Week 3: Multiple Linear Regression

Ekarat Rattagan

August 17, 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.

$$\begin{bmatrix} 1 & 2104 & 460 \\ 1 & 1416 & 232 \\ 1 & 1534 & 315 \\ \vdots & \vdots & \vdots \end{bmatrix} \Rightarrow \begin{bmatrix} x_{0,1} & x_{0,2} & x_{0,3} & x_{0,4} & y_0 \\ x_{1,1} & \cdots & \cdots & \cdots & y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & \cdots & \cdots & y_{N-1} \end{bmatrix}$$

Model Representation

- **Single variable:**

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

- **Multiple variables:**

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

- **Matrix Form:**

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

Cost Function and Batch Gradient Descent

- **Cost Function:**

$$J(\theta_0, \theta_1, \ldots, \theta_d) = \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x_i) - y_i)^2$$

- **Goal:** Find best θ_j to minimize J
- **Batch Gradient Descent (BGD):**

$$\theta_j^{t+1} = \theta_j^t - \eta \frac{\partial J(\theta)}{\partial \theta_j}$$

BGD: Loop Until Converge

- Update parameters using all training samples:

$$\theta_0^{t+1} = \theta_0^t - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x_i) - y_i) \cdot x_{i,0}$$

$$\theta_1^{t+1} := \theta_1^t - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x_i) - y_i) \cdot x_{i,1}$$

$$\dots$$

$$\theta_d^{t+1} := \theta_d^t - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x_i) - y_i) \cdot x_{i,d}$$

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**

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

- **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.

Batch GD vs Stochastic GD

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

$$\theta_j^{t+1} := \theta_j^t - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} (h_\theta(x_i) - y_i) \cdot x_{i,j}$$

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

$$\theta_j^{t+1} := \theta_j^t - \eta (h_\theta(x_i) - y_i) \cdot x_{i,j}$$

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

Learning Rate Scheduling

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

$$\eta^{t+1} = \frac{\eta^0}{1 + \eta^0 \lambda t} \quad \text{where } t = 1, \ldots, T$$

- 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 (randomly selection per loop)
- **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