УДК 519.254:519.61:372.851

NONLINEAR AND QUASILINEAR REGRESSION WITH CLASSPAD-SOFTWARE, USING THE LEVENBERG-MARQUARDT ALGORITHM

Paditz L.

The ClassPad-software by CASIO was introduced in 2003 for a new generation of handheld calculators and simultaneously for the Windows PC (emulator version of the handheld calculator). Meantime several improvements of the software have been appeared. Now we have the ClassPad 400 with the software version 02.01.1000, cp. http://edu.casio.ru/fx-cp400/ or https://edu.casio.com. This nice software is an educational tool in German high schools or universities to support the learning process in mathematics and other fields.
In 2013 in Moscow [ref. 6] and 2016 in Yaroslavl [ref. 7] the author has given some talks and lectures on using the ClassPad-software in German schools with the goal of introducing this modern tool in teacher education in some Russian pedagogical universities and later in Russian schools to support the learning process of the students with the newest software.
The aim of this paper consists in the demonstration of the ClassPad-software by the help of introducing the Levenberg-Marquardt algorithm to solve new nonlinear regression problems. At first it is shown how to write a program in the program language of the ClassPad and further on some practical problems will be considered.

1. Nonlinear and Quasilinear Regression

In the ClassPad-software several well-known regression models were implemented, e.g. the logistic regression $y = f(x) = \dfrac{c}{1 + a \cdot e^{-b \cdot x}}$ or the power regression $y = f(x) = a \cdot x^b$. The aim is the improvement of these regression models by the help of the Levenberg-Marquardt algorithm.

1.1. Power Regression

The power regression is realized in the CP400 by the help of the quasilinear regression $\ln(y) = \ln(a) + b \cdot \ln(x)$ with $F_1(a,b) = \sum_{i=1}^{n} (\ln(y_i) - \ln(a) - b \cdot \ln(x_i))^2 \rightarrow \min$ for a given data set $(x_i, y_i)_{i=1(1)n}$.

We know that the solution (a_o, b_o) of the quasilinear method $\min\limits_{a,b}(F_1(a,b)) = F_1(a_o, b_o)$ is not the optimal solution for $F_2(a,b) = \sum\limits_{i=1}^{n}(y_i - a \cdot x_i^b)^2 \to \min$ [ref. 8]. We get the optimal solution for $\min\limits_{a,b}(F_2(a,b))$ with the CP-software by the help of programming the Levenberg-Marquardt-algorithm [ref. 1,3-5,9-13].

1.2. General Logistic Regression

We introduce the more general logistic regression model $y = f(x,a,b,c,d) = \dfrac{c}{1 + a \cdot e^{-b \cdot x}} + d$ for the CP-software again by programming the Levenberg-Marquardt-algorithm.

1.3. Arctan-Regression

We introduce a new nonlinear regression model, the arctan-regression $y = f(x,a,b,c,d) = a \cdot \arctan(c \cdot (x - b)) + d$, for the CP-software again by programming the Levenberg-Marquardt-algorithm.

2. Levenberg-Marquardt Algorithm

Consider the following task: $\sum\limits_{i=1}^{n}(F(x_i, y_i, a, b))^2 \to \min$ or $\sum\limits_{i=1}^{n}(F(x_i, y_i, a, b, c, d))^2 \to \min$, where $F(x,y,a,b) = y - a \cdot x^b$ or $F(x,y,a,b,c,d) = y - \dfrac{c}{1 + a \cdot e^{-b \cdot x}} - d = y - \dfrac{c}{1 + e^{A - b \cdot x}} - d$ and $F(x,y,a,b,c,d) = y - a \cdot \arctan(c \cdot (x - b)) - d$ respectively.
Here we work in the program for the logistic regression with $A = \ln(a)$ to avoid numerical problems.

2.1. Levenberg-Marquardt Algorithm

Levenberg [ref. 3] and later Marquardt [ref. 5] suggested to use a dumped Gauss-Newton method. Let $\underline{z}$ be the parameter vector $[a, b]$ and $[a, b, c, d]$ or $[A, b, c, d]$ respectively. To get the minimum of the nonlinear function $\sum\limits_{i=1}^{n}(F(x, y, \underline{z}))^2$ we consider the following algorithm

by Levenberg and Marquardt:
Choose an initial vector $\underline{z}^{(0)}$ for the parameters $a, b, ...$ and an additional initial parameter μ_0 for the control of the iteration process $\underline{z}^{(0)}, \underline{z}^{(1)}, \underline{z}^{(2)}, ..., \underline{z}^{(k)}, ...$.
Now compute the vector $\underline{F} = \underline{F}(\underline{z}^{(k)}) = F(\text{listx,listy},\underline{z}^{(k)})$, where listx $= \{x_1, x_2, ..., x_n\}$ and listy $= \{y_1, y_2, ..., y_n\}$ respectively are the connected pairs of the given data set $(x_i, y_i)_{i=1(1)n}$.

Furthermore compute the matrix $\underline{F}' = \underline{F}'(\underline{z}^{(k)}) = \left(\dfrac{\partial \underline{F}}{\partial \underline{z}^{(k)}}\right) =$

$$\begin{pmatrix} \dfrac{\partial F(x_1, y_1, a, b, c, d)}{\partial a} & \cdots & \dfrac{\partial F(x_1, y_1, a, b, c, d)}{\partial d} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial F(x_n, y_n, a, b, c, d)}{\partial a} & \cdots & \dfrac{\partial F(x_n, y_n, a, b, c, d)}{\partial d} \end{pmatrix} \text{ e.g.}$$

Now we solve the linear problem for the correction $\underline{s}^{(k)}$ of a given $\underline{z}^{(k)}$:

$$\left|\left| \begin{pmatrix} \underline{F}'(\underline{z}^{(k)}) \\ \mu \underline{I} \end{pmatrix} \cdot \underline{s}^{(k)} + \begin{pmatrix} \underline{F}(\underline{z}^{(k)}) \\ \underline{O} \end{pmatrix} \right|\right|_2 = \min!$$

$\underline{I}$ is the unit matrix and $\underline{O}$ is the zero matrix.

This problem equals to $\left|\left|\underline{F}'(\underline{z}^{(k)}) \cdot \underline{s}^{(k)} + \underline{F}(\underline{z}^{(k)})\right|\right|_2^2 + \mu^2 \cdot \left|\left|\underline{s}^{(k)}\right|\right|_2^2 = \min!$ where $\mu > 0$ is fixed [ref. 1].

The formal solution is

$$\underline{s}^{(k)} = -((\underline{F}'(\underline{z}^{(k)}))^T \cdot \underline{F}'(\underline{z}^{(k)}) + \mu^2 \cdot I)^{-1} \cdot (\underline{F}'(\underline{z}^{(k)}))^T \cdot \underline{F}(\underline{z}^{(k)})$$

Now we need an heuristic criterion to control μ: We have the M-L-correction $\underline{s}^{(k)} = \underline{s}^{(k)}(\mu)$ for a given $\underline{z}^{(k)}$ and define the control parameter

$$\mu\varrho_o = \frac{\left|\left|\underline{F}(\underline{z}^{(k)})\right|\right|_2^2 - \left|\left|\underline{F}(\underline{z}^{(k)} + \underline{s}^{(k)})\right|\right|_2^2}{\left|\left|\underline{F}(\underline{z}^{(k)})\right|\right|_2^2 - \left|\left|\underline{F}(\underline{z}^{(k)}) + \underline{F}'(\underline{z}^{(k)}) \cdot \underline{s}^{(k)}\right|\right|_2^2}.$$

Now choose two bounds $\beta_o = 0.2$ and $\beta_1 = 0.8$ e.g. and have following decision, [ref. 11]:

If $\mu\varrho_o < \beta_0$: do not accept $\underline{s}^{(k)}$ and choose a greater μ, say double the value of μ and calculate $\underline{s}^{(k)}$ again.
If $\beta_0 \le \mu\varrho_o \le \beta_1$: do accept $\underline{s}^{(k)}$ and do not change μ and compute the next $\underline{s}^{(k+1)}$ by the help of $\underline{z}^{(k+1)} = \underline{z}^{(k)} + \underline{s}^{(k)}$.
If $\mu\varrho_o > \beta_1$: do accept $\underline{s}^{(k)}$ and make μ smaller, say halving the value and compute the next $\underline{s}^{(k+1)}$ by the help of $\underline{z}^{(k+1)} = \underline{z}^{(k)} + \underline{s}^{(k)}$.

2.2. ClassPad Program PowRegLM

The program has the syntax **PowRegLM**(listx,listy,a0,b0,μ0,I), where here I is the wished number of iterations and a0,b0,μ0 are the initial values. The main steps in the program are [ref. 9,10]:

Define $F(x,y,a,b) = y - a \cdot x^b$
Define $r_1(x,a,b) = \text{diff}(F(x,y,a,b),a)$
Define $r_2(x,a,b) = \text{diff}(F(x,y,a,b),b)$

Now compute the correction $\underline{s}^{(k)}$ by the help of a symmetric matrix SP with the elements SP11, SP12, SP21=SP12, SP22:
approx(sum((approx(r_1(listx, a, b)))^2) + μ^2) ⇒SP11
approx(sum((approx(r_1(listx, a, b))·approx(r_2(listx, a, b)))) ⇒SP12
approx(sum((approx(r_2(listx, a, b)))^2) + μ^2) ⇒SP22
approx([[SP11,SP12],[SP12,SP22]])⇒SP

We get the correction $\underline{s}^{(k)}$(=vecs) in the following manner:
approx(F(listx,listy,a,b))⇒FA
approx($-SP^{-1}$·[[sum(approx(r_1(listx,a, b)) $\cdot FA$)],
[sum(approx(r_2(listx,a, b)) $\cdot FA$)]])⇒vecs

Now we define the $\mu-$control $\mu\varrho_o$:
vecs[1,1]⇒s1
vecs[2,1]⇒s2
The nominator of the $\mu-$control $\mu\varrho_o$ should be the value ε:
approx((sum(FA^2)-sum((FA+s1· approx(r_1(listx,a, b))+
s2· approx(r_2(listx, a, b)))^2))) ⇒ ε

If abs(ε)<10^{-12} we stop the computation and finish the program.

Now we compute the $\mu-$control value $\mu\varrho_o$ and find a decision on using $\underline{z}^{(k)} + \underline{s}^{(k)}$(=vecab) and μ:
approx((sum(FA^2)-sum(approx(F(listx,listy,a+s1,b+s2))^2))/ε)⇒ $\mu\varrho_o$

If $\mu\varrho_o$ ≥0.2
Then
approx([[a],[b]]+vecs)⇒vecab
vecab[1,1]⇒a
vecab[2,1]⇒b
Ifend

If $\mu\varrho_o$ >0.8
Then
$\mu/2 \Rightarrow \mu$

ElseIf $\mu\varrho_o$ <0.2
Then
2·$\mu \Rightarrow \mu$
Ifend

Now we compute the next $\underline{s}^{(k+1)}$ or we finish the program if we have done the wished number I of steps of the iteration. The full text of the program see [ref. 9].
The goodness of fitting of the regression to the given data we check with the **Mean Square error MSerr**:
dim(listx)⇒D
approx(sum(FA^2)/(D-2))⇒MSerr

2.3. ClassPad Programs LogiDReg and ArctDReg

The program LogiDReg has the syntax **LogiDReg**(listx,listy,A0,b0,c0, d0,μ0,I) and ArctDReg has the syntax **ArctDReg**(listx,listy,a0,b0,c0, d0,μ0,I) respectively, where here again I is the wished number of iterations and a0(or A0),b0,c0,d0,μ0 are the initial values. The steps in the programs are similar to the program PowRegLM discussed in subsection 2.2. The full text of the program see [ref. 9].

3. Examples and Numerical Results

The ClassPad software allows to work in so-called eActivity-worksheets. Here both is possible in one worksheet: to write text informations in the text mode and to compute in the calculation mode. The programs must be stored in the so-called library-folder and thus you have acess to these programs. Because the software is a CAS-software all mathematical formulas will be changed in an exact form and the storage resources are overloaded. Therefore we work in the decimal mode and use the approx-command in the program files [ref. 10].

3.1. PowRegLM

We discuss the power regression with the following connected data set listx:={1,2,3,4,5,6,7,8,9,10,11} and listy:={0.471, 0.515, 0.648, 0.881, 1.063, 1.431, 1.563, 1.664, 1.050, 2.344, 2.684}, cp [ref. 2]. listx contains the years 2005 upto 2015, i.e. 11 years, and listy contains the financial expenses of Germany for the international climate protection in billion euros.

With the CP-software (power regression, quasilinear) we get with the **PowerReg** command (syntax: PowerReg listx,listy) followig results:

PowerReg listx,listy
done
DispStat
done
quasilinear Regression:
==========================
Power Reg
$y = a \cdot x^b$
a(=aCoef) = 0.3446765894
b(=bCoef) = 0.7782907682
r(=rCoef) = 0.9639981273
r^2(=r^2Coef) = 0.9292923894
MSe(=MSe) = 0.02843125978
==========================

In the brackets you see the names of the system variables: aCoef, bCoef, rCoef, r^2Coef, MSe.
Thus the quasilinear regression gives the concave function

$y = f(x) = 0.3446765894 \cdot x^{0.7782907682} \approx 0.3447 \cdot x^{0.7783}$ however with the L-M-algorithm we get the optimal solution (a convex function)

$y = f(x) = 0.217252558351917 \cdot x^{1.02399959902067} \approx 0.2173 \cdot x^{1.024}$.
To get the last result we called the program
PowRegLM(listx,listy,a0,b0,μ0,I) with following initial parameters:
a0 = aCoef =0.3446765894, b0 = bCoef =0.7782907682, μ0=1, I=15.
Already after 8 steps the program stops because ε =3.15E-13.
Here we get MSerr= 0.01613769519 in comparision with MSe = 0.02843125978 of the quasilinear regression [ref. 10].

3.2. LogiDReg and ArctDReg

The general logistic regression and arctan-regression respectively we discuss with a real data set $(x_i, y_i)_{i=1(1)50}$, which I got from a teacher of the gymnasium Coswig near to my home town Dresden. The listx:=$\{1, 2, ..., 50\}$ contains time points (with step 1) and the listy:={3.25,3.35,3.54,3.65,3.74,3.82,3.87,3.94,4,4.06,4.11,4.22,4.22,4.27, 4.32,4.34,4.39,4.44,4.46,4.52,4.56,4.62,4.7,4.73,4.77,4.82,4.89,4.93,5,5.09, 5.19,5.31,5.47,5.65,6.08,8.33,9.22,9.44,9.61,9.74,9.8,9.88,9.92,9.96,10.01, 10.05,10.06,10.08,10.1,10.13} contains chemical data generated during an experiment in the class room (chemical reaction, pH values). It seems that the general logistic regression is a good statistical model for the given data. Finally the arctan-regression was somewhat better with a smaller mean square error.

We called **LogiDReg**(listx,listy,A0,b0,c0,d0,μ0,I) with the initial parameters (from a 2D-graphic of the regression function with the data plot) A0 = 20.5, b0 = 0.58, c0 = 5.3, d0 = 4.3, μ0=1, I=30.

After 20 steps the calculation stops because $\varepsilon = 0$ (floating-point arithmetic of the ClassPad software with 15 significand digits). We get the regression function

$y = f(x) = \dfrac{c}{1 + a \cdot e^{-b \cdot x}} + d$ with $a = e^A = e^{20.4915612033146} =$ 793180367.5, b=0.582748352840408, c=5.82567960085492, d=4.26608128744704 and MSerr=0.1766749421

Now we call **ArctDReg**(listx,listy,a0,b0,c0,d0,μ0,I) with the initial parameters (from a 2D-graphic of the regression function with the data plot) a0 = 2, b0 = 35.5, c0 = 0.5, d0 = 7, μ0=8, I=25.

After 18 steps the calculation stops because $\varepsilon = 0$ (floating-point arithmetic of the ClassPad software with 15 significand digits). We get the regression function

$y = f(x) = a \cdot \arctan(c \cdot (x - b)) + d$ with a=2.07300040263717, b=35.3512340196774, c=0.496500939636883, d=7.2667648481508 and a somewhat better MSerr = 0.1127910421 [ref. 10].

4. Discussion

The CP-software allows own programming [ref. 9,10], thus it was possible to introduce the L-M-algorithm in the ClassPad. We hope that the CASIO company will implement the L-M-algorithm in the operation system of the calculator with a next update of the software. Thus our students have a nice possibility to make interesting numerical experiments. This will support the learning process.

To open and to check the vcp-file [ref. 10] you have to download the free trial (90 days) of the CP-manager software "ClassPad Manager Subscription for ClassPad II Series v2.01": https://edu.casio.com/products/classroom/cp2m/

References

1. *Dahmen W., Reusken A.:* Numerik fuer Ingenieure und Naturwissenschaftler. Berlin 2008, ISBN 978-3-540-76492-2, Kap. 6.3: Levenberg-Marquardt-Verfahren
2. *Federal Ministry for Economic Cooperation and Development:* Klimawandel - Zeit zu handeln. Berlin 2016, see data p. 15, https://www.bmz.de/de/mediathek/publikationen/reihen/infobroschueren_flyer/infobroschueren/Materialie262_klimaschutz_konkret.pdf
3. *Levenberg K.:* A Method for the Solution of Certain Problems in Least Squares. Quart. Appl. Math. 2, 164-168, 1944.
4. *Madsen K., Nielsen N.B., Tingleff O.:* Methods for nonlinear least squares problems 2004, Lecture Notes. Informatics and Mathematical Modeling, Technical University of Denmark, http://orbit.dtu.dk/ws/files/2721358/imm3215.pdf
5. *Marquardt D.:* An Algorithm for Least-Squares Estimation of Nonlinear Parameters. SIAM J. Appl. Math. 11, 431-441, 1963.
6. *Paditz L.:* Using CAS in Math Education - first experience in Germany with the newest ClassPad technology CP400. – Moscow State Pedagogical University 29.11.2013: talk of the Internat. Conf. on Math Education Today and Tomorrow. http://www.informatik.htw-dresden.de/~paditz/MathEducationMoscow2013.pdf
7. *Paditz L.:* Opportunities and methods of using ClassPad. – Yaroslavl State Pedagogical University 27.-28.10.2016: invited lectures. http://www.informatik.htw-dresden.de/ ~paditz/MathEducationYaroslavl2016.pdf
8. *Paditz L. (co-author):* Lehr- und Uebungsbuch Mathematik, Bd.3. – Leipzig 2001, ISBN 978-3-446-21682-2, Kap.22: Nichtlineare Regression - ein Beispiel
9. *Paditz L.:* Levenberg-Marquardt Programme fuer ClassPad. Dresden 2017. http:// www.informatik.htw-dresden.de/~paditz/LM-Programme.pdf
10. *Paditz L.:* virtual ClassPad file, 2017: http://www.informatik.htw-dresden.de/~paditz/Paditz-LMV-2017-Vladikavkaz.vcp
11. *RWTH Aachen, Institut fuer Geometrie und Praktische Mathematik:* Beispiel: Levenberg-Marquardt-Verfahren, 2010, https://www.igpm.rwth-aachen.de/Numa/NumaMB/SS10/LevMarq.pdf
12. *Portal iskusstvennogo intellekta:* 2014, http://neuronus.com/theory/246-algoritm-levenberga-markvardta.html
13. *Semyonov E., https://habrahabr.ru/users/lightforever2/* Algoritm Levenberga - Markvardta dlya nelineynogo metoda naimen'shikh kvadratov i yego realizatsiya na Python, 2016, https://habrahabr.ru/post/308626/

Paditz L.
Dresden University of Applied Sciences, Germany
E-mail: paditz@htw-dresden.de