EDV in Medizin und Biologie **18** (4), 105–107, ISSN 0300–8282

A PASCAL program for fitting nonlinear regression models on a micro-computer

Johan D. Buys and Klaus von Gadow

Summary

A numerical algorithm which calculates a least squares fit of data to general nonlinear models is implemented as a PASCAL procedure.

1. Introduction

The use of nonlinear regression models is very popular in forestry research and management. Standard statistical packages are available for fitting data to these models, for example, the routine BMDPAR or the procedure NLIN in SAS. However, as the use of micro-computers is increasing, the need for small, efficient fitting routines arises.

There are commercial programs for fitting a nonlinear model on a small computer. However, these programs are inaccessible to the user and cannot be modified to improve the processing performance under specific circumstances.

For this reason we present a numerical algorithm and a corresponding procedure in PASCAL.

2. The algorithm

We have chosen to implement the well-known algorithm of Levenberg and Marquardt (see I. E. DENNIS and R. B. SCHNABEL, 1983) for minimizing the sum of squares of deviations

$$S(a_1, \ldots, a_m) = \sum_{i=1}^{n} [y_i - b(x_i; a_1, \ldots, a_m)]^2$$

where the regression model is

$$y = f(x; a_1, \ldots, a_m) + \varepsilon,$$

the model parameters are $a_1, \ldots, a_m$ and the data points are (x_i, y_i), $i = 1, \ldots, n$.

The algorithm is iterative, and starts with initial estimates $a_i^{(0)}$, $i = 1, \ldots, m$ of the desired parameter values. After k steps of the algorithm, estimates $a_i^{(k)}$, $i = 1, \ldots, m$ have been obtained and improved estimates $a_i^{(k+1)}$ are then computed as follows:

$$a_i^{(k+1)} = a_i^{(k)} + \Delta a_i^{(k)}, \; i = 1, \ldots, m$$

where the corrections $\Delta a_i^{(k)}$ are obtained as the solution of the set of linear equations

$$[J_k^T J_k + \lambda_k D_k] \, \Delta a^{(k)} = J_k^T R_k.$$

In the linear system, $\Delta a^{(k)}$ is the vector of corrections $\Delta a_i^{(k)}$, R_k is the residual vector

$$R_k = [y_i - f(x_i; a_1^{(k)}, \ldots, a_m^{(k)}]^T,$$

J_k is the $n \times m$ Jacobian matrix

$$J_k = \left[\frac{\partial f}{\partial a_j} (x_i, a_1^{(k)}, \ldots, a_m^{(k)}) \right],$$

D_k is a diagonal weight matrix which we take as $D_k = \mathrm{diag}(J_k^T J_k)$, and λ_k is a parameter which is used to enhance convergence of the algorithm (see below).

In our implementation of the algorithm, we compute all derivatives (elements of the matrix J_k) numerically with a finite difference formula, but this can easily be changed if analytical derivatives are available. In order to conserve storage space, the matrix J_k is not stored. $J_k^T J_k$ is formed step by step as the necessary elements are computed. The system of linear equations is solved by the Cholesky method for positive definite systems.

The strategy for choosing the parameter λ_k is as follows. The residual sum of squares $S_k = S(a_1^{(k)}, \ldots, a_m^{(k)})$ is monitored at each step of the algorithm. If $S_{k+1} < S_k$, it is assumed that the algorithm is converging and λ_k is decreased by a fixed factor in order to force the algorithm towards a Gauss-Newton step for fast convergence. If however $S_{k+1} \geq S_k$, the corrections $\Delta a_i^{(k)}$ are not accepted; instead λ_k is increased to force the algorithm towards a steepest gradient step in an effort to enforce convergence, and the corrections $\Delta a_i^{(k)}$ are recomputed.

The algorithm is stopped when the corrections $\Delta a_i^{(k)}$ are small compared to the parameter estimates $a_i^{(k+1)}$.

```
PROGRAM Nonlin;
TYPE
   dimdata = ARRAY[1..33] OF REAL; {max of 33 data pairs}
   dimpar = ARRAY[1..10] OF REAL;  {max of 10 parameters}
   dimpar2 = ARRAY[1..10,1..10] OF REAL;
VAR
   i : INTEGER;
   fail : BOOLEAN;
   fk : CHAR;
   x, y : dimdata;
   a : dimpar;

PROCEDURE constants;
begin
  {tree heights(Y) and ages(X)}
  x[1]:=5.0; x[2]:=10.0; x[3]:=20.0; x[4]:=30.0; x[5]:=40.0;
  y[1]:=5.0; y[2]:=10.6; y[3]:=18.2; y[4]:=24.5; y[5]:=27.9;
end;

{specify the equation}
FUNCTION func(x : REAL; a : dimpar) : REAL;
BEGIN
   func:=a[1]*(1-exp(a[2]*X));    {example}
END; {func}

PROCEDURE nlsq(n,                 {Input: Number of parameters}
               m : INTEGER;  {Input: Number of observations}
               VAR a : dimpar;  {Input and output: vector of
                                 parameter values}
               x,{Input: Vector of independent observations}
               y : dimdata; {Input: Vector of dependent
                                                observations}
               VAR fail : BOOLEAN  {Output: Error indicator}
               );
VAR
   i, j, k, it, rep : INTEGER;
   ai, er, hi, mu, sq, sqq, ss, u : REAL;
   f : dimdata;
   d, dd, s :  dimpar;
   h :  dimpar2;
   sqdec : BOOLEAN;
CONST
   repmax : INTEGER = 10; itmax : INTEGER = 50;
   inc : REAL = 3.; dec : REAL = 5.;
   eps : REAL = 1.e-6; hh : REAL = 1.e-6;
LABEL
   stop;



PROCEDURE cholsolve;
VAR
   i, j, k : INTEGER;
   s, ss : REAL;
BEGIN
   fail:=False;
   FOR j:=1 TO n DO
   BEGIN
      s:=h[j,j];
      FOR i:=1 TO j-1 DO s:=s-h[j,i]*h[j,i];
      IF s>=0. THEN
      BEGIN
         s:=Sqrt(s); h[j,j]:=s;
         FOR i:=j+1 TO n DO
         BEGIN
            ss:=h[j,i];
            FOR k:=1 TO j-1 DO ss:=ss-h[i,k]*h[j,k];
            h[i,j]:=ss/s;
         END
      END
      ELSE fail:=True;
   END;
   IF NOT fail THEN
   BEGIN
      d[1]:=d[1]/h[1,1];
      FOR i:=2 TO n DO
      BEGIN
         s:=d[i];
         FOR j:=1 TO i-1 DO s:=s-h[i,j]*d[j];
         d[i]:=s/h[i,i];
      END;
      d[n]:=d[n]/h[n,n];
      FOR i:=n-1 DOWNTO 1 DO
      BEGIN
         s:=d[i];
         FOR j:=i+1 TO n DO s:=s-h[j,i]*d[j];
         d[i]:=s/h[i,i];
      END
   END
END; {cholsolve}

BEGIN
   mu:=20.; it:=0; sq:=0.;
   FOR k:=1 TO m DO
   BEGIN
      f[k]:=func(x[k],a); er:=y[k]-f[k]; sq:=sq+er*er;
   END;

REPEAT
   it:=it+1;
   FOR i:=1 TO n DO
   BEGIN
      s[i]:=hh*a[i]; IF s[i] < 1.e-10 THEN s[i]:=hh;
      dd[i]:=0.; FOR j:=i TO n DO h[i,j]:=0.
   END;
   FOR k:=1 TO m DO
   BEGIN
      FOR i:=1 TO n DO
      BEGIN
         ai:=a[i]; a[i]:=ai+s[i]; hi:=a[i]-ai;
         d[i]:=(func(x[k],a) - f[k])/hi; a[i]:=ai;
      END;
      FOR i:=1 TO n DO
      BEGIN
         dd[i]:=dd[i]+d[i]*(y[k]-f[k]);
         FOR j:=i TO n DO h[i,j]:=h[i,j]+d[i]*d[j];
      END
   END;
   FOR i:=1 TO n DO s[i]:=h[i,i];
   rep:=0; sqdec:=false;
   REPEAT
      u:=mu+1.;
      FOR i:=1 TO n DO
      BEGIN
         h[i,i]:=u*s[i]; d[i]:=dd[i];
      END;
      cholsolve;
      IF fail THEN GOTO stop;
      ss:=0.;
      FOR i:=1 TO n DO
      BEGIN
         ss:=ss+d[i]*dd[i]; d[i]:=a[i]+d[i];
      END;
      sqq:=0.;
      FOR k:=1 TO m DO



      BEGIN
         f[k]:=func(x[k],d); er:=y[k]-f[k];
               sqq:=sqq+er*er;
      END;
      IF sqq > sq+1.e-4*ss THEN
      BEGIN
         rep:=rep+1; IF rep = repmax THEN fail:=True;
         mu:=mu*inc; IF rep > 2 THEN mu:=mu*inc;
         Writeln(it:3,rep:3,' ',mu:10,' ',sq:15,'
                 ',sqq:15,' ');(ss:6:3,' ',)
      END
      ELSE  sqdec:=True;
   UNTIL sqdec OR fail;
   IF fail THEN GOTO stop;
   sq:=sqq; ss:=0.;
      FOR i:=1 TO n DO
      BEGIN
         ai:=Abs(d[i]-a[i])/(1.e-12+Abs(d[i]));
         IF ai > ss THEN ss:=ai;
         a[i]:=d[i];
      END;
      IF rep = 0 THEN mu:=mu/dec;
      IF it = itmax THEN fail:=True;
      Writeln; Write(it:3,' ',mu:10,' ',sq:10,' ');
      FOR i:=1 TO n DO Write(a[i]:9:5,' ');
      Writeln;
   UNTIL (ss < eps) OR fail;
   stop:
END; {nlsq}

BEGIN
   constants;
   a[1]:=35; a[2]:=-0.05;  (initial parameter values)
   nlsq(2,5,a,x,y,fail);
   Writeln; Write('fail= ',fail,'  ');
   FOR i:=1 TO 2 DO Write(a[i]:10:7,'  ');
END.
```

3. Numerical examples

As a first example, we present the under bark radii (r_i, in centimetres) at different heights (h_i, in metres) of a Eucalyptus cloeziana tree:

h_i: 0 0,6 1,2 1,35 2,4 4,9 7,3 9,8 12,2 15,2 18,3
r_i: 7,37 6,73 6,10 6,06 5,84 5,08 4,57 3,81 3,05 1,52 0

The following taper model (C. F. BRINK and K. VON GADOW, 1986) was fitted to the data:

$$r(h) = i + (r_b - i)e^{p(b-h)} - \frac{pi}{p+q}\left[e^{q(h-H)} - e^{q(b-H)+p(b-h)}\right] \quad (1)$$

where

r(h) = under bark tree radius (cm),
h = tree height (m),
b = breast height (1,35 m),
r_b = under bark radius at breast height (6,06 cm),
H = total tree height (18,3 m),
e = 2,71 ...
i, p, q = parameters to be estimated.

The following parameter values were obtained after 28 iterations:

i = 10,02; p = 2,1826; q = 0,0524.

The error mean square was 0,028. These values were very close to those obtained with the commercial routine BMDPAR:

i = 10,09; p = 2,2126; q = 0,0520.

The second example concerns the mean height in a stand of Pinus patula trees (h_i, metres) at different ages (t_i, years):

t_i: 1,75 4,24 6,62 8,75 12,75 17,67 23,67 29,45 31,04 35,38
h_i: 1,3 5,4 9,3 12,4 17,0 20,9 24,0 25,3 25,7 26,0 26,7

The 3-parameter Chapman Richards model used by O. GARCIA (1984) was fitted to the data:

$$h = A(1 - e^{kt})^{1/m} \quad (2)$$

where

h = mean height (m),
t = stand age (years),
e = 2,71, ...
A, k, m = parameters to be estimated.

Given the initial parameter values: A = 40; k = −0,2; m = 0,5; the following final values were obtained after 10 iterations:

A = 27,7; k = −0,1042; m = 0,6305.

The residual SS was 0,1736.

The mean height/age data set was used again to fit the 4-parameter Chapman Richards model:

$$h = A(1 - be^{kt})^{1/(1-m)} \quad (3)$$

using the initial values A = 40; b = 1; k = −0,2; m = 0,5.

The final parameter values were obtained after 22 iterations:

A = 28,1; b = 1,07; k = −0,09479; m = 0,2341;

and the residual SS was 0,00919 (note the improved fit when compared with the 3-parameter model).

4. References

BRINK, C. F. and K. VON GADOW, 1986: On the use of growth and decay functions for modelling stem profiles. EDV in Medizin und Biologie **17(1/2):** 20–27.

DENNIS, J. E.; and R. B. SCHNABEL, 1983: Numerical methods for unconstrained optimization and nonlinear equations. Prentice-Hall, New Jersey.

GARCIA, O., 1984: New Class of Growth Models for Even-Aged Stands: Pinus radiata in Golden Downs Forest. N. Z. Journal of Forestry Science **14(1).**

Eingegangen am 22. Juli 1987
The authors' addresses: J. D. Buys, Department of Mathematics, K. von Gadow, Department of Forest Science, University of Stellenbosch, 7600 Stellenbosch, South Africa