Machine Learning Notes

Amit Thakur

January 3, 2025

Contents

Chapter 1

Linear Regression

Linear Regression is a statistical method used for modeling the relationship between a dependent variable (target or output) and one or more independent variables (predictors or features).

1.1 Linear Regression with Single Feature

This regression deals with one independent variable (x).

1.1.1 Linear Regression Model

$$y = \theta_0^t + \theta_1^t x + \epsilon \tag{1.1}$$

$$\hat{y} = h_\theta(x) = \theta_0 + \theta_1 x \tag{1.2}$$

where:

- y: the dependent variable (target)
- x: the independent variable or input feature used to predict y.
- θ_1^t: the slope or coefficient of x at iteration t
- θ_0^t: intercept at iteration t
- ϵ: the error term or residual. It captures the noise or other unmodeled effects.
- $\hat{y}$: the output of the linear regression model () for a given input (x)
- $h_\theta(x)$: the hypothesis function for linear regression.

1.1.2 Cost Function

The cost function for linear regression measures the average squared error between predicted and actual values. It is defined as:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right)^2 = \frac{1}{2m} \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x^{(i)}) - y^{(i)}\right)^2 \tag{1.3}$$

where:

- m: Number of input and output datapoints.
- $\frac{1}{2}$: A factor kept for convenience, as it simplifies the derivative calculations during gradient descent.

1.1.3 Minimizing Cost

The error for each data point can be written as:

$$e^{(i)} = y^{(i)} - (\theta_0 + \theta_1 x^{(i)}) \tag{1.4}$$

The cost function becomes:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} e^{(i)^2} \tag{1.5}$$

We need to find θ_0 and θ_1 that minimize J. This requires setting the partial derivatives of J with respect to θ_0 and θ_1 to zero as the cost function is quadratic and there's just one critical point.

Solving for θ_0:

$$\frac{\partial J}{\partial \theta_0} = -\frac{1}{m} \sum_{i=1}^{m} \left(y^{(i)} - (\theta_0 + \theta_1 x^{(i)})\right) = 0$$

$$\sum_{i=1}^{m} \left(y^{(i)} - (\theta_0 + \theta_1 x^{(i)})\right) = 0$$
$$\sum_{i=1}^{m} y^{(i)} = m\theta_0 + \theta_1 \sum_{i=1}^{m} x^{(i)}$$

Divide through by m:

$$\bar{y} = \theta_0 + \theta_1 \bar{x}$$

where $\bar{y}$ and $\bar{x}$ are the means of $y^{(i)}$ and $x^{(i)}$, respectively.
Rearranging:

$$\theta_0 = \bar{y} - \theta_1 \bar{x} \tag{1.6}$$

Solving for θ_1:

$$\frac{\partial J}{\partial \theta_1} = -\frac{1}{m} \sum_{i=1}^{m} x^{(i)} \left(y^{(i)} - (\theta_0 + \theta_1 x^{(i)})\right) = 0$$

Expanding:

$$\sum_{i=1}^{m} x^{(i)} y^{(i)} = \theta_0 \sum_{i=1}^{m} x^{(i)} + \theta_1 \sum_{i=1}^{m} \left(x^{(i)}\right)^2$$

Substitute $\theta_0 = \bar{y} - \theta_1 \bar{x}$:

$$\sum_{i=1}^{m} x^{(i)} y^{(i)} = (\bar{y} - \theta_1 \bar{x}) \sum_{i=1}^{m} x^{(i)} + \theta_1 \sum_{i=1}^{m} (x^{(i)})^2$$

Simplify:

$$\sum_{i=1}^{m} x^{(i)} y^{(i)} = \bar{y} \sum_{i=1}^{m} x^{(i)} - \theta_1 \bar{x} \sum_{i=1}^{m} x^{(i)} + \theta_1 \sum_{i=1}^{m} (x^{(i)})^2$$

Reorganize terms:

$$\theta_1 \left(\sum_{i=1}^{m} (x^{(i)})^2 - \frac{1}{m} \left(\sum_{i=1}^{m} x^{(i)} \right)^2 \right) = \sum_{i=1}^{m} x^{(i)} y^{(i)} - \frac{1}{m} \sum_{i=1}^{m} x^{(i)} \sum_{i=1}^{m} y^{(i)}$$

Using simplified notation:

- $\bar{x} = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$ (mean of x)
- $\bar{y} = \frac{1}{m} \sum_{i=1}^{m} y^{(i)}$ (mean of y)

$$\theta_1 = \frac{\sum_{i=1}^{m} \left(x^{(i)} - \bar{x} \right) \left(y^{(i)} - \bar{y} \right)}{\sum_{i=1}^{m} \left(x^{(i)} - \bar{x} \right)^2} \tag{1.7}$$

1.1.4 Gradient Descent Algorithm to find optimal θ_0 and θ_1

1. Gradient descent moves the parameters in the direction of the negative gradient (steepest descent) of the cost function.

2. The learning rate α determines the size of each step. If α is too large, the algorithm may overshoot the minimum. If it is too small, convergence may take too long.

3. The iterative process ensures gradual improvement in the model parameters until the cost function is minimized.

Algorithm 1 Gradient Descent for Linear Regression

1: **Input:** Learning rate α, initial values for θ_0 and θ_1, and maximum iterations or convergence threshold ϵ.
2: **Output:** Optimized parameters θ_0 and θ_1.
3: Set initial values for θ_0 and θ_1.
4: **repeat**
5: Compute updates for parameters:

$$\text{temp0} := \theta_0 - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)$$

$$\text{temp1} := \theta_1 - \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right) x^{(i)}$$

6: Update parameters simultaneously:

$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

7: Compute cost function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

8: **until** maximum iterations reached **or** change in $J(\theta_0, \theta_1)$ is below threshold ϵ.
9: **return** θ_0, θ_1

Bibliography

[1] Weidong Kuang. *Fundamentals of deep learning: a step-by-step guide.* Self, 2024. `https://faculty.utrgv.edu/weidong.kuang/book/book.html`(visited 2024-12-01).

[2] Simon J.D. Prince. *Understanding Deep Learning.* The MIT Press, 2023.

[3] Amit Thakur. *Maths for ML Notes.* Self, 2025. `http://amitthakur.org/learnings/maths-for-ml`.