Linear Regression: Complete Matrix Derivation

1 Overview

Given a labeled dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, we want to find the optimal parameters $\theta = (\mathbf{w}, w_0)$ that minimize the mean squared error between predictions and truae labels. Bold Symbols are vectors

2 Variable Definitions and Dimensions

We define our variables horizontally as follows: $N \in \mathbb{N}$ (number of samples), $d \in \mathbb{N}$ (number of features), $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$ (feature vector for sample i), $y_i \in \mathbb{R}^{1\times 1}$ (target/label for sample i), $\mathbf{w} \in \mathbb{R}^{d\times 1}$ (weight vector), and $w_0 \in \mathbb{R}^{1\times 1}$ (bias/intercept term).

3 Step 1: Model Definition

The hypothesis function for a single sample:

$$f_\theta(\mathbf{x}_i) = \mathbf{w}^T\mathbf{x}_i + w_0 = w_1x_{i1} + w_2x_{i2} + \cdots + w_dx_{id} + w_0 \tag{1}$$

where $\theta = (\mathbf{w}, w_0)$ are trainable parameters.

4 Step 2: Introducing Residuals and Loss Concept

Before we proceed with matrix formulations, it's crucial to understand what we're trying to optimize. For each data point, our model makes a prediction $\hat{y}_i = f_\theta(\mathbf{x}_i)$, but the true value is y_i. The difference between these is called the **residual** or **error**:

$$e_i = y_i - \hat{y}_i = y_i - (\mathbf{w}^T\mathbf{x}_i + w_0) \tag{2}$$

To find the best parameters, we want to minimize the total error across all samples. A common choice is the **Mean Squared Error (MSE)**:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} e_i^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3}$$

Why do we derive to minimize loss? To find the optimal parameters θ^*, we use calculus. The minimum of a function occurs where its derivative equals zero. By taking the derivative of the loss function with respect to our parameters and setting it to zero, we can solve for the parameter values that minimize our prediction error.

5 Step 3: Augmented Notation

To incorporate the bias term , we define augmented vectors:

$$\tilde{\mathbf{x}}_i = \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix} \in \mathbb{R}^{(d+1)\times 1} \tag{4}$$

$$\tilde{\mathbf{w}} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^{(d+1)\times 1} \tag{5}$$

Now the hypothesis becomes:

$$f_\theta(\mathbf{x}_i) = \tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}} = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i \tag{6}$$

6 Step 4: Design Matrix Construction

Now that we understand our goal (minimizing loss through residuals), we can stack all samples into the design matrix for efficient computation:

$$A = \begin{bmatrix} -\tilde{\mathbf{x}}_1^T - \\ -\tilde{\mathbf{x}}_2^T - \\ \vdots \\ -\tilde{\mathbf{x}}_N^T - \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix} \in \mathbb{R}^{N\times(d+1)} \tag{7}$$

Target vector:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^{N\times 1} \tag{8}$$

7 Step 5: Vectorized Predictions

The prediction vector is obtained by matrix multiplication:

$$\hat{\mathbf{y}} = A\tilde{\mathbf{w}} \in \mathbb{R}^{N\times 1} \tag{9}$$

7.1 Explicit Matrix Multiplication

$$\hat{\mathbf{y}} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix}_{N\times(d+1)} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}_{(d+1)\times 1} \tag{10}$$

$$= \begin{bmatrix} w_0 + w_1 x_{11} + w_2 x_{12} + \cdots + w_d x_{1d} \\ w_0 + w_1 x_{21} + w_2 x_{22} + \cdots + w_d x_{2d} \\ \vdots \\ w_0 + w_1 x_{N1} + w_2 x_{N2} + \cdots + w_d x_{Nd} \end{bmatrix}_{N\times 1} \tag{11}$$

8 Step 6: Residual Vector

The residual (error) vector measures the difference between true and predicted values:

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - A\tilde{\mathbf{w}} \in \mathbb{R}^{N\times 1} \tag{12}$$

8.1 Component-wise Form

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = \begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \\ y_N - \hat{y}_N \end{bmatrix} = \begin{bmatrix} y_1 - (w_0 + w_1 x_{11} + \cdots + w_d x_{1d}) \\ y_2 - (w_0 + w_1 x_{21} + \cdots + w_d x_{2d}) \\ \vdots \\ y_N - (w_0 + w_1 x_{N1} + \cdots + w_d x_{Nd}) \end{bmatrix} \tag{13}$$

9 Step 7: Loss Function

Mean Squared Error (MSE):

$$\mathcal{L}(\tilde{\mathbf{w}}) = \frac{1}{N}\mathbf{e}^T\mathbf{e} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{N}||\mathbf{y} - A\tilde{\mathbf{w}}||_2^2 \tag{14}$$

10 Step 8: Gradient Derivation via Chain Rule

Why we derive: To minimize the loss function $\mathcal{L}(\tilde{\mathbf{w}})$, we need to find where its gradient equals zero. This gives us the optimal parameters that minimize prediction error.

10.1 Chain Rule Setup

$$\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{w}}} = \frac{\partial \mathcal{L}}{\partial \mathbf{e}} \cdot \frac{\partial \mathbf{e}}{\partial \tilde{\mathbf{w}}} \tag{15}$$

Key insight: We will compute this as a matrix product where:

- $\frac{\partial \mathcal{L}}{\partial \mathbf{e}}$ is a row vector of shape $1 \times N$
- $\frac{\partial \mathbf{e}}{\partial \tilde{\mathbf{w}}}$ is a matrix of shape $N \times (d+1)$
- The product gives a row vector of shape $1 \times (d+1)$

10.2 First Factor: $\frac{\partial \mathcal{L}}{\partial \mathbf{e}}$

Since $\mathcal{L} = \frac{1}{N}\mathbf{e}^T\mathbf{e} = \frac{1}{N}\sum_{i=1}^{N} e_i^2$:

$$\frac{\partial \mathcal{L}}{\partial e_i} = \frac{2}{N}e_i \tag{16}$$

Collecting all partials into a row vector:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{e}} = \frac{2}{N}[e_1, e_2, \ldots, e_N] = \frac{2}{N}\mathbf{e}^T \in \mathbb{R}^{1\times N} \tag{17}$$

10.3 Second Factor: $\frac{\partial \mathbf{e}}{\partial \tilde{\mathbf{w}}}$

Since $\mathbf{e} = \mathbf{y} - A\tilde{\mathbf{w}}$ and $\mathbf{y}$ is constant:

$$\frac{\partial \mathbf{e}}{\partial \tilde{\mathbf{w}}} = -A \in \mathbb{R}^{N\times(d+1)} \tag{18}$$

10.4 Matrix Multiplication

10.4.1 Explicit Visual Form

$$\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{w}}} = \frac{2}{N}\underbrace{\begin{bmatrix} e_1 & e_2 & \cdots & e_N \end{bmatrix}}_{\mathbb{R}^{1\times N}} \underbrace{\begin{bmatrix} -1 & -x_{11} & -x_{12} & \cdots & -x_{1d} \\ -1 & -x_{21} & -x_{22} & \cdots & -x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -x_{N1} & -x_{N2} & \cdots & -x_{Nd} \end{bmatrix}}_{\mathbb{R}^{N\times(d+1)}} \tag{19}$$

10.4.2 Computing the Product

$$= -\frac{2}{N}\begin{bmatrix} e_1 & e_2 & \cdots & e_N \end{bmatrix}\begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix} \tag{20}$$

$$= -\frac{2}{N}\begin{bmatrix} \sum_{i=1}^{N} e_i & \sum_{i=1}^{N} e_i x_{i1} & \sum_{i=1}^{N} e_i x_{i2} & \cdots & \sum_{i=1}^{N} e_i x_{id} \end{bmatrix}_{\mathbb{R}^{1\times(d+1)}} \tag{21}$$

This can be written compactly since each entry is a dot product between the error vector and one column of A:

$$\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{w}}} = -\frac{2}{N}\mathbf{e}^T A \in \mathbb{R}^{1\times(d+1)} \tag{22}$$

10.5 Transpose to Column Gradient

$$\nabla_{\tilde{\mathbf{w}}}\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{w}}}\right)^T \tag{23}$$

$$= \left(-\frac{2}{N}\mathbf{e}^T A\right)^T \tag{24}$$

$$= -\frac{2}{N}A^T\mathbf{e} \in \mathbb{R}^{(d+1)\times 1} \tag{25}$$

10.6 Substituting $\mathbf{e} = \mathbf{y} - A\tilde{\mathbf{w}}$

$$\nabla_{\tilde{\mathbf{w}}}\mathcal{L} = -\frac{2}{N}A^T(\mathbf{y} - A\tilde{\mathbf{w}}) \tag{26}$$

$$= -\frac{2}{N}A^T\mathbf{y} + \frac{2}{N}A^T A\tilde{\mathbf{w}} \tag{27}$$

$$= \frac{2}{N}A^T(A\tilde{\mathbf{w}} - \mathbf{y}) \tag{28}$$

11 Step 9: Normal Equation Derivation

11.1 Setting Gradient to Zero

For the minimum, set $\nabla_{\tilde{\mathbf{w}}}\mathcal{L} = \mathbf{0}$:

$$\frac{2}{N}A^T(A\tilde{\mathbf{w}} - \mathbf{y}) = \mathbf{0}_{(d+1)\times 1} \tag{29}$$

11.2 Simplifying

$$A^T(A\tilde{\mathbf{w}} - \mathbf{y}) = \mathbf{0} \tag{30}$$

$$A^T A\tilde{\mathbf{w}} - A^T\mathbf{y} = \mathbf{0} \tag{31}$$

$$A^T A\tilde{\mathbf{w}} = A^T\mathbf{y} \tag{32}$$

This is the **Normal Equation**.

11.3 Solving for $\tilde{\mathbf{w}}^*$

If $A^TA \in \mathbb{R}^{(d+1)\times(d+1)}$ is invertible (full column rank):

Step 1: Multiply both sides by $(A^TA)^{-1}$ on the left:

$$\underbrace{(A^TA)^{-1}}_{\mathbb{R}^{(d+1)\times(d+1)}}\ \underbrace{(A^TA)}_{\mathbb{R}^{(d+1)\times(d+1)}}\ \underbrace{\tilde{\mathbf{w}}}_{\mathbb{R}^{(d+1)\times 1}} = \underbrace{(A^TA)^{-1}}_{\mathbb{R}^{(d+1)\times(d+1)}}\ \underbrace{A^T}_{\mathbb{R}^{(d+1)\times N}}\ \underbrace{\mathbf{y}}_{\mathbb{R}^{N\times 1}} \tag{33}$$

Step 2: Apply the identity property $(A^TA)^{-1}(A^TA) = I_{(d+1)}$:

$$\underbrace{I_{(d+1)}}_{\mathbb{R}^{(d+1)\times(d+1)}}\ \underbrace{\tilde{\mathbf{w}}}_{\mathbb{R}^{(d+1)\times 1}} = \underbrace{(A^TA)^{-1}A^T\mathbf{y}}_{\mathbb{R}^{(d+1)\times 1}} \tag{34}$$

Step 3: Since $I_{(d+1)}\tilde{\mathbf{w}} = \tilde{\mathbf{w}}$:

$$\tilde{\mathbf{w}} = (A^TA)^{-1}A^T\mathbf{y} \tag{35}$$

Therefore, the optimal solution is:

$$\boxed{\tilde{\mathbf{w}}^* = (A^TA)^{-1}A^T\mathbf{y} \in \mathbb{R}^{(d+1)\times 1}} \tag{36}$$

12 Dimension Verification

Matrix/Vector	Dimensions
A	$\mathbb{R}^{N\times(d+1)}$
A^T	$\mathbb{R}^{(d+1)\times N}$
A^TA	$\mathbb{R}^{(d+1)\times(d+1)}$
$(A^TA)^{-1}$	$\mathbb{R}^{(d+1)\times(d+1)}$
$\mathbf{y}$	$\mathbb{R}^{N\times 1}$
$A^T\mathbf{y}$	$\mathbb{R}^{(d+1)\times 1}$
$(A^TA)^{-1}A^T\mathbf{y}$	$\mathbb{R}^{(d+1)\times 1}$

13 Separating Weights and Bias

From $\tilde{\mathbf{w}}^* = \begin{bmatrix} w_0^* \\ \mathbf{w}^* \end{bmatrix}$, we can derive:

$$w_0^* = \bar{y} - (\mathbf{w}^*)^T\bar{\mathbf{x}} \tag{37}$$

$$\mathbf{w}^* = (X^TX)^{-1}X^T(\mathbf{y} - \bar{y}\mathbf{1}) \tag{38}$$

where $\bar{y}$ is the mean of targets, $\bar{\mathbf{x}}$ is the mean of features, and X is A without the bias column.