Cost Function for Logistic Regression

luel

Contents

1 Introduction

Machine Learning models require a high level of accuracy to work in the actual world. But how do you calculate how wrong or right your model is? This is where the cost function comes into the picture. A machine learning parameter that is used for correctly judging the model, cost functions are important to understand to know how well the model has estimated the relationship between your input and output parameters.

1.1 Cost function

After training your model, you need to see how well your model is performing. While accuracy functions tell you how well the model is performing, they do not provide you with an insight on how to better improve them. Hence, you need a correctional function that can help you compute when the model is the most accurate, as you need to hit that small spot between an undertrained model and an overtrained model.

A Cost Function is used to measure just how wrong the model is in finding a relation between the input and output. It tells you how badly your model is behaving/predicting

1.2 Logistic regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

1.2.1 Types of logistic regression

- Binary (*Pass/Fail*)
- Multi (*Cats, Dogs, Sheep*)
- Ordinal (*Low, Medium, High*)

2 The Derivative of Cost Function

The gradient of the cost function of linear regression has a very simplified form given below. The gradient for the loss function of logistic regression also comes out to have the same form of terms in spite of having a complex log loss error function.

$$\begin{bmatrix} \frac{\partial J(\theta)}{\partial \theta_0} \\ \frac{\partial J(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial J(\theta)}{\partial \theta_n} \end{bmatrix} = \frac{1}{m} x^T (h(x) - y)$$

$Gradiant for Linear Regression Loss Function$

In order to preserve the convex nature for the loss function, a log loss error function has been designed for logistic regression. The cost function is split for two cases y=1 and y=0.

For the case when we have y=1 we can observe that when hypothesis function tends to 1 the error is minimized to zero and when it tends to 0 the error is maximum. This criterion exactly follows the criterion as we wanted.

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

In order to optimize this convex function, we can either go with gradient-descent or newtons method. For both cases, we need to derive the gradient of this complex loss function. The mathematics for deriving gradient is shown in the steps given below.

From equation (1) on page 4, combining both the equation we get a convex log loss function as shown below.

$$cost(h_\theta(x),\ y) = -y\log(h_\theta(x)) - (1-y)\log(1 - h_\theta(x))$$
$$if\ y = 1:\ cost(h_\theta(x),\ y) = -\log(h_\theta(x))$$
$$if\ =:\ cost(h_\theta(x),\ y) = -\log(1 - h_\theta(x))$$

$$Combinedcostfunction$$

Since the hypothesis function for logistic regression is sigmoid in nature hence, The First important step is finding the gradient of the sigmoid function. We can see from the derivation below that gradient of the sigmoid function follows a certain pattern.

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$
$$HypothesisFunction$$

$$\sigma(x) = \tfrac{1}{1+e^{-x}}$$

$$\tfrac{d(\sigma(x))}{dx} = \tfrac{0(1+e^{-x})-(1)(e^{-x}(-1))}{(1+e^{-x})^2}$$

$$\tfrac{d(\sigma(x))}{dx} = \tfrac{e^{-x}}{(1+e^{-x})^2}$$

$$\tfrac{d(\sigma(x))}{dx} = \tfrac{1-1+e^{-x}}{(1+e^{-x})^2}$$

$$\tfrac{d(\sigma(x))}{dx} = \tfrac{1+e^{-x}}{(1+e^{-x})^2} - \tfrac{1}{(1+e^{-x})^2}$$

$$\tfrac{d(\sigma(x))}{dx} = \tfrac{1}{1+e^{-x}}(1 - \tfrac{1}{1+e^{-x}})$$

Step 1:

Applying Chain rule and writing in terms of partial derivatives.

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{-1}{m} * \sum_{i=1}^{m}[y^{(i)} * \frac{1}{h_\theta(x^{(i)})} * \frac{\partial(h_\theta(x^{(i)}))}{\partial(\theta_j)}] + \sum_{i=1}^{m}[(1-y^{(i)}) * \frac{1}{(1-h_\theta(x^{(i)}))} * \frac{\partial(1-h_\theta(x^{(i)}))}{\partial(\theta_j)}]$$

$$\begin{aligned}\frac{\partial J(\theta)}{\partial \theta_j} = \frac{-1}{m} * (&\sum_{i=1}^{m}[y^{(i)} * \frac{1}{h_\theta(x^{(i)})} * \sigma(z)(1-\sigma(z)) * \frac{\partial(\theta^{Tx})}{\partial(\theta_j)}] \quad + \\ &\sum_{i=1}^{m}[(1-y^{(i)}) * \frac{1}{(1-h_\theta(x^{(i)}))} * (-\sigma(z)(1-\sigma(z))) * \frac{\partial(\theta^{Tx})}{\partial(\theta_j)}])\end{aligned}$$

Step 2:

Evaluating the partial derivative using the pattern of the derivative of the sigmoid function.

$$\begin{aligned}\frac{\partial J(\theta)}{\partial \theta_j} = \frac{-1}{m} * (&\sum_{i=1}^{m}[y^{(i)} * \frac{1}{h_\theta(x^{(i)})} * \sigma(z)(1-\sigma(z)) * \frac{\partial(\theta^{Tx})}{\partial(\theta_j)}] \quad + \\ &\sum_{i=1}^{m}[(1-y^{(i)}) * \frac{1}{(1-h_\theta(x^{(i)}))} * (-\sigma(z)(1-\sigma(z))) * \frac{\partial(\theta^{Tx})}{\partial(\theta_j)}])\end{aligned}$$

$$\begin{aligned}\frac{\partial J(\theta)}{\partial \theta_j} = \frac{-1}{m} * (&\sum_{i=1}^{m}[y^{(i)} * \frac{1}{h_\theta(x^{(i)})} h_\theta(x^{(i)})(1-h_\theta(x^{(i)})) * x_j^i] \quad + \\ &\sum_{i=1}^{m}[(1-y^{(i)}) * \frac{1}{(1-h_\theta(x^{(i)}))} * (-h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))) * x_j^i])\end{aligned}$$

Step 3:

Simplifying the terms by multiplication.

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{-1}{m} * (\sum_{i=1}^{m}[y^{(i)} * (1-h_\theta(x^{(i)}) * x_j^i - (1-y^{(i)}) * h_\theta(x^{(i)}) * x_j^i]$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{-1}{m} * (\sum_{i=1}^{m}[y^{(i)} - y^{(i)} * h_\theta(x^{(i)}) - h_\theta(x^{(i)}) + y^{(i)} * h_\theta(x^{(i)})] * x_j^i)$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{-1}{m} * (\sum_{i=1}^{m}[y^{(i)} - h_\theta(x^{(i)})] * x_j^i)$$

Step 4:

Removing the summation term by converting it into a matrix form for the gradient with respect to all the weights including the bias term.

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{m} X^T[h_\theta(x) - y]$$

3 Conclusion

This is the basic concept about Cost Function, Logistic Regression, and the cost function for the logistic regression.

References

[1] Logistic Regression

[2] What is Cost Function in Machine Learning

[3] The Derivative of Cost Function for Logistic Regression