Logistic regression analysis

Enzhi Li

August 10, 2017

1 Introduction

Logistic regression can be used to cope with the 0 -1 problem in data analysis. Take the data from wikipedia as the example. Below is the result for an exam. Denote h as the hours that a student spends on preparing for an exam, and p as the result of the exam. p is a binary result that can only take the value 0 which means fail, and 1 which means pass. The result is tabulated below.

hours	0.5	0.75	1.00	1.25	1.5	1.75	1.75	2.00	2.25	2.50
pass	0	0	0	0	0	0	1	0	1	0

hours	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
pass	1	0	1	0	1	1	1	1	1	1

2 Maximum likelihood function

We want to use the above result to predict if a student will be table to pass the exam or not given the number of hours that the student spent on the exam preparation. In order to do this, we invoke the logistic model, which states that the probability of passing the exam given h is

$$p = g(\theta^T X) =: \frac{1}{1 + e^{\theta^T X}}, \tag{1}$$

where we have used the notation $g(\alpha) = \frac{1}{1+e^{\alpha}}$, $\theta = (\theta_0, \theta_1)^T, X = (1, h)^T$. We now try to estimate the parameters θ_0, θ_1 based upon the original data, using the maximum likelihood assumption. We first define the likelihood function as

$$L(\theta) = \prod_{i=1}^{n} g(\theta^T X^{(i)})^{y^{(i)}} (1 - g(\theta^T X^{(i)}))^{1-y^{(i)}}. \tag{2}$$

Here, $y^{(i)} = 0, \text{or } 1$ is the exam result. Our goal is find the parameters θ that can maximize this function. It proves to be more convenient to solve an equivalent problem, that is, to minimize the minus log likelihood function which is defined as

$$\begin{aligned} \Lambda(\theta) &= -\sum_{i=1}^{n} y^{(i)} \log g(\theta^T X^{(i)}) + (1 - y^{(i)}) \log(1 - g(\theta^T X^{(i)})) \\ &= \sum_{i=1}^{n} y^{(i)} \log(1 + e^{\theta^T X^{(i)}}) + (1 - y^{(i)}) \log(1 + e^{-\theta^T X^{(i)}}) \end{aligned} \tag{3}$$

3 An excursion to a simpler function

Before we start the minimization of the negative log likelihood function, we first study a simpler version of the negative log likelihood function which is defined as

$$f(\alpha) = y \log(1 + e^{\alpha}) + (1 - y) \log(1 + e^{-\alpha}), \alpha = \theta^T X \tag{4}$$

Partial differentiation yields

$$\begin{aligned} \frac{\partial f(\alpha)}{\partial \theta_\mu} &= \frac{df(\alpha)}{d\alpha} \frac{\partial \alpha}{\partial \theta_\mu} = f'(\alpha) X_\mu \\ \frac{\partial^2 f(\alpha)}{\partial \theta_\mu \partial \theta_\nu} &= \frac{\partial}{\partial \theta_\nu} \Big(\frac{df(\alpha)}{d\alpha} \Big) X_\mu = f''(\alpha) X_\mu X_\nu \end{aligned} \tag{5}$$

The Hessian matrix of $f(\alpha)$ is thus

$$H = f''(\alpha) X X^T \tag{6}$$

The eigenvalues of matrix XX^T, where $X = (1, x)^T$, can be easily calculated to be $0, 1 + x^2$. We see immediately that the Hessian matrix is singular.

We also have

$$\begin{aligned} f'(\alpha) &= \frac{y}{1 + e^{-\alpha}} - \frac{1 - y}{1 + e^{\alpha}}, \\ f''(\alpha) &= \frac{1}{4 \cosh^2 \frac{\alpha}{2}} > 0 \end{aligned} \tag{7}$$

Therefore, we see that the Hessian matrix of this function is positive semi-definite, which enables us to employ the gradient descent method to find the minimum value of this function.

4 Back to the original negative log likelihood function $\Lambda(\theta)$

The gradient of $\Lambda(\theta)$ with respect to θ is

$$\frac{\partial \Lambda(\theta)}{\partial \theta_\mu} = \sum_{i=1}^{n} \left(\frac{y^{(i)}}{1 + e^{-\theta^T X^{(i)}}} - \frac{1 - y^{(i)}}{1 + e^{\theta^T X^{(i)}}} \right) X_\mu^{(i)} \tag{8}$$

The Hessian matrix of $\Lambda(\theta)$ is

$$\begin{aligned} H_{\mu\nu} &= \frac{\partial^2 \Lambda(\theta)}{\partial \theta_\mu \partial \theta_\nu} \\ &= \sum_{i=1}^{n} \frac{1}{4 \cosh^2 \frac{\theta^T X^{(i)}}{2}} X_\mu^{(i)} X_\nu^{(i)} \end{aligned} \tag{9}$$

We have seen in the previous section that the Hessian matrix for function $f(\alpha)$ is singular. However, here, the Hessian matrix, which is a sum of many singular matrices, is not necessarily singular in itself. If the Hessian matrix is not singular, then we can use Newton's method instead of the much slower gradient descent method to find the minimum value. Newton's method is defined as

$$\theta_{n+1} = \theta_n - H^{-1} J, \tag{10}$$

where H is the Hessian matrix, and J is the gradient. We can start with some initial θ_0 from guesswork, and recursively calculate the updated θ values using the Newton's method, and exit the program once two consecutive $\theta' s$ differ from each other negligibly. The simpler and slower gradient descent method can be seen as a naive implementation of the Newton's method where the often complicated Hessian matrix is replaced with a much simpler diagonal matrix with all the diagonal elements being the same.

However, in practice, the Newton's method is not always superior to the gradient descent method, with the reason being that the Hessian matrix is generally not well behaved during the iteration process. In this case, we can replace the singular Hessian matrix with a scalar matrix, which means a matrix that is generated from the identity matrix multiplied with a

constant. With this substitution of the Hessian matrix, the Newton's method can be recast into this much simpler form, which is

$$\theta_{n+1} = \theta_n - r * J. \tag{11}$$

Here, r is a number, which we call the learning rate. This method is called the gradient descent method, and it is simpler to implement than Newton's method due to the elimination of the Hessian matrix and its inverse from this method. The calculation of the Hessian matrix and its inverse are pretty time consuming. From the gradient descent, we can obtain the θ that can minimize the negative log likelihood function, and obtain a logistic curve that can be compared to the experimental results.