Mathematical Intuition: Equations, Cost Function & Optimization

Saturday, December 6, 2025 6:29 PM

MLR formula:

$$\hat{y} = w_1x_1 + w_2x_2 + \dots + w_px_p + b$$

$$y = mx + c$$

$w \rightarrow$

$b \rightarrow$

Monthly rent:

x_1 = size (sq ft)

x_2 = no of bedroom

x_3 = floor

$w_1 = 2$, $w_2 = 1.5$, $w_3 = 0.2$, $b = 5$

Datapoint: $x_1 = 8$, $x_2 = 3$, $x_3 = 6$

prediction:

$$\hat{y} = 2\times8 + 1.5\times3 + 0.2\times6 + 5 = 26.7$$ thousands taka

The dataset with n samples and p features:

The prediction for all sample is?

$$\hat{y} = X w + b\mathbf{1}$$

$X \rightarrow$ feature matrix

$b \rightarrow$ bias

$w \rightarrow$ weight vector

$\mathbf{1} \rightarrow$ vector of 1 of ones of length n.

Consider two samples and two features:

$$X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad w = \begin{bmatrix} 10 \\ 5 \end{bmatrix}, \quad b = 1$$

Compute:

$$Xw = \begin{bmatrix} 1\times10 + 2\times5 \\ 3\times10 + 4\times5 \end{bmatrix} = \begin{bmatrix} 20 \\ 50 \end{bmatrix}$$

Then we add the bias:

$$\hat{y} = Xw + b\mathbf{1} = \begin{bmatrix} 20 \\ \dots \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 21 \\ 51 \end{bmatrix}$$

Then we add the bias:

$$\hat{y} = \underline{Xw + b1} = \begin{bmatrix} 20 \\ 50 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \ldots \\ 51 \end{bmatrix}$$

* vector of ones of dimension $n \times \underline{1}$

Cost function:

Formula: $J(w,b) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

MSE

Vector form using $\hat{y} = Xw + b1$

$$J(w,b) = \frac{1}{2n} \| y - (Xw + b1) \|_2^2$$

$\| \cdot - \|_2$: Euclidean norm

<u>Given,</u>

$y = [10, 12, 14]$ $\hat{y} = [9, 13, 15]$

Residuals = $e = y - \hat{y} = [1, -1, -1]$

Squared error = $[1, 1, 1]$

Then, $J = \frac{1}{2 \cdot 3} (1+1+1) = \frac{1}{6} \times 3 = \frac{3}{6} = 0.5$

Gradient Descent:

Cost: $J(w) = \frac{1}{2n} \| y - Xw \|_2^2$

... cost with respect to w:

errors/ residuals

Cost: $J(w) - 2n$

Gradient of the cost with respect to w:

$$\underbrace{\nabla_w J(w)}_{\text{Gradient of the cost respect to } w} = -\frac{1}{n} X^T \overbrace{(y - Xw)}^{\text{errors/ residuals}}$$

X^T → transpose of the feature matrix X

The gradient descent update is:

$$w_{new} = w_{old} - \alpha \nabla_w J(w_{old})$$

α is the learning rate

$[0.001, 0.0001, 0.01]$

Substituting the gradient:

$$w_{new} = w_{old} + \frac{\alpha}{n} X^T (y - X w_{old})$$

w_{new} → updated weight

w_{old} → current weight before update

Suppose,

$w_{old} = 2$, $\nabla_w J(w_{old}) = -0.6$, $\alpha = 0.1$

Then,

$w_{new} = 2 - 0.1(-0.6) = 2 + 0.06 = 2.06$

0.06 step

Polynomial Features and Overfitting

Saturday, December 6, 2025 10:07 PM

Feature map and model:

For a single feature x and polynomial degree d,

feature map

$$\varphi_d(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \\ \vdots \\ x^d \end{bmatrix}$$

The model becomes:

$$\hat{y} = w_1 x + w_2 x^2 + w_3 x^3 + \cdots + w_d x^d + b$$
$$= w^T \varphi_d(x) + b$$

Example:

Led $d = 3$,

and, $\varphi_3(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$, $w = \begin{bmatrix} 1 \\ -0.5 \\ 0.1 \end{bmatrix}$, $b = 0.1$

For, $x = 2$

$$\hat{y} = 1 \times 2 + (-0.5) \times 4 + 0.1 \times 8 + 0.1 = 0.9$$

x, y

x^2, y^2, xy

* Polynomial features for multiple variables

Number of terms $= \binom{p+d}{d}$ ✓

$\binom{p+d}{d} - 1$

Case 1: $p = 1, d = 3$

Number of terms $= \binom{1+3}{3} = \binom{4}{3} = {}^4C_3 = 4$

Terms: $1, x, x^2, x^3$

Number of ...

Terms: $1, x, x^2, x^3$

Case 2: $p = 2, d = 2$

Number of terms $= \binom{2+2}{2} = \binom{4}{2} = {}^4C_2 = 6$

terms $= 1, x_1, x_2, x_1^2, x_1 x_2, x_2^2$

* polynomial regression as linear regression in feature space

ϕ can have more columns than X

$$\phi = \begin{bmatrix} \phi_d(x^{(1)})^T \\ \phi_d(x^{(2)})^T \\ \vdots \\ \phi_d(x^{(n)})^T \end{bmatrix}$$

$\hat{y} = WX + b$

$\Rightarrow \hat{y} = wX$

The model $\Rightarrow \hat{y} = \phi w$

* Training error versus polynomial degree

Let,

$J_{train}(d)$ be the minimum training error when using degree d.

Then, $J_{train}(1) \geq J_{train}(2) \geq J_{train}(d)$

Example;

degree 1, 2, 3, we observe

$J_{train}(1) = 5.0$, $J_{train}(2) = 3.0$, $J_{train}(3) = 1.2$

* Test error and bias variance

Approximate decomposition:

$$\text{Error} \approx \text{bias}(d)^2 + \text{variance}(d) + \text{noise}$$

bias $\downarrow$ Systematic error

variance $\downarrow$ error for model being too sensative to train data

noise $\downarrow$ Error due to randomness in the data

Increasing d $\rightarrow$ bias $\downarrow$, variance $\uparrow$

Hypothetical situation:

→ Degree 1: bias^2 = 9, variance = 1 → total error ≈ 10

→ Degree 3: " = 4 " = 3 → " ≈ 7

→ " 10: " = 1 " = 20 → " ≈ 21