A DETAILED DERIVATION OF LINEAR REGRESSION

PERSONAL NOTES ON MACHINE LEARNING

Junaid H. Rahim
School of Computer Engineering, KIIT University
junaidrahim5a@gmail.com

August 1, 2020

ABSTRACT

The following is a detailed derivation of the linear regression algorithm usually used to model linear functions. In Linear Regression, a linear combination of the input vector and a weight vector is taken and then a bias element is added. These weights are then used to predict a real value from an input vector. For training the model, we use gradient descent to find the weight vector as such that a specified loss function is at its minimum. The main objective of this document is to clearly describe the mathematics behind linear regression.

Keywords Logistic Regression · Machine Learning

1 Derivation

Given that we have m training examples with n features each

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_m\} \quad \forall\, \mathbf{x}_i \in \mathbb{R}^n$$
$$\mathbf{y} = \{y_1, y_2, y_3, ..., y_m\} \quad \forall\;\, y_i \in \mathbb{R}$$

We can think of X as a $\mathbb{R}^{n\times m}$ matrix and Y as a $\mathbb{R}^m$ vector holding the predicted values for each training sample. Now we define a weight vector $\mathbf{w} \in \mathbb{R}^n$ and a scalar value $\mathbf{b} \in \mathbb{R}^m$ which is also known as the *bias* vector

The predicted value $\hat{\mathbf{y}} \in \mathbb{R}$ is specified as

$$\hat{\mathbf{y}} = \mathbf{w}^T X + \mathbf{b}$$

The loss function for a single training sample is expressed as,

$$Loss(y, \hat{y}) = (\hat{y} - y)^2$$

For the entire training set,

$$Cost \;=\; J(\mathbf{w}, \mathbf{b}) = \frac{1}{m}\sum_{i=1}^{m} Loss(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)})$$

$$J(\mathbf{w}, \mathbf{b}) = \frac{1}{2m}\sum_{i=1}^{m} (\hat{\mathbf{y}}^{(i)} - \mathbf{y}^{(i)})^2$$

Substituting the value of $\hat{\mathbf{y}}$

$$J(\mathbf{w}, \mathbf{b}) = \frac{1}{2m} \sum_{i=1}^{m} (\mathbf{w}^T X^{(i)} + \mathbf{b} - \mathbf{y}^{(i)})^2$$

Therefore the derivative,

$$\frac{\partial J(\mathbf{w}, \mathbf{b})}{\partial \mathbf{w}} = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w}^T X^{(i)} + \mathbf{b} - \mathbf{y}^{(i)})\ X^{(i)}$$

$$\frac{\partial J(\mathbf{w}, \mathbf{b})}{\partial \mathbf{w}} = \frac{1}{m} \sum_{i=1}^{m} (\hat{\mathbf{y}}^{(i)} - \mathbf{y}^{(i)})\ X^{(i)}$$

$$\frac{\partial J(\mathbf{w}, \mathbf{b})}{\partial \mathbf{b}} = \frac{1}{m} \sum_{i=1}^{m} (\hat{\mathbf{y}}^{(i)} - \mathbf{y}^{(i)})$$

The Gradient Descent optimization step now is,

$$\mathbf{w} := \mathbf{w} - \alpha \frac{\partial J}{\partial \mathbf{w}}$$

$$\mathbf{b} := \mathbf{b} - \alpha \frac{\partial J}{\partial \mathbf{b}}$$

Where α is the learning rate usually set to 10^{-3}

2 Conclusion

These formulae precisely explain the entire linear regression algorithm.