom{n}{2}.$$

The program prints pure independent variables, bilaterally dependent variables and the new defined relations in the form:

$$X_i \,\mathfrak{f}\, (X_{j_1}, X_{j_2}, \ldots, X_{j_k}).$$

The program prints "mistake" and stops in the case of an "absent unit" described in item 8.

```
procedure redcompl (m, n);
integer m, n;
comment m was described before, n is the number of variables;
begin boolean array A, B[1 : n, 1 : n];
    integer array S, Z[1 : n], P[1 : m]; integer i, j, k, l;
    procedure print;
procedure min (S, k);
integer array S; integer k;
begin integer t, r;
  k := 1; t := S[k];
  for r := 2 step 1 until n do if t > S[r] then begin t := S[r];
      k := r end; S[k] := n + 1
end;
comment the boolean array A represents the matrix of cross relations. It is filled
          using the parameter k, values of which are zero or unit;
```

```
for i := 1 step 1 until n do for j := 1 step 1 until n do
  A[i,j] := if k = 1 then true else false;
k := 0; l := 1; for i := 1 step 1 until n do S[i] := 0;
for i := 1 step 1 until n do
begin for j := 1 step 1 until n do
  begin S[i] := (if A[i,j] then 1 else 0) + S[i];
    if i = j then B[i,j] := false else B[i,j] := true;
    if (i < j) ∧ A[i,j] ∧ A[j,i] then
      begin P[l] := i; P[l + 1] := j; l := l + 2 end;
  end;
  if S[i] = 1 then begin k := k + 1; Z[k] := i; S[i] := n + 1 end;
end;
if k ≠ 0 then begin print ('INDEPENDENT VARIABLES :');
  for j := 1 step 1 until k do print ('X', Z[j])
        end;
if l ≠ 1 then
begin print ('BILATERALLY DEPENDENT VARIABLES :');
  for i := 2 step 2 until l − 1 do print ('X', P[i], 'X', P[i + 1]);
  for i := 1 step 2 until l − 1 do for j := 1 step 1 until n do
    if ¬ (A[P[i],j] ≡ A[P[i + 1],j]) then
      begin print ('VARIABLES X', P[i], 'X', P[i + 1], 'DIFFER
        IN LOCATION OF UNITS'); go to L 3;
      end;
end;
L 1 : min (S, k); print ('X', k, '= f(');
for j := 1 step 1 until n do if A[k,j] ∧ (j ≠ k) then
  begin print ('X', j, ','); B[k,j] := false end; print (')');
for i := 1 step 1 until n do if (S[i] ≠ n + 1) ∧ A[i, k] then
begin B[i, k] := false; j := 1;
  V 1; if j = k then j := j + 1; if ¬ B[k, j] then
    begin if (i ≠ j) ∧ ¬ A[i,j] ∧ B[i,j] then
      begin print ('IN THE SQUARE', i, j, 'THERE IS AN ABSENT UNIT');
        go to L3
      end else if (i ≠ j) ∧ A[i,j] then
    begin A[i,j] := false; B[i,j] := false; S[i] := S[i] − 1
    end;
    end; j := j + 1; if j ≦ n then go to V 1;
end; i := 1;
L2 : if S[i] = n + 1 then
begin i := i + 1; if i > n then go to L3 else go to L2 end
else go to L1;
```

comment the element of the boolean array B is now false, since the corresponding square of the matrix of cross relations contains unit. The array B can be used for output of the whole matrix of cross relations;

*L*3 : **end** *redcompl*;

A simple example with a matrix of order seven is presented. The computer Odra 1204 was used for the computation.

	1	2	3	4	5	6	7	S
1	1	1	1	0	1	1	0	5
2	1	1	1	0	1	1	0	5
3	0	0	1	0	0	1	0	2
4	1	1	1	1	1	1	0	6
5	0	0	0	0	1	1	0	2
6	0	0	0	0	0	1	0	1
7	1	1	1	1	1	1	1	7
Z	4	4	5	2	5	7	1	

By means of the above mentioned algorithm we have got the following result:

Independent variable is X_6, bilaterally dependent are X_1, X_2 and $X_3\ \mathfrak{f}\,(X_6)$, $X_5\ \mathfrak{f}\,(X_6)$, $X_1\ \mathfrak{f}\,(X_2, X_3, X_5)$, $X_2\ \mathfrak{f}\,(X_1)$, $X_4\ \mathfrak{f}\,(X_1)$, $X_7\ \mathfrak{f}\,(X_4)$.

We can seen that our situation has been simplified: we have to consider only five pair dependences and one dependence with three independent variables. Originally, we should have to study two pair dependences, two with four, one with five and one with six independent variables. We need much less measurings for further computations than before and the accuracy of results will increase.

The program is relatively less demanding for memory and the computation is very quick.

Acknowledgment. The author wishes to thank RNDr. Petr Hájek, CSc. for his helpful criticism which lead to improvements in this paper.

References

[1] *J. Raichl:* Programming for Computers (Czech). Academia, Praha 1974.

[2] *L. Beran:* Groups and Lattices (Czech). SNTL, Praha 1974.

Souhrn

ALGORITMUS PRO REDUKCI SLOŽITOSTI VZTAHŮ V SYSTÉMU PROMĚNNÝCH

VILÉM NOVÁK

Mějme dán systém proměnných, mezi nimiž existují složité vzájemné závislosti. Za předpokladu, že vztah „X závisí na Y“ je reflexivní a tranzitivní, je v článku navržen jednoduchý algoritmus, který umožňuje vyjádřit všechny závislosti co nejúspornějším způsobem, aniž by došlo ke ztrátě informace.

Author's address: Ing. *Vilém Novák*, OKR, Automatizace řízení, Gregorova 3, 701 00 Ostrava I.