Linear Regression and Multiple Linear Regression with Gradient Descent

Ugenteraan Manogaran

February 9, 2019

1 Introduction

1. Linear Regression is a machine learning algorithm.
2. 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.
3. The method of minimizing the sum of the squares of the vertical distance from each data point to the line is known as the method of least-squares.
4. The variables in Linear Regression is known as dependent variable and independent variable. The idea is to derive the independent variable using the dependent variable.
5. 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 using Gradient Descent

Suppose the inputs are :

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & . & . & . & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & . & . & . & x_n^{(2)} \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ x_1^{(m)} & x_2^{(m)} & . & . & . & x_n^{(m)} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ . \\ . \\ . \\ y^{(n)} \end{bmatrix} \tag{1}$$

where each row in $\mathbf{X}$ is the *i-th* sample. Each column in $\mathbf{X}$ represents the feature (dependent variable) of the dataset.

The goal is to find a linear function $\mathbf{h}$ to approximate $y^{(i)}$, given $\mathbf{x}^{(i)}$

$x_0^{(i)}$ will be added into $\mathbf{X}$ where $x_0^{(i)} = 1$ to simplify the notations for the finding of the constant in the linear equation later. Hence,

$$\mathbf{X} = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} & . & . & . & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & . & . & . & x_n^{(2)} \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ x_0^{(m)} & x_1^{(m)} & . & . & . & x_n^{(m)} \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ . \\ . \\ . \\ y^{(m)} \end{bmatrix} \tag{2}$$

The linear function $\mathbf{h}$ is

$$\mathbf{h}_\theta(\mathbf{x}^{(i)}) = \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + ... + \theta_n x_n^{(i)} \tag{3}$$

or

$$\mathbf{h}_\theta(\mathbf{x}^{(i)}) = \sum_{j=0}^{n} \theta_j x_j^{(i)} \tag{4}$$

where $\theta_i \in \mathbb{R}$ and i = 1 , . . . , m, such that

$$\mathbf{J}(\theta_0, ..., \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (\mathbf{h}_\theta(\mathbf{x}^{(i)}) - y^{(i)})^2 \tag{5}$$

is minimized.

2.1 Additional Note

Taking θ as a vector,

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ . \\ . \\ . \\ \theta_n \end{bmatrix} \tag{6}$$

we can rewrite **(3)** or **(4)** as

$$\mathbf{h}_\theta(\mathbf{x}^{(i)}) = \boldsymbol{\theta}^T \mathbf{x}^{(i)} \tag{7}$$

Since

$$\mathbf{X}\boldsymbol{\theta} = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} & . & . & . & x_n^{(1)} \\ x_0^{(2)} & x_1^{(2)} & . & . & . & x_n^{(2)} \\ . & . & . & & & . \\ . & . & & . & & . \\ . & . & & & . & . \\ x_0^{(m)} & x_1^{(m)} & . & . & . & x_n^{(m)} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ . \\ . \\ . \\ \theta_n \end{bmatrix} = \sum_{i=1}^{m} \mathbf{h}_\theta(\mathbf{x}^{(i)}) \quad , \tag{8}$$

then, **(5)** can be rewritten as (Note that $\mathbf{X}\boldsymbol{\theta}$ is a vector.)

$$\mathbf{J}(\boldsymbol{\theta}) = \frac{1}{2m}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}) \tag{9}$$

2.2 Gradient Descent

Initialize $\boldsymbol{\theta}$ with randomly generated numbers once and repeatedly perform the following update until a stopping criteria is met to minimize $\mathbf{J}$:

$$\theta_j := \theta_j - \sigma(\frac{\partial}{\partial\theta_j}\mathbf{J}(\theta)) \tag{10}$$

where j = 0 , . . . , n and $\sigma \in \mathbb{R}^+$. Usually σ will be between 0 and 1.

Since,

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}\mathbf{J}(\theta) &= \frac{\partial}{\partial(\theta_j)}\frac{1}{2m}\sum_{i=1}^{m}(\mathbf{h}_\theta(\mathbf{x}^{(i)}) - y^{(i)})^2 \\
&= 2.\frac{1}{2m}\sum_{i=1}^{m}(\mathbf{h}_\theta(\mathbf{x}^{(i)}) - y^{(i)})\frac{\partial}{\partial(\theta_j)}(\mathbf{h}_\theta(\mathbf{x}^{(i)}) - y^{(i)}) \\
&= \frac{1}{m}\sum_{i=1}^{m}(\mathbf{h}_\theta(\mathbf{x}^{(i)}) - y^{(i)})\frac{\partial}{\partial(\theta_j)}(\sum_{j=0}^{n}\theta_j x_j^{(i)} - y^{(i)}) \\
&= \frac{1}{m}\sum_{i=1}^{m}(\mathbf{h}_\theta(\mathbf{x}^{(i)}) - y^{(i)})x_j^{(i)}
\end{aligned}$$

then,

$$\frac{\partial}{\partial\theta_j}\mathbf{J}(\theta) = \frac{1}{m}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})\mathbf{x}_j \tag{11}$$

References

[1] Andrew Ng, *CS229 Lecture notes.*

[2] *Linear Regression. (2019). Retrieved from http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm*