A DETAILED DERIVATION OF LOGISTIC 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 Logistic Regression. It's simple algorithm that can be used for binary classification tasks. The algorithm on its own is pretty simple, we take a linear combination of the input features with a weight vector and pass that into a sigmoid function. As the sigmoid function squishes values passed between 0 and 1, we use that as a measure of the probability of truthness $y = 1$. 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 logistic regression.

Keywords Logistic Regression · Machine Learning

1 Derivation

Given that we have m training examples with n features each

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_m\} \quad \forall\, \mathbf{x}_i \in \mathbb{R}^n$$
$$\mathbf{y} = \{y_1, y_2, y_3, ..., y_m\} \quad \forall\;\; y_i \in \{0, 1\}$$

We can think of X as a $\mathbb{R}^{n\times m}$ matrix and Y as a $\mathbb{R}^m$ vector holding the ground truths 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 sigmoid function is defined as

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

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

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

Here the sigmoid function is applied to all the elements of the vector, thus we get the m sized vector $\hat{\mathbf{y}} \quad \forall\, \hat{y}_i \in [0, 1]$

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

$$Loss(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$$

For the entire training set,

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

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

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

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

The derivative $\sigma'(z) = \sigma(z).(\sigma(z) - 1)$. Therefore the derivative

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

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

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

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

Similarly, we can calculate, as the derivative of $\mathbf{b}$ is 1

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

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 logistic regression algorithm.