Logistic Regression

Classification Algorithm

Introduction

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud, Tumor Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as the '**Sigmoid function**' or also known as the 'logistic function' instead of a linear function.

Sigmoid function range = (0 to 1)

Sigmoid function :

In order to maps the predicted values to probability , we use the sigmoid function.

$$f(x) = \frac{1}{1 + e^{-(x)}}$$

Representation :

Linear equation : $Z = \beta_0 + \beta_1 X$

Apply the sigmoid function

$h\Theta(x) = \text{sigmoid}(Z)$

$h\Theta(x) = 1/(1 + e\text{^}-(\beta_0 + \beta_1 X)$

Cost Function:

We learnt about the cost function $J(\vartheta)$ in the *Linear regression*, the cost function represents optimization objective i.e. we create a cost function and minimize it so that we can develop an accurate model with minimum error.

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

Cost function of linear regression

For logistic regression , the Cost function is defined as:

*−log(*hθ*(*x*)) if y = 1*

*−log(1−*hθ*(*x*)) if y = 0*

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Cost function of logistic regression

The above two functions can be compressed into a single function i.e.

$$J(\theta) = -\frac{1}{m}\sum\left[y^{(i)}\log(h\theta(x(i))) + \left(1 - y^{(i)}\right)\log(1 - h\theta(x(i)))\right]$$

Gradient Descent:

Now the question arises, how do we reduce the cost value. Well, this can be done by using **Gradient Descent.** The main goal of Gradient descent is to **minimize the cost value.** i.e. min J(**ϑ**).

Now to minimize the cost function we need to run gradient descent function on each parameter.

$$\theta j := \theta j - \alpha \frac{\partial}{\partial \theta j} J(\theta)$$

Want $\min_\theta J(\theta)$:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update all θ_j)

$\}$

Thank You