Aplikace matematiky

Zdeněk Režný; Ivan Dylevský

Algorithms. 49. PIECE LIN REGR. An algorithm for fitting discontinuous multiphase linear least-square regression

Aplikace matematiky, Vol. 29 (1984), No. 1, 76--80

Persistent URL: http://dml.cz/dmlcz/104070

ALGORITMY

49. PIECE LIN REGR

AN ALGORITHM FOR FITTING DISCONTINUOUS MULTIPHASE LINEAR LEAST-SQUARE REGRESSION

Ing. ZDENĚK REŽNÝ, CSc., Doc. MUDr. IVAN DYLEVSKÝ, CSc.,

Ústav biofyziky a nukleární medicíny Fakulty všeobecného lékařství Karlovy university, Salmovská 3, 120 00 Praha 2,
Anatomický ústav Fakulty všeobecného lékařství Karlovy university, U nemocnice 3, 128 00 Praha 2

The algorithm suggested in this paper concerns a model of piecewise linear regression of two variables which may be formulated as follows:

Given positive integers n, $r[i]$, k and empirical points $(X[i], Y[i,j])$ $(i = 1, \ldots, n,$ $j = 1, \ldots, r[i]; X[1] < X[2] < \ldots < X[n])$, we consider k intervals $\langle X[m_{i-1} + 1],$ $X[m_i]\rangle$ $(i = 1, \ldots, k;\ m_0 = 0 < m_1 < \ldots < m_k = n)$, and in each of them, independently of the others, the linear regression of Y on X. Naturally, if $m_{i-1} + 1 = m_i$ for some i, then the corresponding linear segment degenerates into a point. The integers $m_1, \ldots, m_{k-1}$, together with the linear segments (or points) within single intervals, are determined so as to minimize the total square error.

Thus, the method presented is adequate for cases of regression relations between two variables, where different conditions in the individual intervals cause changes in the nature of the regression function, including such changes that manifest themselves in the measured data by jumps, this latter occurrence being precluded in the continuous model of multiphase regression (see [1], especially §§ 7.6 and 8.3 and further literature referred to herein). However, the authors met with just this type of case when evaluating growth changes of selected structures of the human embryonic hand. The growth process in the early embryonic period is disproportionate and the individual developing structures attain mutually stable growth and spatial relations only after a relatively long time period. The growth values of the dependent variable, i.e. length, volume, outer boundary of the developing skeleton and muscles of the embryonic hand change by sudden jumps in certain time intervals. These jumps are due to proteosynthetic activity of growing structure, to differentiational capacity of tissues, to nutritional possibilities and to various types of morphogenetical control mechanisms. The piecewise regression model enabled the identification of critical locations of this type in order to subject them to subsequent morphological analysis

[2] and thus proved practically useful, despite its limitation to linearity of segments and to the criterion of non-weighted sum of squares.

We now define the parameters of the procedure. The first four have already been explained. The remaining input parameter *Nupp* is the upper bound to the number k of intervals. The program then repeats the calculation for k starting from one up to a certain *Nm* which is the maximum integer h with the following three properties: i. $h \leq Nupp$; ii. $h < n/2$; iii. by dividing into h intervals it is possible to achieve a reduction in the total square error compared with the optimal division into $h - 1$ intervals. – Further, $SSQ[k]$ is the total square error attained with division into k intervals and $(XX[i, k, j], YY[i, k, j])$ for $i = 1$ and 2 is the left or right boundary point of the j-th linear segment respectively ($j = 1, \ldots, k$), for the equation of which the parameters are given such that $YY[i, k, j] = a1[k, j] \times XX[i, k, j] + a0[k, j]$. – For the degenerate segment, where the boundary points coincide, $a1$ is put equal to zero.

The computation of the linear regression within individual intervals proceeds by the familiar routine. Optimal division into intervals is based on elementary dynamic programming.

```
procedure PIECE LIN REGR(X, Y, r, n, Nupp)
          results: (SSQ, XX, YY, a1, a0, Nm);
value n, Nupp;
integer n, Nupp, Nm; integer array r;
array X, Y, SSQ, XX, YY, a1, a0;
begin
Nm := if Nupp > (n − 1) ÷ 2 then (n − 1) ÷ 2 else Nupp;
if Nm > 0
then
begin
integer i, j, k, m; real a, b;
integer array SUCC[1 : Nm, 0 : n − 1]; array v[0 : 5, 0 : n], S[1 : Nm, 0 : n − 1];
real procedure G(k); integer k; G := v[k, j] − v[k, i];
real procedure bb;
  begin real a, c, d, e;
    c := 1/G(0); b := c × G(3);
    if j = i + 1
    then bb := 0
    else
    begin a := c × G(1); e := c × G(2); d := c × G(4);
      c := 1/(e − a × a); bb := c × (d − a × b);
      b := c × (b × e − a × d)
    end
  end bb;
```

```
real procedure SE;
  begin real a;
    a := bb × G(4); SE := G(5) − a − b × G(3)
  end SE;
for i := 1 step 1 until n do
  begin a := b := 0;
    for j := 1 step 1 until r[i] do
      begin a := a + Y[i, j];
            b := b + Y[i, j] ↑ 2
      end j;
    v[3, i] := a; v[5, i] := b; v[0, i] := r[i];
    for j := 1, 2, 4 do v[j, i] := X[i] × v[j − 1, i]
  end i;
for j := 0 step 1 until 5 do
  begin v[j, 0] := 0;
    for i := 1 step 1 until n do v[j, i] := v[j, i] + v[j, i − 1]
  end j;
j := n;
for i := 0 step 1 until if Nm = 1 then 0 else n − 1 do
  begin S[1, i] := SE; SUCC[1, i] := n
  end;
for k := 2 step 1 until Nm do
  for i := 0 step 1 until if k = Nm then 0 else n − k do
    if SUCC[k − 1, i] > i
    then
    begin
    a := S[k − 1, i]; m := i;
    for j := i + 1 step 1 until n − k + 1 do
      begin b := SE + S[k − 1, j];
        if b < a then begin a := b; m := j end
      end j;
    if m = 0 then begin Nm := k − 1; go to AA end;
    for j := k step 1 until if m = i then Nm else k do
      begin S[j, i] := a; SUCC[j, i] := m end
    end i and k;
AA:
for k := 1 step 1 until Nm do
  begin
  SSQ[k] := S[k, 0]; i := 0;
  for m := 1 step 1 until k do
    begin j := SUCC[k − m + 1, i];
      XX[1, k, m] := X[i + 1]; XX[2, k, m] := X[j];
```

```
        a := a1[k, m] := bb; a0[k, m] := b;
        for i := 1, 2 do YY[i, k, m] := a × XX[i, k, m] + b;
        i := j
      end m
    end k
  end block
end PIECE LIN REGR
```

Check example: Given values according to Tab. 1, the statement *PIECE LIN REGR*(X, Y, r, 12, 3, SSQ, XX, YY, $a1$, $a0$, Nm), where the integer 3 can be replaced by an arbitrary greater integer, leads to the results presented in Tab. 2.

Tab. 1. Input values for the check example.

i	$X[i]$	$r[i]$	$Y[i,j]$ $(j = 1, \ldots, r[i])$				
1	0.1	2	0.8	2.2			
2	0.2	3	1.0	4.3	3.1		
3	0.3	2	4.4	3.8			
4	0.5	4	6.1	7.5	5.6	7.6	
5	0.6	2	11.0	10.2			
6	0.9	4	16.0	20.1	10.5	19.54	
7	1.1	1	15.5				
8	1.2	2	16.5	13.5			
9	1.4	1	14.0				
10	1.6	3	14.9	9.0	15.1		
11	1.9	5	11.8	10.7	14.8	9.3	10.9
12	2.0	5	12.0	8.5	11.5	9.5	13.5

The algorithm has been tested in the symbolic language FORTRAN IV-PLUS [3] and implemented in the Institute of Biophysics and Nuclear Medicine, Faculty of General Medicine, Charles University using the computer PDP 11/34.

References

[1] *G. A. F. Seber:* Linear Regression Analysis. J. Wiley, New York 1977.

[2] *V. Jirček, I. Dylevský:* Morphometrical analysis of the metacarpal bones. Folia Morphol. 29 (1981), Nr. 1, 1–5.

[3] PDP-11 FORTRAN. Language Reference Manual, Digital Equipment Corporation, Maynard, Mass., 1979.

Tab. 2. Results of the check example.

k	m	$Nm = 3$ $XX[1, k, m]$	$YY[1, k, m]$	$XX[2, k, m]$	$YY[2, k, m]$	$a1[k, m]$	$a0[k, m]$	$SSQ[k]$
1	1	0.1	6.040	2.0	13.821	4.095	5.630	583.813
2	1	0.1	1.112	0.6	9.164	16.104	−0.499	136.506
	2	0.9	16.5	2.0	11.0	−5.0	21.0	
3*)	1	0.1	1.5	0.3	4.1	13.0	0.2	129.04
	2	0.5	6.7	0.6	10.6	39.0	−12.8	
	3	0.9	16.5	2.0	11.0	−5.0	21.0	
	1	0.1	1.5	0.5	6.7	13.0	0.2	129.04
	2	0.6	10.6	0.6	10.6	0.0	10.6	
	3	0.9	16.5	2.0	11.0	−5.0	21.0	
	1	0.1	1.5	0.5	6.7	13.0	0.2	129.04
	2	0.6	10.6	0.9	17.5	23.0	−3.2	
	3	1.1	15.5	2.0	11.0	−5.0	21.0	

*) For $k = 3$, three equivalent possibilities are given.