Linear Equation - Different solving approaches

Ankesh Kumar

1 Least-Squares Analysis

Least-squares analysis is a fundamental technique in statistics and machine learning for estimating the parameters of a model by minimizing the sum of the squared residuals. It is particularly useful in regression problems, such as line fitting.

1.1 Definition

Given a set of n data points (x_i, y_i), $i = 1, \ldots, n$, and a model function $f(x, \boldsymbol{\beta})$ with parameters $\boldsymbol{\beta}$, the least-squares method finds the optimal parameters by minimizing the sum of squared residuals:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} [y_i - f(x_i, \boldsymbol{\beta})]^2 \tag{1}$$

1.2 Mathematical Formulation

For linear regression, we can express the problem in matrix form:

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \tag{2}$$

where $\mathbf{y}$ is the vector of observed values, $\mathbf{X}$ is the design matrix, and $\boldsymbol{\beta}$ is the vector of parameters.

1.3 Example: Line Fitting

Consider fitting a straight line $y = mx + b$ to a set of data points. Our model is:

$$f(x, \boldsymbol{\beta}) = \beta_0 + \beta_1 x \tag{3}$$

The least-squares solution is given by:

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{4}$$

where $\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$.

1.3.1 Example Calculation

Given data points: $(1, 2)$, $(2, 3)$, $(3, 5)$.

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

Compute $\mathbf{X}^T\mathbf{X}$:

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix}$$

Compute $(\mathbf{X}^T\mathbf{X})^{-1}$:

$$(\mathbf{X}^T\mathbf{X})^{-1} = \frac{1}{(3)(14) - (6)(6)} \begin{bmatrix} 14 & -6 \\ -6 & 3 \end{bmatrix} = \begin{bmatrix} \frac{14}{6} & -1 \\ -1 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{7}{3} & -1 \\ -1 & \frac{1}{2} \end{bmatrix}$$

Compute $\mathbf{X}^T\mathbf{y}$:

$$\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \end{bmatrix}$$

Compute $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{bmatrix} \frac{7}{3} & -1 \\ -1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 10 \\ 20 \end{bmatrix} = \begin{bmatrix} \frac{70}{3} - 20 \\ -10 + 10 \end{bmatrix} = \begin{bmatrix} \frac{10}{3} \\ 0 \end{bmatrix} = \begin{bmatrix} 3.33 \\ 1 \end{bmatrix}$$

The fitted line is $y = 1 + 1x$.

2 Recursive Least Squares Algorithm

The Recursive Least Squares (RLS) algorithm is an adaptive filter algorithm that recursively finds the coefficients that minimize a weighted linear least squares cost function relating to the input signals. It is especially useful in time-varying systems and online learning scenarios.

2.1 Definition

RLS is an algorithm that recursively updates the least-squares estimate of the parameters as new data becomes available, without the need to recompute using all past data.

2.2 Mathematical Formulation

Let $\boldsymbol{\beta}_k$ be the parameter estimate at time k. The RLS algorithm updates the estimate as follows:

$$\mathbf{K}_k = \mathbf{P}_{k-1}\mathbf{x}_k(\lambda + \mathbf{x}_k^T\mathbf{P}_{k-1}\mathbf{x}_k)^{-1} \tag{5}$$

$$\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} + \mathbf{K}_k(y_k - \mathbf{x}_k^T\boldsymbol{\beta}_{k-1}) \tag{6}$$

$$\mathbf{P}_k = \lambda^{-1}(\mathbf{P}_{k-1} - \mathbf{K}_k\mathbf{x}_k^T\mathbf{P}_{k-1}) \tag{7}$$

where λ is the forgetting factor, $\mathbf{K}_k$ is the Kalman gain, and $\mathbf{P}_k$ is the inverse correlation matrix.

2.3 Example: Online Linear Regression

Consider an online linear regression problem where data points arrive sequentially. We can use RLS to update our model parameters with each new data point, allowing the model to adapt to changing patterns in the data.

2.3.1 Example Calculation

Given initial estimates $\boldsymbol{\beta}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\mathbf{P}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, and the first data point $(\mathbf{x}_1, y_1) = (\begin{bmatrix} 1 \\ 2 \end{bmatrix}, 5)$, with $\lambda = 1$:

$$\mathbf{K}_1 = \mathbf{P}_0\mathbf{x}_1(\lambda + \mathbf{x}_1^T\mathbf{P}_0\mathbf{x}_1)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} (1 + \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix})^{-1} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} (1+5)^{-1} = \begin{bmatrix} \frac{1}{6} \\ \frac{2}{6} \end{bmatrix}$$

$$\boldsymbol{\beta}_1 = \boldsymbol{\beta}_0 + \mathbf{K}_1(y_1 - \mathbf{x}_1^T\boldsymbol{\beta}_0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{6} \\ \frac{2}{6} \end{bmatrix} (5 - \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}) = \begin{bmatrix} \frac{5}{6} \\ \frac{10}{6} \end{bmatrix} = \begin{bmatrix} 0.83 \\ 1.67 \end{bmatrix}$$

$$\mathbf{P}_1 = \lambda^{-1}(\mathbf{P}_0 - \mathbf{K}_1\mathbf{x}_1^T\mathbf{P}_0) = 1^{-1}(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{6} \\ \frac{2}{6} \end{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}) = \begin{bmatrix} 0.83 & -0.33 \\ -0.33 & 0.67 \end{bmatrix}$$

The updated parameter estimate after the first data point is $\boldsymbol{\beta}_1 = \begin{bmatrix} 0.83 \\ 1.67 \end{bmatrix}$.

3 Solution to a Linear Equation with Minimum Norm

Finding the solution to a linear equation with minimum norm is important when the system is underdetermined, meaning there are more unknowns than equations. The minimum norm solution minimizes the Euclidean norm of the solution vector.

3.1 Definition

Given a system of linear equations $\mathbf{Ax} = \mathbf{b}$, the minimum norm solution $\mathbf{x}^*$ satisfies:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_2 \quad \text{subject to} \quad \mathbf{Ax} = \mathbf{b} \tag{8}$$

3.2 Mathematical Formulation

The minimum norm solution can be found using the Moore-Penrose pseudoinverse $\mathbf{A}^+$:

$$\mathbf{x}^* = \mathbf{A}^+\mathbf{b} \tag{9}$$

where $\mathbf{A}^+ = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ if $\mathbf{A}$ has full column rank.

3.3 Example: Underdetermined System

Consider the system:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \tag{10}$$

This system has more equations than unknowns, so we seek the minimum norm solution.

3.3.1 Example Calculation

Compute the Moore-Penrose pseudoinverse $\mathbf{A}^+$:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

Compute $\mathbf{A}^T\mathbf{A}$:

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 35 & 44 \\ 44 & 56 \end{bmatrix}$$

Compute $(\mathbf{A}^T\mathbf{A})^{-1}$:

$$(\mathbf{A}^T\mathbf{A})^{-1} = \frac{1}{(35)(56) - (44)(44)} \begin{bmatrix} 56 & -44 \\ -44 & 35 \end{bmatrix} = \frac{1}{196} \begin{bmatrix} 56 & -44 \\ -44 & 35 \end{bmatrix} = \begin{bmatrix} 0.286 & -0.224 \\ -0.224 & 0.179 \end{bmatrix}$$

Compute $\mathbf{A}^T\mathbf{b}$:

$$\mathbf{A}^T\mathbf{b} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 76 \\ 100 \end{bmatrix}$$

Compute $\mathbf{x}^*$:

$$\mathbf{x}^* = \mathbf{A}^+\mathbf{b} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} = \begin{bmatrix} 0.286 & -0.224 \\ -0.224 & 0.179 \end{bmatrix} \begin{bmatrix} 76 \\ 100 \end{bmatrix} = \begin{bmatrix} 1.36 \\ 0.64 \end{bmatrix}$$

The minimum norm solution is $\mathbf{x}^* = \begin{bmatrix} 1.36 \\ 0.64 \end{bmatrix}$.

4 Kaczmarz's Algorithm

Kaczmarz's algorithm is an iterative method for solving systems of linear equations. It is particularly useful for large, sparse systems and is known for its simplicity and efficiency.

4.1 Definition

Kaczmarz's algorithm iteratively projects the current estimate onto the solution hyperplanes defined by the individual equations in the system.

4.2 Mathematical Formulation

Given a system of linear equations $\mathbf{Ax} = \mathbf{b}$, Kaczmarz's algorithm updates the solution estimate $\mathbf{x}_k$ as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{b_i - \mathbf{a}_i^T \mathbf{x}_k}{\|\mathbf{a}_i\|_2^2} \mathbf{a}_i \tag{11}$$

where $\mathbf{a}_i^T \mathbf{x} = b_i$ is the i-th equation in the system.

4.3 Example: Iterative Solution

Consider the system:

$$\begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 5 \end{bmatrix} \tag{12}$$

4.3.1 Example Calculation

Starting with an initial guess $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, we iterate using Kaczmarz's algorithm.

First iteration ($i = 1$):

$$\mathbf{x}_1 = \mathbf{x}_0 + \frac{2 - \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}}{\| \begin{bmatrix} 1 & 1 \end{bmatrix} \|_2^2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{2}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Second iteration ($i = 2$):

$$\mathbf{x}_2 = \mathbf{x}_1 + \frac{5 - \begin{bmatrix} 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\| \begin{bmatrix} 2 & 3 \end{bmatrix} \|_2^2} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \frac{5 - 5}{13} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Third iteration ($i = 1$):

$$\mathbf{x}_3 = \mathbf{x}_2 + \frac{2 - \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{\| \begin{bmatrix} 1 & 1 \end{bmatrix} \|_2^2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \frac{2 - 2}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The solution converges to $\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

5 Linear Equation in General

Linear equations are fundamental in mathematics and have wide applications in science and engineering. A system of linear equations can be represented in matrix form as $\mathbf{Ax} = \mathbf{b}$.

5.1 Full Rank Factorization

If the matrix $\mathbf{A}$ has full rank, it can be factorized as $\mathbf{A} = \mathbf{LU}$, where $\mathbf{L}$ is a lower triangular matrix and $\mathbf{U}$ is an upper triangular matrix. This factorization is useful for solving the system efficiently using forward and backward substitution.

5.2 Example: LU Factorization

Consider the system:

$$\begin{bmatrix} 2 & 3 \\ 5 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 20 \end{bmatrix} \tag{13}$$

5.2.1 Example Calculation

Factorize $\mathbf{A}$ into $\mathbf{L}$ and $\mathbf{U}$:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 5 & 4 \end{bmatrix}$$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 \\ 2.5 & 1 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 2 & 3 \\ 0 & -3.5 \end{bmatrix}$$

Solve $\mathbf{Ly} = \mathbf{b}$ using forward substitution:

$$\begin{bmatrix} 1 & 0 \\ 2.5 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 20 \end{bmatrix}$$

$$y_1 = 8, \quad 2.5 \cdot 8 + y_2 = 20 \implies y_2 = 20 - 20 = 0$$

Solve $\mathbf{Ux} = \mathbf{y}$ using backward substitution:

$$\begin{bmatrix} 2 & 3 \\ 0 & -3.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 0 \end{bmatrix}$$

$$-3.5x_2 = 0 \implies x_2 = 0, \quad 2x_1 + 3 \cdot 0 = 8 \implies x_1 = 4$$

The solution is $\mathbf{x} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$.