Least Squares

320210093
Shahd Samy Mohamed

320210080
Mahmoud Mohamed shaban

320210255
Mohamed said Mahmoud

320210069
Sama Essam eldin tantawy

320210264
Noran Galal Shehata

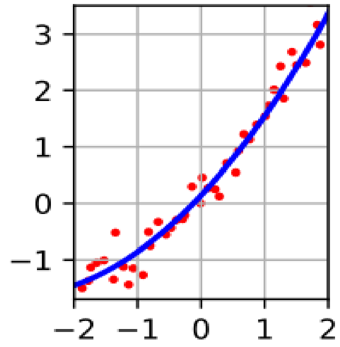
320210302
Alshaimaa Shahat Rabie

320210071
Arwa Fawzy Hussein

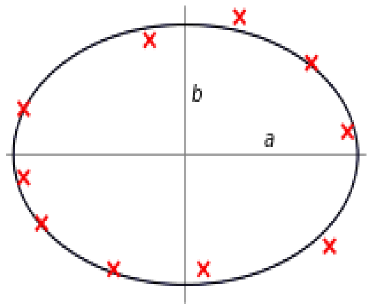
320210081
Mariam Elsherbeny

Supervised by Prof. Waheed Zahra

Least squares method is a standard approach to regression analysis for approximating the solution of overdetermined systems by minimizing the sum of squares of the residuals. Residual is the difference between the observed value and the approximate value provided by the model as the result of the individual equations. The result of fitting a set of data points with a quadratic function is shown in **(Graph 1)**. Conic section of a set of points using least least-square estimation is shown in **(Graph 2)**.



Graph 1



Graph 2

Least Square Method Application

The least-squares approach truly defines the solution for minimizing the sum of squares of deviations or errors in each equation's result. In data fitting, the least-squares method is frequently used. The best fit result is thought to lower the sum of squared errors, or residuals.

Least-squares problems are divided into two categories:

- Ordinary or linear least squares
- Nonlinear least squares

A. Linear least squares

1. numerical solution (y = mx + b) is displayed in **(Table 1)**.
   y = how far up
   x = how far along
   m = Slope or Gradient (how steep the line is)
   b = the Y Intercept (where the line crosses the Y axis)
   n = the number of elements

Table 1

x	1	2	3	4	5	6	7
y	1.5	3.8	6.7	9.0	11.2	13.6	16

Step 1: For each (x, y) point, calculate x^2 and xy **(Table 2)**

Table 2

xy	1.5	7.6	20.1	36	56	81.6	112
x^2	1	4	9	16	25	36	49

Step 2: Sum x, y, x^2 and xy (gives us $\sum x$, $\sum y$, $\sum x^2$ and $\sum xy$ **(Table 3)**):

Table 3

$\sum x$	$\sum y$	$\sum xy$	$\sum x^2$
28	61.8	314.8	140

$$m = \frac{n\ \Sigma(xy) - \Sigma(x)\,\Sigma(y)}{n\ \Sigma(x^2) - \Sigma(x)^2}$$

$$m = \frac{7\ \Sigma(314.8) - \Sigma(28)\,\Sigma(61.8)}{7\ \Sigma(140^2) - \Sigma(28)^2}$$

$$m = \ 2.414$$

$$b = \frac{\Sigma(y)\ -\ m\ \Sigma(x)}{n}$$

$$b = \frac{\Sigma(6.18) - (2.414)\,\Sigma(28)}{7}$$

$$b = \ -0.8274$$

step 5: Assemble the equation of a line:

y = mx + b

y = 2.414 (x) + (-0.8274)

Table 4

x	Y	y = 2.414 (x) + (0.8274)		error
1	1.5	1.5866		0.0866
2	3.8	4.0006		0.2006
3	6.7	6.4146		0.2854
4	9.0	8.8286		0.1714
5	11.2	11.2426		0.0426
6	13.6	13.6566		0.0566
7	16	16.0706		0.0706

2. matrices solution

$A_{n \cdot k}\, \vec{x} = \vec{b},\ \ \vec{x} \in R^k, \vec{b} \in R^n$

This solution is used in a system that has no solution, no linear combination in the coulomb vector of A will be equal to b.., or $\vec{b}$ is not in coulomb space of A. The least-square solution is used to find $\vec{x}$ where Ax is as close to $\vec{b}$ as possible. Minimizing the Error is required $\| \vec{b}$ - A$\vec{x} \|$.

Let A$\vec{x}$= $\vec{v}$

$$| \begin{vmatrix} \vec{b} & \overrightarrow{-v_1} \\ \vec{b} & \overrightarrow{-v_n} \end{vmatrix} | = (b_1 - v_1) + (b_2 - v_2) + \ldots + (b_n - v_n)$$

The absolute values might have discontinuous values, we can redefine the Error function and use the Error squared.

$$| \begin{vmatrix} \vec{b} & \overrightarrow{-v_1} \\ \vec{b} & \overrightarrow{-v_n} \end{vmatrix} |^2 = (b_1 - v_1)^2 + (b_2 - v_2)^2 + \ldots + (b_n - v_n)^2$$

The squared function has the same minimum as the absolute value function, but with the continuous derivative.

x = {12,4,3,7}, so the least is {3}, and x^2= {141,16,9,49}, so the least square is {9}.

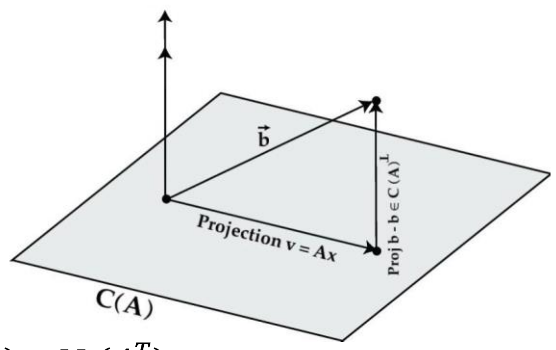
Knowing that the closest vector in any subspace to a vector not in that subspace is the projection of that vector onto that subspace as shown in **(graph 3)**. This graph can be used to derive the equation below.

A$\vec{x}$ = proj $\vec{b}$ c(A). >>> *Subtract* $\vec{b}$

A$\vec{x} -$ $\vec{b}$ = proj $\vec{b}$ c (A) - $\vec{b}$

A$\vec{x} - \vec{b} \in$ c $(A^{\perp})$ (a member of the orthogonal complement of the subspace).

Since the orthogonal complement of the subspace is equal to the NULL space of A transpose c $(A^{\perp})$ = N (A^T).



Graph 3

A$\vec{x} -$ $\vec{b} \in$ N (A^T)

Multiply by A^T

$$A^T \, A\vec{x} - A^T\vec{b} = \vec{0}$$

$$A^T \, A\vec{x} = A^T \, \vec{b}$$

This equation has a solution, the least square solution.

Example: Find a least-squares solution of the inconsistent system Ax = b for

$$A = \begin{bmatrix} 4 & 0 \\ 0 & 2 \\ 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 0 \\ 11 \end{bmatrix}$$

Solution: $A^T A = \begin{bmatrix} 4 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 17 & 1 \\ 1 & 5 \end{bmatrix}$

$$A^T b = \begin{bmatrix} 4 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 11 \end{bmatrix} = \begin{bmatrix} 19 \\ 11 \end{bmatrix}$$

$\because A^T A\hat{x} = A^T b$, $\begin{bmatrix} 17 & 1 \\ 1 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 19 \\ 11 \end{bmatrix}$, $(A^T A)^{-1} = \frac{1}{84}\begin{bmatrix} 5 & -1 \\ -1 & 17 \end{bmatrix}$

$\therefore \hat{x} = (A^T A)^{-1} A^T b = \frac{1}{84}\begin{bmatrix} 5 & -1 \\ -1 & 17 \end{bmatrix}\begin{bmatrix} 19 \\ 11 \end{bmatrix} = \frac{1}{84}\begin{bmatrix} 84 \\ 168 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$A\hat{x} = \begin{bmatrix} 4 & 0 \\ 0 & 2 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 3 \end{bmatrix}$, $\therefore b - A\hat{x} = \begin{bmatrix} 2 \\ 0 \\ 11 \end{bmatrix} - \begin{bmatrix} 4 \\ 4 \\ 3 \end{bmatrix} = \begin{bmatrix} -2 \\ -4 \\ 8 \end{bmatrix}$

$\|b - A\hat{x}\| = \sqrt{(-2)^2 + (-4)^2 + 8^2} = \sqrt{84}$, $\therefore$ The least-squares error is $\sqrt{84}$ For any x in R^2.

□□□□□□□□

B. **The Nonlinear solution:**

<u>The Gauss Newton method:</u>

$y = f(x\,;\,\beta) + \epsilon$,ϵ is the Error or Noise, $x \in R^n : R$.

$f(x)$ is nonlinear, 〘*i and j are variables for looping in procedural programming*〙

The residuals are calculated as below:

$r_i = y_i - f(x_i\,;\,\beta)$

Minimize the residuals is needed as much as possible, so the Loss function is defined; it is the sum of the square residual values over all the observations.

The loss function is calculated as below:

$$L = \sum_i r_i^2$$

Minimizing the loss function is required with regards to β which is the x-factor.

Taking the first gradient is as the following:

$\nabla_{\beta_j} L = \sum_i 2r_i \,.\, \frac{\partial r_i}{\partial \beta_j} = \sum_i 2r_i \,.\, \frac{-\partial f_i}{\partial \beta_j} = -2\sum_i r_i \,.\, \frac{\partial f_i}{\partial \beta_j}$ ∇ is delta

Writing it in a matrix form is equal to:

$$\begin{bmatrix} \frac{df_1}{d\beta_1} & \cdots & \frac{df_1}{d\beta_p} \\ \vdots & & \vdots \\ \frac{df_n}{d\beta_1} & & \frac{df_n}{d\beta_p} \end{bmatrix} . \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix}$$

>> *observation:* the first matrix is the transpose of the Jacobin matrix, therefore,

$\nabla_{\beta_j} = -2J^T \,.\, \vec{r}$

Taking the second gradient which is known as the hessian matrix:

$$\nabla^2_{\beta_j\beta_k} L = -2\sum_i \left(-\frac{\partial f_i}{\partial \beta_k}.\frac{\partial f_i}{\partial \beta_j} + r_i \frac{\partial^2 f_i}{\partial \beta_j\, \partial \beta_k}\right)$$

Jacobian method set $r_i \frac{\partial^2 f_i}{\partial \beta_j\, \partial \beta_k}$ to zero to simplify the equation, so far

$\nabla^2_{\beta_j\beta_k} L \approx 2\sum_i \frac{\partial f_i}{\partial \beta_k}.\frac{\partial f_i}{\partial \beta_j}$, which is equal to the Jacobian matrix transpose times the Jacobian matrix

$$\nabla^2_{\beta_j\beta_k} = 2\, J^T.J$$

The final equation after some steps and procedures is:

$$\beta_{next} = \beta_{previous} - \left(\nabla^2_{\beta} f\right)^{-1}.\nabla_{\beta} f$$

$$\beta_{n+1} = \beta_n + (J_n^T.J_n)^{-1} J_n^T.r_n$$

MATLAB implementation:

In several real-life applications, the resulting system of linear equation $Ax = b$ has no solution. According to the least square linear matrices solution, it is possible to ignore that by finding the closest value of x that makes Ax as close as possible to b. The following illustration demonstrates that the code can be implemented in any valid matrices of a system of linear equation (SLE).

$$A_{n\times k} = \begin{matrix} A_{11} & A_{12}\dots & A_{1k} \\ A_{21} & A_{22}\dots & A_{2K} \\ A_{n1} & A_{n2}\cdots & A_{nk} \end{matrix} \qquad b_{1\times k} = b_1 \quad b_2 \dots \quad b_k$$

$$Ax = b \quad \xrightarrow{multiplying\ by\ A^T} AA^T x = A^T b \quad \xrightarrow{the\ result\ is} \hat{x} = (AA^T)^{-1} A^T b = A^T b$$

Typing the first line identifies the input data, A, and the output one, b, for MATLAB.

```
>> A=[1 1 1; 1 2 4]

A =

     1     1     1
     1     2     4
```

Fig 1

```
b =

     2     3     5

>> |
```

Fig 2

Using the built-in functions in MATLAB, this is the pseudo inverse of the A matrix — *pinv* is a built-in function that executes the same order:

```
>> A'*inv(A*A')

ans =

    1.0000   -0.2857
    0.5000   -0.0714
   -0.5000    0.3571
```

Fig 3

```
>> pinv(A)

ans =

    1.0000   -0.2857
    0.5000   -0.0714
   -0.5000    0.3571
```

Fig 4

One and one are the exact solutions of the data points assuming that $b = A + 1$ is a relation between them. However, changing the output value from 5 to six can change the result as shown below:

```
>> b=[2 3 6]

b =

     2     3     6
```

Fig 5

```
>> x=b*pinv(A)

x =

    0.5000    1.3571
```

Fig 6

At this time, the equation was found out to be $1.35A + 0.5$. The estimated value of output *bhat* highly approached the value of the original output data, b.

```
bhat =

    1.8571    3.2143    5.9286

>> b

b =

     2     3     6
```

Fig 7

The illustrated image highlights the linear function handle which describes the data with the minimum amount of squared errors, and this is the outcome of the least-squares solution method.

```
>> f=@(A) x(1)+ x(2)*A

f =

  function_handle with value:

    @(A)x(1)+x(2)*A
```

Fig 8

By typing edit, the editor file is opened `>> edit` to visualize the fitting graph after writing the following code:

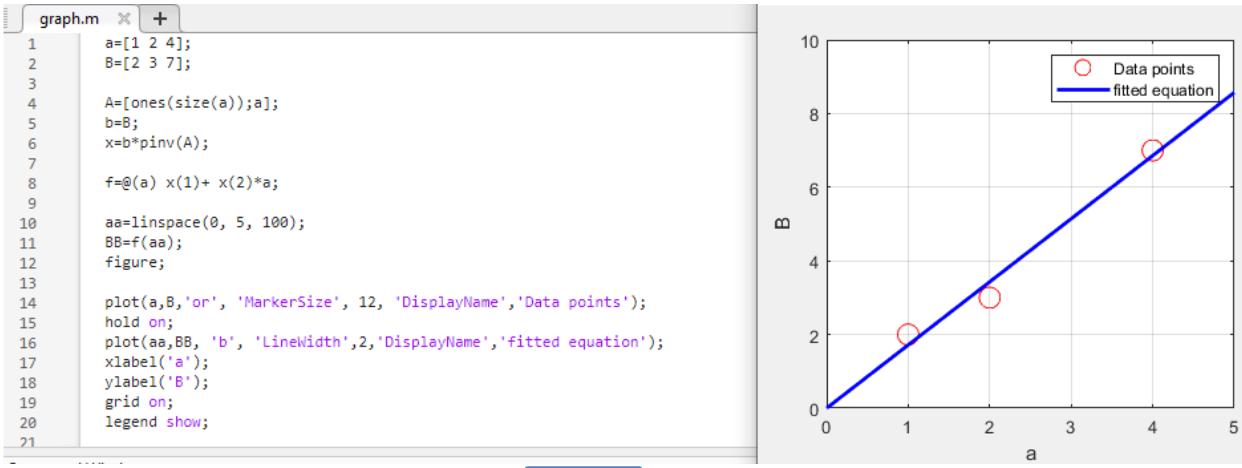


Fig 9

The general form of polynomial repressors can exhibit the non-linear system of equations, quadratic form, by changing only these lines in the linear equation code and saving it in a different path or new file. As shown below, the nonlinear fitting is more accurate to the data points.

The non-linear graph is displayed in the figure below:

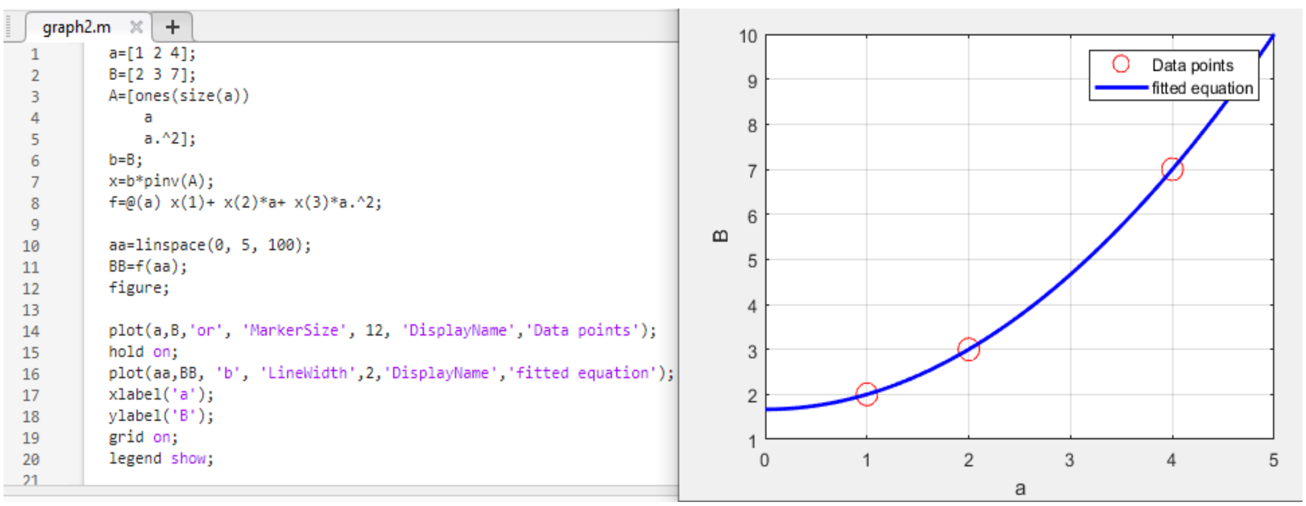


Fig 10

Kindly, check the attached link to run the sample code through MATLAB: https://drive.google.com/drive/u/0/folders/1bCXXQbyhkOp18KrXrPUS6orsboBupySc

References

Admin. (2021, June 24). Least square method - definition, graph and formula. BYJUS. Retrieved June 12, 2022, from https://byjus.com/maths/least-square-method/

Google. (n.d.). Silo.pub_linear-algebra-and-its-applications-4th-edition-G-5520760.PDF. Google Drive. Retrieved June 12, 2022, from https://drive.google.com/file/d/1tdv0CkREK1B7YRc0ovqpJFeMFoDeY84_/view

Iruoghene. I. “Linear Regression Problems with Solutions.” *Academia.edu*, 19 Feb. 2020. https://www.academia.edu/42017737/Linear_Regression_Problems_with_Solutions

Levenberg. K. (1944) “A Method for the Solution of Certain Non-Linear Problems in Least square.” Rhode Island, USA. American Mathematical society.https://www.ams.org/journals/qam/1944-02-02/S0033-569X-1944-10666-0/S0033-569X-1944-10666-0.pdf.

Least squares regression line: Ordinary and partial. Statistics How To. (2022, March 1). Retrieved June 12, 2022, from https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/least-squares-regression-line/

Least squares regression. (n.d.). Retrieved June 12, 2022, from https://www.mathsisfun.com/data/least-squares-regression.html

The least squares regression line. (n.d.). Retrieved June 12, 2022, from https://saylordotorg.github.io/text_introductory-statistics/s14-04-the-least-squares-regression-l.html