Week 3: Multiple Linear Regression

Ekarat Rattagan

July 28, 2025

Week 3: Multiple Variable Linear Regression

- A data set $X \in \mathbb{R}^{N \times d}$ that has N rows and d dimensions.
- $h_\theta(x) = \theta_0 + \theta_1 x$, a hypothesis or model.
- Notation: $x_{i,j}$ means a sample at row i, column j.

$$\begin{bmatrix} 1 & 2104 & 460 \\ 1 & 1416 & 232 \\ 1 & 1534 & 315 \\ \vdots & \vdots & \vdots \end{bmatrix} \Rightarrow \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & y_1 \\ x_{2,1} & \cdots & \cdots & \cdots & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & \cdots & \cdots & \cdots & y_N \end{bmatrix}$$

Model Representation

- **Single variable:**

$$h_\theta(x) = \theta_0 + \theta_1 x$$

- **Multiple variables:**

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d = \theta^T x$$

- **Matrix Form:**

$$\theta^T = [\theta_0, \theta_1, \ldots, \theta_d], \quad x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

Cost Function and Batch Gradient Descent

- **Cost Function:**

$$J(\theta_0, \theta_1, \ldots, \theta_d) = \frac{1}{2N} \sum_{i=1}^{N} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

- **Goal:** Find best θ_j to minimize J

- **Batch Gradient Descent (BGD):**

$$\theta_j := \theta_j - \eta \frac{\partial J(\theta)}{\partial \theta_j}$$

BGD: Loop Until Converge

- Update parameters using all training samples:

$$\theta_0 := \theta_0 - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_{i,0}$$

$$\theta_1 := \theta_1 - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_{i,1}$$

$$\cdots$$

$$\theta_d := \theta_d - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_{i,d}$$

- Result: $\theta^* = \arg\min\,(\text{half MSE})$

Feature Scaling

- Problem: $x_2 \gg x_1$ leads to slow/unstable gradient descent
- Solution: Standardize or normalize features
- Min-Max scaling: $x' \in [0, 1]$
- Z-score: $x' = \frac{x-\mu}{\sigma}$ to ensure $\approx [-3, 3]$
- Helps gradients move in balanced directions

Scaling Methods: Min-Max vs Z-score

- **S1: Min-Max Scaling**

$$x'_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$$

- **S2: Z-score Scaling**

$$x'_i = \frac{x_i - \mu}{\sigma} \quad \text{(use if data is normally distributed)}$$

- **Question:** When to use each? Depends on data distribution.

BGD vs SGD

- **BGD:** Uses full dataset in each iteration

$$\theta_j := \theta_j - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_{i,j}$$

- **SGD:** Uses one sample per iteration

$$\theta_j := \theta_j - \eta(h_\theta(x^{(i)}) - y^{(i)}) \cdot x_{i,j}$$

- SGD introduces more noise, but converges faster
- **Question:** How to reduce SGD noise?

SGD: Loop Until Converge

- For random $i \in [1, N]$, update:

$$\theta_0 := \theta_0 - \eta \cdot (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_{i,0}$$

$$\theta_1 := \theta_1 - \eta \cdot (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_{i,1}$$

$$\cdots$$

$$\theta_d := \theta_d - \eta \cdot (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_{i,d}$$

- Result: θ^* minimizes half MSE

Learning Rate Scheduling

- Control step size η during training
- Reduce η over iterations:

$$\eta_{t+1} = \frac{\eta_0}{1 + \eta_0 \lambda t} \quad \text{where } t = 1, \ldots, M$$

- Example: Let $\eta_0 = 0.01$, $\lambda = 0.1$

$$\eta_2 = \frac{0.01}{1 + 0.01 \cdot 0.1 \cdot 1} = 0.0099$$

- Reference: Bottou (2012) – Stochastic Gradient Descent Tricks

BGD vs SGD vs Mini-batch GD

- **BGD:** Use all N samples
- **SGD:** Use one sample ($b = 1$)
- **Mini-batch GD:** Use b samples, where $1 < b < N$ (e.g. $b = 10$)
- **Performance:**
  - BGD: stable path to minimum
  - SGD: fast but noisy
  - Mini-batch: compromise between stability and speed

Mini-batch GD smooths out noise while accelerating convergence.

Polynomial Regression

- Use higher-degree terms of input features
- Total number of terms: combination formula

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}, \quad n = \#\text{features} + \text{degree}, \quad r = \text{degree}$$

- Example: x_1, x_2, degree $d = 2$

$$\binom{4}{2} = \frac{4!}{2!(4-2)!} = \frac{4 \cdot 3 \cdot 2 \cdot 1}{2 \cdot 2 \cdot 1 \cdot 1} = 6$$

- Polynomial hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2$$

Reference

- Uyanık,G.K.,and Güler,N.(2013). A study on multiple linear regression analysis. Procedia-Social and Behavioral Sciences, 106, 234-240