Multi-class classification Regression Model

Dihui Lai

November 1, 2020

Contents

1 Likelihood function of Multinomial distribution

1.1 Multinomial Distribution

The multinomial distribution has density function

$$f(y_1, y_2, y_3, ...y_c) = \frac{N!}{y_1!y_2!...y_c!}p_1^{y_1}p_2^{y_2}...p_c^{y_c}$$

where c is the number of classes that is observed in the dataset. For example, if you have 3 types of outcomes then c=3. N is the total number of observations that we have in the dataset. y_j, $j = 1, 2, ..., c$ is the number of observations that belong to the the jth categories. Accordingly $\sum\limits_{j=1}^{c} y_j = N$

When we perform ith experiemnt we have $N = 1$, the likelihood function of the observation is accordingly, $L_i = \prod\limits_{j=1}^{c} p_j^{y_j^i}$, note that only one of the y_j^i will be 1 the rest will all be 0, as $\sum\limits_{j=1}^{c} y_j^i = 1$

The likelihood function for a dataset of size N is then $L = \prod_{i=1}^{N} \prod_{j=1}^{c} p_j^{y_j^i}$. Note y_j^i is either 1 or 0.

The log-likelihood function is

$$\ell = \sum_{i=1}^{n} \sum_{j=1}^{c} y_j^i \log(p_j) \tag{1}$$

The reverse of the log-likelihood is also known as the log-loss function

$$logloss = \log(L) = -\sum_{i=1}^{n} \sum_{j=1}^{c} y_j^i \log(p_j)$$

Reference: `https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html`

1.2 Maximum Likelihood Estimation of p_j

The p_k that maximize the log-likelihood function that subject to the constraint $\sum_{j=1}^{c} p_j = 1$ has to satisfy the following condition (using method of Lagrange multipliers)

$$\begin{cases} \frac{\partial}{\partial p_k} \left(\ell - \lambda \sum_{i=1}^{n} (1 - \sum_{j=1}^{c} p_j) \right) = 0 \\ \frac{\partial}{\partial \lambda} \left(\ell - \lambda \sum_{i=1}^{n} (1 - \sum_{j=1}^{c} p_j) \right) = 0 \end{cases}$$

$$\Rightarrow \begin{cases} \sum_{i=1}^{n} \frac{\partial}{\partial p_k} \left(\sum_{j=1}^{c} x_j^i \log(p_j) - \lambda (1 - \sum_{j=1}^{c} p_j) \right) = 0 \\ \sum_{i=1}^{n} \frac{\partial}{\partial \lambda} \left(\sum_{j=1}^{c} x_j^i \log(p_j) - \lambda (1 - \sum_{j=1}^{c} p_j) \right) = 0 \end{cases}$$

$$\Rightarrow \lambda = -\frac{y_k}{p_k}$$

where x_k is the total number of outcome that belong to category k. Because

$$\sum_{k=1}^{c} y_k = N$$

We have

$$-\sum_{k=1}^{c} \lambda p_k = N \Rightarrow \lambda = -N$$

Therefore

$$p_k = \frac{y_k}{N}$$

1.3 Modeling p_j and Softmax

A reasonable modeling methods for p_j in equation (1) is to use a softmax transformation of the lienar core $\vec{\beta} \cdot \vec{x}$, where $\vec{x}$ is the vector composed of the predictors.

For a given data point i, the probability that the outcome being j, $j = 1, 2, 3, ..., c$ is

$$p_j = \frac{\exp(\vec{\beta_j} \cdot \vec{x}^i)}{\sum_{j=1}^{c} \exp(\vec{\beta_j} \cdot \vec{x}^i)} \tag{2}$$

Note that we have total c vector βs, where each $\vec{\beta_j}$, $j = 1, 2, 3, ...c$ belongs to one of the c possible outcomes. For example, if you have 3 possible outcomes, you will have to estimate 3 $\vec{\beta}$s i.e. $\vec{\beta_1}$, $\vec{\beta_2}$ and $\vec{\beta_3}$.

In a special case where you have $c = 2$, equation 2 can be reduced to

$$p_1 = \frac{\exp(\vec{\beta_1} \cdot \vec{x}^i)}{\exp(\vec{\beta_1} \cdot \vec{x}^i) + \exp(\vec{\beta_2} \cdot \vec{x}^i)} = \frac{1}{1 + \exp(\vec{\beta_2} - \vec{\beta_1}) \cdot \vec{x}^i} \tag{3}$$

, which is equivanet to a logistic function.

1.4 Optimization

The MLE estimation can be accomplished using Newton-Raphson method