
Least Squares Curve Fitting

Tamas Kis | tamas.a.kis@outlook.com | <https://tamaskis.github.io>

Contents

| | | |
|----------|--|-----------|
| 1 | An Algebraic Approach | 2 |
| 1.1 | Vector Representation of a Data Set | 2 |
| 1.2 | The Normal Equation | 2 |
| 1.3 | Linear Least Squares Solution | 3 |
| 2 | Fitting Linear and Polynomial Models to Data | 4 |
| 2.1 | Polynomial Fit | 4 |
| 2.2 | Linear Fit | 5 |
| 3 | Approximating Nonlinear Fits to Data | 7 |
| 3.1 | Power Fit | 7 |
| 3.2 | Exponential Fit | 7 |
| 3.3 | Logarithmic Fit | 8 |
| 4 | Evaluating the Goodness of Fit | 10 |
| 4.1 | Calculating the Coefficient of Determination (r^2) | 10 |
| | References | 10 |

Copyright © 2021 Tamas Kis

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



1 AN ALGEBRAIC APPROACH

1.1 Vector Representation of a Data Set

Consider a set of m data points, where x is the independent variable and y is the dependent variable: $\{(x_i, y_i)\}_{i=1}^m$. In a computational setting, we organize this data set into two column vectors: \mathbf{x} stores the values of the independent variables, while \mathbf{y} stores the values of the dependent variables. The i^{th} element of \mathbf{y} corresponds to the i^{th} element of \mathbf{x} .

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_m \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_m \end{bmatrix}$$

1.2 The Normal Equation

Let $\mathbf{b} \in \mathbb{R}^m$ and let S be a subspace of \mathbb{R}^m . Let's consider the problem where we want to find the vector $\mathbf{p} \in S$ that best approximates \mathbf{b} . For an arbitrary vector $\mathbf{y} \in \mathbb{R}^m$, there is a unique element \mathbf{p} of S that is closest to \mathbf{b} ; that is $\|\mathbf{b} - \mathbf{y}\| > \|\mathbf{b} - \mathbf{p}\|$ for any $\mathbf{y} \neq \mathbf{p}$ in S . Furthermore, a given vector $\mathbf{p} \in S$ will be closest to a given vector $\mathbf{b} \in \mathbb{R}^m$ if and only if $\mathbf{b} - \mathbf{p} \in S^\perp$. This situation is depicted in Fig. 1. Slightly reworded, the vector $\mathbf{p} \in S$ that

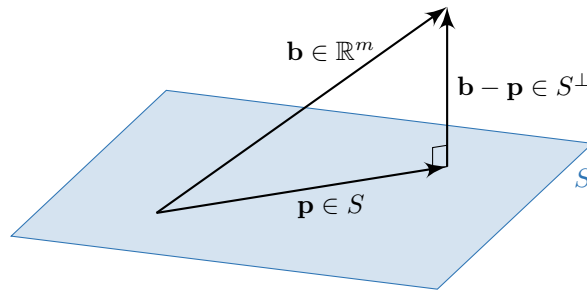


Figure 1: Closest vector $\mathbf{p} \in S$ to a vector $\mathbf{x} \in \mathbb{R}^m$.

best approximates (i.e. is closest to) a vector $\mathbf{b} \in \mathbb{R}^m$ is orthogonal to a vector $\mathbf{b} - \mathbf{p} \in S^\perp$. This is important for the next scenario, where we consider the linear system $\mathbf{Ax} = \mathbf{b}$. If \mathbf{b} is *not* in the column space of A , that is $\mathbf{b} \notin C(\mathbf{A})$, then we know that $\mathbf{Ax} = \mathbf{b}$ is inconsistent. Nonetheless, we still wish to find a vector $\hat{\mathbf{x}}$ that best approximates the solution to $\mathbf{Ax} = \mathbf{b}$. Since $\mathbf{Ax} \in C(\mathbf{A})$ and $\mathbf{b} \notin C(\mathbf{A})$, we know that the \mathbf{Ax} that best approximates \mathbf{b} must satisfy $\mathbf{b} - \mathbf{Ax} \in C(\mathbf{A})^\perp$ (this is essentially the same scenario as before, where \mathbf{Ax} takes the place of \mathbf{p} and $C(\mathbf{A})$ takes the places of S). This situation is depicted in Fig. 2. Since $\mathbf{b} - \mathbf{Ax} \in C(\mathbf{A})^\perp$, we know $\mathbf{b} - \mathbf{Ax} \in N(\mathbf{A}^T)$, where

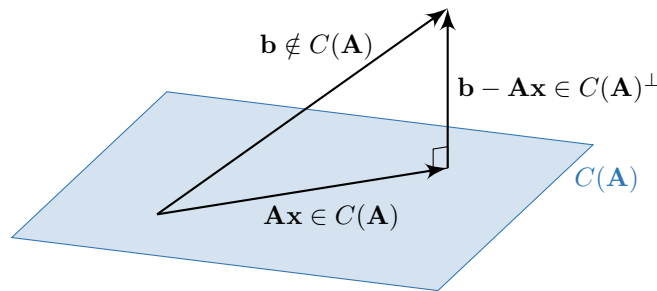


Figure 2: Closest vector $\mathbf{Ax} \in C(\mathbf{A})$ to a vector $\mathbf{b} \notin C(\mathbf{A})$.

$N(\mathbf{A}^T)$ is the null space of \mathbf{A}^T . Recall that $N(\mathbf{A}^T) = \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{A}^T \mathbf{x} = \mathbf{0}\}$. Therefore, if $\mathbf{b} - \mathbf{Ax} \in N(\mathbf{A}^T)$, then

$$\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}) = \mathbf{0}$$

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

$$\boxed{\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}} \quad (1)$$

Equation (1) is known as the **normal equation** [4].

1.3 Linear Least Squares Solution

The linear least squares solution (which we denote as $\hat{\mathbf{x}}$) is the solution of the normal equation for \mathbf{x} . We can note that $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{A} = \mathbf{I}$, where \mathbf{I} is the identity matrix. Therefore, multiplying both sides of Eq. (1) from the left by $(\mathbf{A}^T\mathbf{A})^{-1}$ [4],

$$(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{A}\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

$$\mathbf{I}\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

$$\boxed{\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}} \quad (2)$$

In Section 2, we will discuss two different models that can be fit to data using linear least squares:

1. Linear fit: $y = mx + b$
2. Polynomial fit: $y = \sum_{i=0}^n a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$

Additionally, in Section 3, we will discuss fitting the following three nonlinear models to do data:

1. Power fit: $y = ax^b$
2. Exponential fit: $y = ae^{bx}$
3. Logarithmic fit: $y = a + b \ln x$

The linear and polynomial fits are performed by directly solving a linear least squares problem.

The power, exponential, and logarithmic fits require us to first linearize the data, find a linear fit to the linearized data, and then transform the resulting least squares coefficients to describe the nonlinear data. **Therefore, while these fits often closely approximate the true least squares fit, they are still, in general, not optimal fits to the data in the least squares sense.**

Note that the coefficients of each model are slightly different and use different symbols. However, in the algorithms, we will be returning these coefficients packaged into a single coefficient vector, which we refer to as the **model coefficient vector**. To help clarify the difference between the coefficient vector obtained through solving the least squares problem and the coefficient vector for a specific model fit, we introduce Convention 1.

Convention 1: Coefficient vectors.

- $\hat{\mathbf{a}}$ = least squares coefficient vector (linear least squares solution to $\mathbf{X}\mathbf{a} = \mathbf{y}$)
- \mathbf{c} = model coefficient vector (vector storing coefficients defining the model that has been fit to the data)

2 FITTING LINEAR AND POLYNOMIAL MODELS TO DATA

2.1 Polynomial Fit

A polynomial p_n of degree n is defined as

$$p_n(x) = \sum_{i=0}^n a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

Let's say we have a set of data $\{(x_1, y_1), \dots, (x_m, y_m)\}$. We wish to find a polynomial of degree n that best approximates this data set. For an arbitrary x , we have $p_n(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$. Therefore, from each pair (x_i, y_i) , we can form the system of equations

$$\begin{aligned} a_0 + a_1 x_1 + a_2 x_1^2 + \dots + a_n x_1^n &= y_1 \\ a_0 + a_1 x_2 + a_2 x_2^2 + \dots + a_n x_2^n &= y_2 \\ &\vdots \\ a_0 + a_1 x_m + a_2 x_m^2 + \dots + a_n x_m^n &= y_m \end{aligned}$$

which is linear in a_i . Thus, writing this linear system in matrix form,

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Let

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^n \end{bmatrix} \quad (3)$$

$$\mathbf{a} = (a_0, \dots, a_n)^T \quad (4)$$

$$\mathbf{y} = (y_1, \dots, y_m)^T \quad (5)$$

Then the linear system can be written simply as $\mathbf{X}\mathbf{a} = \mathbf{y}$ where $\mathbf{X} \in \mathbb{R}^{m \times (n+1)}$, $\mathbf{a} \in \mathbb{R}^{n+1}$, and $\mathbf{y} \in \mathbb{R}^m$. In general, the number of data points (m) is far greater than the number of coefficients corresponding to an n^{th} -degree polynomial ($n+1$). Since $m > n+1$, the linear system $\mathbf{X}\mathbf{a} = \mathbf{y}$ is overdetermined. From Section 1, we know that (a) the best approximation of the solution to an overdetermined linear system is the least squares solution and (b) that the least squares solution to an overdetermined linear system can be found by solving the corresponding normal equation for that linear system. To form the normal equation corresponding to $\mathbf{X}\mathbf{a} = \mathbf{y}$, we just need to multiply both sides by \mathbf{A}^T from the left.

$$\mathbf{X}^T \mathbf{X} \mathbf{a} = \mathbf{X}^T \mathbf{y}$$

Solving this linear system for $\hat{\mathbf{a}}$,

$$\hat{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (6)$$

In this case, the model coefficients $\mathbf{c} = (c_0, \dots, c_n)^T$ are the same as the least squares coefficients. Thus [4],

$$\mathbf{c} = \hat{\mathbf{a}} \quad (7)$$

Algorithm 1 below outlines how to obtain a polynomial least squares fit to a set of data.

Algorithm 1:

Fitting a polynomial to a data set.

Given:

- $\mathbf{x} \in \mathbb{R}^m$ - independent variable data
- $\mathbf{y} \in \mathbb{R}^m$ - dependent variable data
- $n \in \mathbb{Z}$ - degree of approximately polynomial

Procedure:

1. Determine m from the length of \mathbf{x} .
2. Preallocate the matrix $\mathbf{X} \in \mathbb{R}^{m \times (n+1)}$.
3. Populate the matrix \mathbf{X} .

```

for  $i = 1$  to  $m$ 
    |
    |   for  $j = 1$  to  $n + 1$ 
    |   |    $X_{i,j} = x_i^{j-1}$ 
    |   |   end
    |   end
end

```

4. Obtain the least squares solution to the normal equations.

$$\hat{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

5. Obtain the model coefficient vector.

$$\mathbf{c} = \hat{\mathbf{a}}$$

Return:

- $\mathbf{c} = (a_0, \dots, a_n)^T$ - defines the polynomial model

$$y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

2.2 Linear Fit

To produce a linear fit, we can simply use Algorithm 1 with $n = 1$ to obtain $\hat{\mathbf{a}} = (a_0, a_1)^T$. We can then set

$$\boxed{m = a_1} \tag{8}$$

$$\boxed{b = a_0} \tag{9}$$

to obtain the linear fit

$$y = mx + b$$

Algorithm 2:

Fitting a linear model to a data set.

Given:

- $\mathbf{x} \in \mathbb{R}^m$ - independent variable data
- $\mathbf{y} \in \mathbb{R}^m$ - dependent variable data

Procedure:

1. Find the least squares coefficient vector $\hat{\mathbf{a}}$ for the linear fit to the data using Algorithm 1 with $n = 1$.
2. Determine the linear model (i.e. slope and y -intercept) coefficients m and b from $\hat{\mathbf{a}} = (a_0, a_1)^T$.

$$m = a_1, \quad b = a_0$$

Return:

- $\mathbf{c} = (m, b)^T$ - defines the linear model $y = mx + b$

3 APPROXIMATING NONLINEAR FITS TO DATA

3.1 Power Fit

The power model is

$$y = ax^b$$

Taking the natural logarithm of both sides,

$$\ln y = \ln ax^b \rightarrow \ln y = \ln a + \ln x^b \rightarrow \ln y = \ln a + b \ln x$$

$$\boxed{y = ax^b \iff \ln y = \ln a + b \ln x} \quad (10)$$

Eq. (10) represents a 1st-order polynomial (i.e. a linear function) and can thus be written as

$$y_\ell = a_0 + a_1 x_\ell$$

where we have performed the change of variables

$$y_\ell = \ln y, \quad x_\ell = \ln x$$

It follows that [1, 3]

$$\ln a = a_0 \rightarrow \boxed{a = e^{a_0}} \quad (11)$$

$$\boxed{b = a_1} \quad (12)$$

Algorithm 3 outlines the procedure for finding the power fit to a data set $\{(x_i, y_i)\}_{i=1}^m$.

Algorithm 3:

Fitting a power model to a data set.

Given:

- $\mathbf{x} \in \mathbb{R}^m$ - independent variable data
- $\mathbf{y} \in \mathbb{R}^m$ - dependent variable data

Procedure:

1. Linearize the data set by calculating the natural logarithm of all the x_i 's and y_i 's.

$$\mathbf{x}_\ell = \ln \mathbf{x}, \quad \mathbf{y}_\ell = \ln \mathbf{y}$$

2. Find the least squares coefficient vector $\hat{\mathbf{a}}$ for the linear fit to the linearized data \mathbf{y}_ℓ vs. \mathbf{x}_ℓ using Algorithm 1 with $n = 1$.
3. Determine the power model coefficients a and b from $\hat{\mathbf{a}} = (a_0, a_1)^T$.

$$a = e^{a_0}, \quad b = a_1$$

Return:

- $\mathbf{c} = (a, b)^T$ - defines the power model $y = ax^b$

3.2 Exponential Fit

The exponential model is

$$y = ae^{bx}$$

Taking the natural logarithm of both sides,

$$\ln y = \ln a e^{bx} \quad \rightarrow \quad \ln y = \ln a + \ln e^{bx} \quad \rightarrow \quad \ln y = \ln a + bx$$

$$\boxed{y = a e^{bx} \iff \ln y = \ln a + bx} \quad (13)$$

Equation (13) represents a 1st-order polynomial (i.e. a linear function) and can thus be written as

$$y = a_0 + a_1 x$$

where we have performed the change of variables

$$x_\ell = x, \quad y_\ell = \ln y$$

It follows that [1, 3]

$$\ln a = a_0 \quad \rightarrow \quad \boxed{a = e^{a_0}} \quad (14)$$

$$\boxed{b = a_1} \quad (15)$$

Algorithm 4 outlines the procedure for finding the exponential fit to a data set $\{(x_i, y_i)\}_{i=1}^m$.

Algorithm 4:

Fitting an exponential model to a data set.

Given:

- $\mathbf{x} \in \mathbb{R}^m$ - independent variable data
- $\mathbf{y} \in \mathbb{R}^m$ - dependent variable data

Procedure:

1. Linearize the data set by calculating the natural logarithm of all the y_i 's.

$$\mathbf{x}_\ell = \mathbf{x}, \quad \mathbf{y}_\ell = \ln \mathbf{y}$$

2. Find the least squares coefficient vector $\hat{\mathbf{a}}$ for the linear fit to the linearized data \mathbf{y}_ℓ vs. \mathbf{x}_ℓ using Algorithm 1 with $n = 1$.
3. Determine the exponential model coefficients a and b from $\hat{\mathbf{a}} = (a_0, a_1)^T$.

$$a = e^{a_0}, \quad b = a_1$$

Return:

- $\mathbf{c} = (a, b)^T$ - defines the exponential model $y = a e^{bx}$

3.3 Logarithmic Fit

The logarithmic model is

$$y = a + b \ln x$$

Unlike the power and exponential models, we do not need to linearize the logarithmic model; it is already linear in $\ln x$. Thus, it can be written as

$$y_\ell = a_0 + a_1 x_\ell$$

where we have performed the change of variables

$$x_\ell = \ln x, \quad y_\ell = y$$

It follows that [1, 3]

$$a = a_0 \quad (16)$$

$$b = a_1 \quad (17)$$

Algorithm 5 outlines the procedure for finding the logarithmic fit to a data set $\{(x_i, y_i)\}_{i=1}^m$.

Algorithm 5:

Fitting a logarithmic model to a data set.

Given:

- $\mathbf{x} \in \mathbb{R}^m$ - independent variable data
- $\mathbf{y} \in \mathbb{R}^m$ - dependent variable data

Procedure:

1. Linearize the data by calculating the natural logarithm of all the x_i 's.

$$\mathbf{x}_\ell = \ln \mathbf{x}, \quad \mathbf{y}_\ell = \mathbf{y}$$

2. Find the least squares coefficient vector $\hat{\mathbf{a}}$ for the linear fit to the linearized data \mathbf{y}_ℓ vs. \mathbf{x}_ℓ using Algorithm 1 with $n = 1$.
3. Determine the logarithmic model coefficients a and b from $\hat{\mathbf{a}} = (a_0, a_1)^T$.

$$a = a_0, \quad b = a_1$$

Return:

- $\mathbf{c} = (a, b)^T$ - defines the logarithmic model $y = a + b \ln x$

4 EVALUATING THE GOODNESS OF FIT

To evaluate the goodness of fit of a certain fit to the data, we can calculate the coefficient of determination, r^2 . If r^2 is close to 1, the fit is good, while if it is close to 0, the fit is poor.

4.1 Calculating the Coefficient of Determination (r^2)

The coefficient of determination, r^2 , can be calculated as

$$r^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \quad (18)$$

where the residual sum of squares (SS_{res}) and the total sum of squares (SS_{tot}) are defined as

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2 \quad (19)$$

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2 \quad (20)$$

In Eq. (19), \bar{y} is the mean of the y -values of the data (i.e. mean of $\{y_i\}_i^m$), while in Eq. (20), f_i are the evaluations of the fit at the x -values of the data (i.e. at $\{x_i\}_i^m$) [2].

REFERENCES

- [1] Henry Chan. *Checking Assumptions and Transforming Data*. MATH 2810 Lecture (Vanderbilt University). (notes taken by Tamas Kis on April 17, 2020).
- [2] *Coefficient of determination*. Wikipedia. https://en.wikipedia.org/wiki/Coefficient_of_determination (accessed: June 7, 2021).
- [3] *Linearizing the Equation*. MacEwan University Physics Laboratories. <http://academic.macewan.ca/physlabs/Linearization.pdf> (accessed: June 14, 2020).
- [4] Gieri Simonett. *5.3 Least Squares Problems*. MATH 2410 Lecture (Vanderbilt University). (notes taken by Tamas Kis on May 27, 2018).