Linear Regression and Multiple Linear Regression with Gradient Descent

Lauvindra Raj

July 2021

1) Introduction

1. Regression is a task when a model attempts to predict continuous values.

2. The purpose regression in Machine Learning is for prediction. Linear Regression is a widely used algorithm in this field.

3. It attempts to model the relationship between TWO variables by fitting a ”best-fit” line to the observed data points where the ”best-fit” line has the minimum sum of the squares of the vertical distance from each data point to the ”best-fit” line.

4. This is where correlation takes places as it has the definition of measure how strong a relationship between two variables.

5. Least-Squares Regression is a method used for fitting a regression line by calculating the “best-fit” line by *minimizing the sum of squares* of vertical deviations from each data points to the line.

6. Dependent and Independent variable are the variables in Linear regression. The main idea is to derive the independent variable using the dependent variable.

7. In Multiple Linear Regression, there are *more than one dependent variable* and *exactly one independent variable.*

2) Algorithm for Linear Regression and Multiple Linear Regression

Input given are as follows :

$$\mathrm{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{bmatrix}, y \quad = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \qquad (1)$$

Each row in X is the iteration *(i-th)* sample. Each column in X represents the feature (dependent variable) of the dataset.
The goal is to find linear function of h approximate to $y^{(i)}$ given $x^{(i)}$.

The linear function h is :

$$h_\theta\left(x^{(i)}\right) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots \theta_n x_n^{(i)} \qquad (2)$$

or can be written as :

$$h_\theta\left(x^{(i)}\right) = \sum_{j=0}^{n} \theta_j \, x_j \qquad (3)$$

We assume $x_0^{(i)} = 1$ as it is the *intercept term*, and to simplify notation for the finding of constant in the linear equation.

When $\theta_j \in \mathbb{R}$ and i = 1, … , m, such that :

We use loss function (*least square function)* :

$$\frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2 \tag{4}$$

Note : $\hat{y}$ means y estimates

In this case:

$$\hat{y} = h_\theta(x) \tag{5}$$

Hence:

$$J(\theta) = \frac{1}{2}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \tag{6}$$

$$sum = \frac{1}{m}\sum_{j=0}^{n} h_\theta\left(x^{(i)}\right) \tag{7}$$

Subs eq 7 into 6 :

$$J(\theta_0, \dots, \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \tag{8}$$

3) Additional Notes

Taking θ as a vector ,

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \tag{9}$$

Eq (2) and (3) can we written as:

$$h_\theta\left(x^{(i)}\right) = \theta^T x^{(i)} \tag{10}$$

To further continue this equation, we need to understand about least square using matrix derivatives.

3.1) Least Square Using Matrix Derivatives

Training examples :

$$X = \begin{pmatrix} \cdots & \left(x^{(1)}\right)^T & \cdots \\ \cdots & \left(x^{(2)}\right)^T & \cdots \\ \cdots & \left(x^{(3)}\right)^T & \cdots \end{pmatrix}$$

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

$\vec{y}$ is the n-dimensional vector containing all targets from training set

$$X\theta = \begin{bmatrix} (x^1)^T\theta \\ \vdots \\ \left(x^{(n)}\right)^T\theta \end{bmatrix}$$

From eq (10):

$$X\theta - \vec{y} = \begin{bmatrix} (x^1)^T\theta \\ \vdots \\ \left(x^{(n)}\right)^T\theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

$$= \begin{bmatrix} h_\theta\left(x^{(1)}\right) - y^{(1)} \\ \vdots \\ h_\theta\left(x^{(n)}\right) - y^{(n)} \end{bmatrix}$$

$$\frac{1}{2m}(X\theta - \vec{y})^T(X\theta - \vec{y}) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \tag{11}$$

We can simplify the eq as:

$$(X\theta - \vec{y})^T(X\theta - \vec{y}) = \sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \tag{12}$$

Eq (7) can be written as

$$J(\theta) = \frac{1}{2m}(X\theta - y)^T(X\theta - y) \tag{13}$$

4) Gradient Descent

- This is an algorithm that repeatedly changes θ to minimize $J(\theta)$ until a stopping criterion is met.

$$\theta_j = \theta_j - \alpha \frac{d}{d\theta_j} J(\theta) \tag{14}$$

$\alpha = Learning\ Rate$

$\theta_j = Parameters$

Where j = 0 … , n and $\alpha \in \mathbb{R}$ positive. Usually, the α will be in the range of 0 and 1.

Before deriving this algorithm , let's take a look at the concept of partial derivative and basic differentiation :

Basic differentiation concept :

$$\frac{\partial}{\partial x}(ax+b)^n = n(ax+b)^{n-1}(a)$$
$$= (n)(a)(ax+b)$$

We take the eq from eq (8) :

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

We differentiate this eq to find θ_j:

$$\frac{\partial}{\partial(\theta_j)} J(\theta) = \frac{\partial}{\partial(\theta_j)} \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$\frac{\partial}{\partial(\theta_j)} J(\theta) \; = \; \frac{1}{2m}.(2)\sum_{i=1}^{m}\frac{\partial}{\partial\left(\theta_j\right)}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right).\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$= \frac{1}{m}\frac{\partial}{\partial\left(\theta_j\right)}\left(\sum_{j=0}^{n}\theta_j\, x_j{}^{(i)} - y^{(i)}\right).\left(\sum_{i=1}^{m} h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left[x_j{}^{(i)}.\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)\right]$$

$$= \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right).x_j{}^{(i)} \tag{15}$$

Then, from the eq 12, we can also write this as :

$$\frac{\partial}{\partial\left(\theta_j\right)}J(\theta) = \frac{1}{m}\,(X\theta - y)x_{\,j} \tag{16}$$