Logistic Regression Notes

by Jansen Ken Pegrasio

23 February 2025

1 Introduction

Similarly to linear regression, logistic regression works by assuming a linear relationship between features and targets. However, linear and logistic regression have different purposes. Instead of using it to predict continuous values, logistic regression is used for classification.

2 Sigmoid Function

A common question is "Why don't we just use linear regression for classification?" The answer is because the y-axis in linear regression can range from $-\infty$ to ∞, whereas in logistic regression, we want the y-axis to range from 0 to 1, to indicate probability, assume it p.

Notice how the log of the odds also ranges from $-\infty$ to ∞. Instead of setting the probability as the y-axis, let us use the log of the odds. Now, similarly to linear regression, we will predict the y-values. Then, because we need the information about the probability, we need to convert the log of the odds back.

$$\begin{aligned}
\log(odds) &= \log\left(\frac{p}{1-p}\right) \\
e^{\log(odds)} &= \frac{p}{1-p} \\
(1-p)\cdot e^{\log(odds)} &= p \\
e^{\log(odds)} - p\cdot e^{\log(odds)} &= p \\
e^{\log(odds)} &= p + p\cdot e^{\log(odds)} \\
e^{\log(odds)} &= p\cdot(1+e^{\log(odds)}) \\
p &= \frac{e^{\log(odds)}}{1+e^{\log(odds)}}
\end{aligned}$$

In other scenarios, you may see other form of this conversion.

$$p = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} \rightarrow p = \frac{1}{\frac{1+e^{\log(odds)}}{e^{\log(odds)}}} \rightarrow p = \frac{1}{\frac{1}{e^{\log(odds)}} + 1} \rightarrow p = \frac{1}{1 + e^{-\log(odds)}}$$

This function is usually called as the sigmoid function, $\sigma(z) = \frac{1}{1+e^{-z}}$.

3 Loss Function

In logistic regression, the loss function that we are using is called the binary cross entropy. Before diving further, assume N is the number of samples, y_i is the actual prediction for every sample i from 1 to N, and y_{p_i} is the predicted value when given the features for every sample i from 1 to N.

Given that $y_{p_i} = \sigma(Xw)$ where X is the $n \times (m+1)$ design matrix, w is the weight vector – $(m+1) \times 1$ matrix. The binary cross entropy loss function can be formulated as

$$BCE(w) = -\frac{1}{N} \cdot \sum_{i=1}^{N} (y_i \cdot \log(y_{p_i}) + (1 - y_i) \cdot \log(1 - y_{p_i}))$$

4 Derivatives of Loss Function

Now, let's take the derivatives of the loss function both with respect to the weight and bias. We will use it for gradient descent. First, let's find the derivatives with respect to the weight!

$$\frac{\partial BCE(w)}{\partial w} = \frac{\partial BCE(w)}{\partial y_{p_i}} \cdot \frac{\partial y_{p_i}}{\partial (Xw)} \cdot \frac{\partial (Xw)}{\partial w}$$

Let's break it one by one!

1. For the first part:

$$\begin{aligned}
&\frac{\partial}{\partial y_{p_i}}(-\frac{1}{N} \cdot \sum_{i=1}^{N} (y_i \cdot \log(y_{p_i}) + (1 - y_i) \cdot \log(1 - y_{p_i}))) \\
&= -\frac{1}{N} \cdot \sum_{i=1}^{N} \frac{\partial}{\partial y_{p_i}} (y_i \cdot \log(y_{p_i}) + (1 - y_i) \cdot \log(1 - y_{p_i})) \\
&= -\frac{1}{N} \cdot \sum_{i=1}^{N} (\frac{y_i}{y_{p_i}} - \frac{1 - y_i}{1 - y_{p_i}})
\end{aligned}$$

2. For the second part:

$$\begin{aligned}
\frac{\partial y_{p_i}}{\partial (Xw)} &= \frac{\partial \sigma(Xw)}{\partial (Xw)} = \frac{\partial}{\partial (Xw)} \frac{1}{1+e^{-Xw}} \\
&= \frac{\partial}{\partial (1+e^{-Xw})} \frac{1}{1+e^{-Xw}} \cdot \frac{\partial (1+e^{-Xw})}{\partial (Xw)} \\
&= \frac{\partial (1+e^{-Xw})^{-1}}{\partial (1+e^{-Xw})} \cdot \frac{\partial (1+e^{-Xw})}{\partial (Xw)} \\
&= -(1+e^{-Xw})^{-2} \cdot (-e^{-Xw}) \\
&= \frac{e^{Xw}}{(1+e^{-Xw})^2} = \frac{1}{1+e^{-Xw}} \cdot \frac{e^{-Xw}}{1+e^{-Xw}} \\
&= \frac{1}{1+e^{-Xw}} \cdot (1 - \frac{1}{1+e^{-Xw}}) \\
&= y_{p_i} \cdot (1 - y_{p_i})
\end{aligned}$$

3. For the third part:

$$\frac{\partial (Xw)}{\partial w} = X^T$$

Combining these together, we will have:

$$\begin{aligned}
&\frac{\partial BCE(w)}{\partial w} = -\frac{1}{N} \cdot \sum_{i=1}^{N} (\frac{y_i}{y_{p_i}} - \frac{1-y_i}{1-y_{p_i}}) \cdot y_{p_i} \cdot (1-y_{p_i}) \cdot X^T \\
&= -\frac{1}{N} \cdot \sum_{i=1}^{N} (\frac{y_i \cdot (1-y_{p_i}) - (1-y_i) \cdot y_{p_i}}{y_{p_i} \cdot (1-y_{p_i})}) \cdot y_{p_i} \cdot (1-y_{p_i}) \cdot X^T \\
&= -\frac{1}{N} \cdot \sum_{i=1}^{N} (y_i - y_i \cdot y_{p_i} - y_{p_i} + y_i \cdot y_{p_i}) \cdot X^T \\
&= -\frac{1}{N} \cdot \sum_{i=1}^{N} (y_i - y_{p_i}) \cdot X^T = -\frac{1}{N} \cdot (Y - Y_p) \cdot X^T
\end{aligned}$$

Now, let's analyze the dimension of each matrix. $(Y - Y_p)$ is a $n \times 1$ matrix, X^T is a $(m+1) \times n$ matrix. Recap that we expect $(m+1) \times 1$ matrix for our gradient descent. Thus, we need to rearrange the matrix equation to be $-\frac{1}{N} \cdot X^T \cdot (Y - Y_p) = \frac{1}{N} \cdot X^T \cdot (Y_p - Y)$.

5 Implementation Details

When implementing logistic regression, we can choose to separate weight and intercept to two different variables: *intercept* and *weight*. *intercept* will just be filled by w_0, whereas *weight* is a column vector corresponds to $w_1, w_2, ..., w_m$.

To calculate the rate of change of w_0, we can use the formula $\frac{1}{N}X^T(Y_p - Y)$. Then, because X^T is all ones, we can simplify the formula to be $\frac{1}{N}(Y_p - Y)$. Then, to calculate the rate of change of w_i, we can use the formula: $\frac{1}{N}X^T(Y_p - Y)$.