MLR formula:

$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_p x_p + b$$

$$y = mx + c$$

$\vec{w}$
$\vec{b}$

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:

$$\hat{y} = 2 \times 8 + 1.5 \times 3 + 0.2 \times 6 + 5 = 26.7 \text{ thousands taka}$$

The dataset with n samples and p features:

The prediction for all sample is?

→ feature matrix

$$\hat{y} = Xw + b\mathbf{1}$$

→ bias

→ weight vector

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:

$$Xw = \begin{bmatrix} 1 \times 10 + 2 \times 5 \\ 3 \times 10 + 4 \times 5 \end{bmatrix} = \begin{bmatrix} 20 \\ 50 \end{bmatrix}$$

Then we add the bias!

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

Then we add the bias:

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

* vector of ones of dimension $n \times 1$

Cost function:

Formula:

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

(MSE)

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

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

$|| -- ||_2$: Euclidean norm

Given,

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

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

$$\text{squared error} = [1, 1, 1]$$

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: $\nearrow$ errors/residuals

Cost: $J(w) - 2n$

Gradient of the cost with respect to w:

$$\nabla_w J(w) = -\frac{1}{n} X^T \underbrace{(y - Xw)}_{}$$

errors/residuals

$\underbrace{\nabla_w J(w)}_{}$ — Gradient of the cost respect to w

$\searrow$ 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:

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

$\swarrow$ updated weight   current weight before update

Suppose,

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

Then,

$$w_{new} = 2 - 0.1(-0.6) = 2 + 0.06 = \boxed{2 \enspace 06}$$

0.06 step

Feature map and model:

For a single feature x and polynomial degree d,

feature map

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

The model becomes:

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

Example:

Let $d = 3$,

and,

$$\phi_3(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix} \quad, \quad w = \begin{bmatrix} 1 \\ -0.5 \\ 0.1 \end{bmatrix} \quad, \quad b = 0.1$$

For, $x = 2$

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

* polynomial features for multiple variables

$\quad\quad\quad\quad\quad\quad\quad\quad\quad x, y$

$\quad\quad\quad\quad\quad\quad\quad\quad\quad x^2, y^2, xy$

Number of terms $= \dbinom{p+d}{d}$ ✓

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

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

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

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 $= \begin{pmatrix} 2+2 \\ 2 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \end{pmatrix} = {}^4C_2 = 6$

Terms $= 1, x_1, x_2, x_1^2, x, x_2, x_2^2$

*Polynomial regression as linear regression in feature space

φ can have more columns than X

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

$\hat{y} = WX \boxed{+b}$

$\Rightarrow \hat{y} = wX \leftarrow$

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

*Training error versus polynomial degree

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

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

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:

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

↓ Systematic error

↓ error for model being too sensative to train data

↓ Error due to randomness in the data

Increasing d → ↓   ↗↑

Hypothetical situation:

→ Degree 1: bias² = 9, variance = 1 → total error ≈ 10

→ Degree 3: " = 4   " = 3 → " ≈ 7

→ " 10: " = 1   " = 20 → " ≈ 21