Logistic Regression
Epoch IIT Hyderabad

Ankit Saha
AI21BTECH11004

15 Jul 2022

1. Introduction

Logistic regression is a supervised machine learning algorithm that predicts categorical labels for a given input. Estimating real-valued output is known as regression and estimating categorical output is known as classification. It first uses regression to obtain a confidence factor for each input and then uses a decision rule to classify that input with a probability associated with it equal to the confidence factor.

2. Mathematical Formulation

Suppose we are given N labelled samples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$ where $x_i \in \mathbb{R}^n, y_i \in \{0, 1\} \ \forall i \in [N]$

We choose the sigmoid function to model our probability. Any function that takes values between 0 and 1 can be used, but the sigmoid function is one of the most popular ones.

$$S(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

$$1 - S(x) = 1 - \frac{1}{1 + e^{-x}} \tag{2.2}$$

$$= \frac{1 + e^{-x} - 1}{1 + e^{-x}} \tag{2.3}$$

$$= \frac{e^{-x}}{1 + e^{-x}} \tag{2.4}$$

$$= \frac{1}{1 + e^{x}} \tag{2.5}$$

$$= S(-x) \tag{2.6}$$

Our goal is to find a weight vector $\mathbf{w} \in \mathbb{R}^n$ that maximizes the log-likelihood J

$$L = \prod_{y_i=1} S(z_i) \prod_{y_i=0} (1 - S(z_i)) \tag{2.7}$$

$$J(\mathbf{w}) = \frac{1}{N} \log L = \frac{1}{N} \left(\sum_{y_i=1} \log S(z_i) + \sum_{y_i=0} \log(1 - S(z_i)) \right) \tag{2.8}$$

$$= \frac{1}{N} \sum_{i=1}^{N} y_i \log(S(z_i)) + (1 - y_i) \log(1 - S(z_i)) \tag{2.9}$$

$$= \frac{1}{N} \sum_{i=1}^{N} y_i \log \left(\frac{1}{1 + e^{-z_i}} \right) + (1 - y_i) \log \left(\frac{1}{1 + e^{z_i}} \right) \tag{2.10}$$

$$= -\frac{1}{N} \sum_{i=1}^{N} y_i \log \left(1 + e^{-z_i}\right) + (1 - y_i) \log \left(1 + e^{z_i}\right) \tag{2.11}$$

where $z_i = \mathbf{w}^\top \mathbf{x}_i + b$ where b is the bias term. $S(z_i)$ is the confidence value of our predicted output for input i.

3. Finding the Optimal Weights

The objective is to maximize the log-likelihood function $J(\mathbf{w})$. In other words, we have to minimize $-J(\mathbf{w}) = C$, which is going to be our cost function. This can be done using the steepest descent algorithm which is as follows:

i) Initialize $\mathbf{w} = \mathbf{w}^{(0)}$
ii) Compute the gradient $g^{(k)} = \nabla C(\mathbf{w}^{(k)})$
iii) At the next iteration, set $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - t_k g^{(k)}$ where t_k is the step size
iv) Keep iterating over k until $\nabla C(\mathbf{w}^{(k)})$ is sufficiently small

The step size t_k can be either constant or varying depending on the model.

This algorithm works because the direction of steepest descent of a function is along its gradient, i.e., if we go along the gradient, we will observe the maximum possible decrease in a function. This algorithm is repeatedly taking small steps in the direction of the gradient at each iteration until it reaches a point where there is hardly any change in the function, i.e., the gradient is very small (nearly zero), i.e., the function has reached a minima.

4. Classification

Once the optimal weight vector has been obtained, we can finally assign a label to an input based on a decision rule after computing $S(z_i) = S(\mathbf{w}^\top \mathbf{x}_i + b)$

$$\hat{y}_i = \begin{cases} 1 & S(z_i) \geq 0.5 \\ 0 & S(z_i) < 0.5 \end{cases} \tag{4.1}$$

This classification is with a probability (confidence factor) of $S(z_i)$, i.e., the closer $S(z_i)$ is to 1, the more likely it is to have a label of 1 and vice versa.

5. Regularization

One of the major drawbacks of logistic regression is that it can't be used for linearly separable data because the weights approach infinity. In order to avoid this, we can add a regularizer term $\lambda \|\mathbf{w}\|^2$ to our cost function.

$$C = \frac{1}{N} \sum_{i=1}^{N} y_i \log(1 + e^{-z_i}) + (1 - y_i) \log(1 + e^{z_i}) + \lambda \|\mathbf{w}\|^2 \tag{5.1}$$

This helps control the weight vector because here we are minimizing the sum of the negative log-likelihood and the norm of the weight vector multiplied by an arbitrary scale factor λ. This will ensure that the norm of the weight vector does not become very large.

6. Questions

i) What is the significance of the bias term b?
ii) Does $S(z)$ ever become equal to 0 or 1?
iii) Calculate z for which $S(z) = 0.5$
iv) In the steepest descent algorithm, what happens when the step size is constant?
v) Can logistic regression be used for multi-class classification?

7. Answers

i) The bias term b represents the y-intercept. It adds flexibility to the model by not constraining it to pass through the origin.
ii) The sigmoid function never becomes exactly equal to 0 or 1. It only approaches them at negative and positive infinity respectively.

$$\lim_{z\to-\infty} \frac{1}{1+e^{-z}} = 0 \tag{7.1}$$

$$\lim_{z\to\infty} \frac{1}{1+e^{-z}} = 1 \tag{7.2}$$

iii) $S(z) = 0.5$

$$\implies \frac{1}{1+e^{-z}} = \frac{1}{2} \tag{7.3}$$

$$\implies 1 + e^{-z} = 2 \tag{7.4}$$

$$\implies e^{-z} = 1 \tag{7.5}$$

$$\implies z = 0 \tag{7.6}$$

iv) When the step size is constant, at some point, the gradient just keeps oscillating about the minima but doesn't apporach the minima.
v) Yes, logistic regression can be used for multi-class classification too because multi-class classification can be thought of as a series of binary classifications. This is done by forming code matrices for several possible methods: data replication for ordinal labels, one vs rest and one vs one for cardinal labels.