Logistic Regression: From Binary to Multi-Class

Shuiwang Ji
Texas A&M University
College Station, TX 77843
sji@tamu.edu

Yaochen Xie
Texas A&M University
College Station, TX 77843
ethanycx@tamu.edu

1 Introduction

This introduction to the multi-class logistic regression (LR) aims at providing a complete, self-contained, and easy-to-understand introduction to multi-class LR. We start with a quick review of the binary LR and then generalize the binary LR to multi-class case. We further discuss the connections between the binary LR and the multi-class LR. This document is based on lecture notes by Shuiwang Ji and compiled by Yaochen Xie at Texas A&M University. It can be used for undergraduate and graduate level classes.

2 Binary Logistic Regression

The binary LR predicts the label $y_i \in \{-1, +1\}$ for a given sample $\boldsymbol{x}_i$ by estimating a probability $P(y|\boldsymbol{x}_i)$ and comparing with a pre-defined threshold.

Recall the sigmoid function is defined as

$$\theta(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}}, \tag{1}$$

where $s \in \mathbb{R}$ and θ denotes the sigmoid function. The θ maps any value in $\mathbb{R}$ to a number in $(0, 1)$ and meanwhile preserves the order of any two input numbers as $\theta(\cdot)$ is a monotonically increasing function.

The probability is thus represented by

$$P(y|\boldsymbol{x}) = \begin{cases} \theta(\boldsymbol{w}^T\boldsymbol{x}) & \text{if } y = 1 \\ 1 - \theta(\boldsymbol{w}^T\boldsymbol{x}) & \text{if } y = -1. \end{cases}$$

This can also be expressed compactly as

$$P(y|\boldsymbol{x}) = \theta(y\boldsymbol{w}^T\boldsymbol{x}), \tag{2}$$

due to the fact that $\theta(-s) = 1 - \theta(s)$. Note that in the binary case, we only need to estimate one probability, as the probabilities for +1 and -1 sum to one.

3 Multi-Class Logistic Regression

The binary logistic regression assumes that the label $y_i \in \{-1, +1\}$ $(i = 1, \cdots, N)$, while in the multi-class cases there are more than two classes, i.e., $y_i \in \{1, 2, \cdots, K\}$ $(i = 1, \cdots, N)$, where K is the number of classes and N is the number of samples. In this case, we need to estimate the probability for each of the K classes. The hypothesis in binary LR is hence generalized to the multi-class case as

$$\boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{x}) = \begin{bmatrix} P(y=1|\boldsymbol{x}; w) \\ P(y=2|\boldsymbol{x}; w) \\ \cdots \\ P(y=K|\boldsymbol{x}; w) \end{bmatrix} \tag{3}$$

A critical assumption here is that there is no ordinal relationship between the classes. So we will need one linear signal for each of the K classes, which should be independent conditioned on $\boldsymbol{x}$. As a result, in the multi-class LR, we compute K linear signals by the dot product between the input $\boldsymbol{x}$ and K **independent** weight vectors $\boldsymbol{w}_k,\ k = 1, \cdots, K$ as

$$\begin{bmatrix} \boldsymbol{w}_1^T \boldsymbol{x} \\ \boldsymbol{w}_2^T \boldsymbol{x} \\ \vdots \\ \boldsymbol{w}_K^T \boldsymbol{x} \end{bmatrix}. \tag{4}$$

So far, the only thing left to obtain the hypothesis is to map the K linear outputs (as a vector in $\mathbb{R}^K$) to the K probabilities (as a probability distribution among the K classes).

3.1 Softmax

In order to accomplish such a mapping, we introduce the softmax function, which is generalized from the sigmoid function and defined as below. Given a K-dimensional vector $\boldsymbol{v} = [v_1, v_2, \cdots, v_K]^T \in \mathbb{R}^K$,

$$\text{softmax}(\boldsymbol{v}) = \frac{1}{\sum_{k=1}^K e^{v_k}} \begin{bmatrix} e^{v_1} \\ e^{v_2} \\ \vdots \\ e^{v_K} \end{bmatrix}. \tag{5}$$

It is easy to verify that the softmax maps a vector in $\mathbb{R}^K$ to $(0,1)^K$. All elements in the output vector of softmax sum to 1 and their orders are preserved. Thus the hypothesis in (3) can be written as

$$\boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{x}) = \begin{bmatrix} P(y=1|\boldsymbol{x}; w) \\ P(y=2|\boldsymbol{x}; w) \\ \cdots \\ P(y=K|\boldsymbol{x}; w) \end{bmatrix} = \frac{1}{\sum_{k=1}^K e^{\boldsymbol{w}_k^T \boldsymbol{x}}} \begin{bmatrix} e^{\boldsymbol{w}_1^T \boldsymbol{x}} \\ e^{\boldsymbol{w}_2^T \boldsymbol{x}} \\ \cdots \\ e^{\boldsymbol{w}_K^T \boldsymbol{x}} \end{bmatrix}. \tag{6}$$

We will further discuss the connection between the softmax function and the sigmoid function by showing that the sigmoid in binary LR is equivalent to the softmax in multi-class LR when $K = 2$ in Section 4.

3.2 Cross Entropy

We optimize the multi-class LR by minimizing a loss (cost) function, measuring the error between predictions and the true labels, as we did in the binary LR. Therefore, we introduce the cross-entropy in Equation (7) to measure the distance between two probability distributions.

The cross entropy is defined by

$$H(\boldsymbol{P}, \boldsymbol{Q}) = -\sum_{i=1}^K p_i \log(q_i), \tag{7}$$

where $\boldsymbol{P} = (p_1, \cdots, p_K)$ and $\boldsymbol{Q} = (q_1, \cdots, q_K)$ are two probability distributions. In multi-class LR, the two probability distributions are the true distribution and predicted vector in Equation (3), respectively.

Here the true distribution refers to the one-hot encoding of the label. For label k (k is the correct class), the one-hot encoding is defined as a vector whose element being 1 at index k, and 0 everywhere else.

3.3 Loss Function

Now the loss for a training sample $\boldsymbol{x}$ in class c is given by

$$\begin{aligned}
loss(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{w}) &= H(\boldsymbol{y}, \hat{\boldsymbol{y}}) && (8)\\
&= -\sum_k \boldsymbol{y}_k \log \hat{\boldsymbol{y}}_k && (9)\\
&= -\log \hat{\boldsymbol{y}}_c && (10)\\
&= -\log \frac{e^{\boldsymbol{w}_c^T \boldsymbol{x}}}{\sum_{k=1}^K e^{\boldsymbol{w}_k^T \boldsymbol{x}}} && (11)
\end{aligned}$$

where $\boldsymbol{y}$ denotes the one-hot vector and $\hat{\boldsymbol{y}}$ is the predicted distribution $h(\boldsymbol{x}_i)$. And the loss on all samples $(\boldsymbol{X}_i, \boldsymbol{Y}_i)_{i=1}^N$ is

$$loss(\boldsymbol{X}, \boldsymbol{Y}; \boldsymbol{w}) = -\sum_{i=1}^N \sum_{k=1}^K I[y_i = k] \log \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x}_i}}{\sum_{k=1}^K e^{\boldsymbol{w}_k^T \boldsymbol{x}_i}} \tag{12}$$

4 Properties of Multi-class LR

4.1 Shift-invariance in Parameters

The softmax function in multi-class LR has an invariance property when shifting the parameters. Given the weights $\boldsymbol{w} = (\boldsymbol{w}_1, \cdots, \boldsymbol{w}_K)$, suppose we subtract the same vector $\boldsymbol{u}$ from each of the K weight vectors, the outputs of softmax function will remain the same.

Proof. To prove this, let us denote $\boldsymbol{w}' = \{\boldsymbol{w}_i'\}_{i=1}^K$, where $\boldsymbol{w}_i' = \boldsymbol{w}_i - \boldsymbol{u}$. We have

$$\begin{aligned}
P(y = k|\boldsymbol{x}; \boldsymbol{w}') &= \frac{e^{(\boldsymbol{w}_k - \boldsymbol{u})^T \boldsymbol{x}}}{\sum_{i=1}^K e^{(\boldsymbol{w}_i - \boldsymbol{u})^T \boldsymbol{x}}} && (13)\\
&= \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x}} e^{-\boldsymbol{u}^T \boldsymbol{x}}}{\sum_{i=1}^K e^{\boldsymbol{w}_i^T \boldsymbol{x}} e^{-\boldsymbol{u}^T \boldsymbol{x}}} && (14)\\
&= \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x}} e^{-\boldsymbol{u}^T \boldsymbol{x}}}{(\sum_{i=1}^K e^{\boldsymbol{w}_i^T \boldsymbol{x}}) e^{-\boldsymbol{u}^T \boldsymbol{x}}} && (15)\\
&= \frac{e^{(\boldsymbol{w}_k)^T \boldsymbol{x}}}{\sum_{i=1}^K e^{(\boldsymbol{w}_i)^T \boldsymbol{x}}} && (16)\\
&= P(y = k|\boldsymbol{x}; \boldsymbol{w}), && (17)
\end{aligned}$$

which completes the proof. □

4.2 Equivalence to Sigmoid

Once we have proved the shift-invariance, we are able to show that when $K = 2$, the softmax-based multi-class LR is equivalent to the sigmoid-based binary LR. In particular, the hypothesis of both LR are equivalent.

Proof.

$$\boldsymbol{h}_{\boldsymbol{w}}(\boldsymbol{x}) = \frac{1}{e^{\boldsymbol{w}_1^T\boldsymbol{x}} + e^{\boldsymbol{w}_2^T\boldsymbol{x}}}\begin{bmatrix} e^{\boldsymbol{w}_1^T\boldsymbol{x}} \\ e^{\boldsymbol{w}_2^T\boldsymbol{x}} \end{bmatrix} \tag{18}$$

$$= \frac{1}{e^{(\boldsymbol{w}_1-\boldsymbol{w}_1)^T\boldsymbol{x}} + e^{(\boldsymbol{w}_2-\boldsymbol{w}_1)^T\boldsymbol{x}}}\begin{bmatrix} e^{(\boldsymbol{w}_1-\boldsymbol{w}_1)^T\boldsymbol{x}} \\ e^{(\boldsymbol{w}_2-\boldsymbol{w}_1)^T\boldsymbol{x}} \end{bmatrix} \tag{19}$$

$$= \begin{bmatrix} \frac{1}{1+e^{(\boldsymbol{w}_2-\boldsymbol{w}_1)^T\boldsymbol{x}}} \\ \frac{e^{(\boldsymbol{w}_2-\boldsymbol{w}_1)^T\boldsymbol{x}}}{1+e^{(\boldsymbol{w}_2-\boldsymbol{w}_1)^T\boldsymbol{x}}} \end{bmatrix} \tag{20}$$

$$= \begin{bmatrix} \frac{1}{1+e^{-\hat{\boldsymbol{w}}^T\boldsymbol{x}}} \\ \frac{e^{-\hat{\boldsymbol{w}}^T\boldsymbol{x}}}{1+e^{-\hat{\boldsymbol{w}}^T\boldsymbol{x}}} \end{bmatrix} \tag{21}$$

$$= \begin{bmatrix} \frac{1}{1+e^{-\hat{\boldsymbol{w}}^T\boldsymbol{x}}} \\ 1-\frac{1}{1+e^{-\hat{\boldsymbol{w}}^T\boldsymbol{x}}} \end{bmatrix} = \begin{bmatrix} h_{\hat{\boldsymbol{w}}}(\boldsymbol{x}) \\ 1-h_{\hat{\boldsymbol{w}}}(\boldsymbol{x}) \end{bmatrix}, \tag{22}$$

where $\hat{\boldsymbol{w}} = \boldsymbol{w}_1 - \boldsymbol{w}_2$. This completes the proof. □

4.3 Relations between binary and multi-class LR

In the assignment, we've already proved that minimizing the logistic regression loss is equivalent to minimizing the cross-entropy loss with binary outcomes. We hereby show the proof again as below.

Proof.

$$\begin{aligned}
\arg\min_{\boldsymbol{w}} E_{in}(\boldsymbol{w}) &= \arg\min_{\boldsymbol{w}} \frac{1}{N}\sum_{n=1}^{N} \ln(1+e^{-y_n\boldsymbol{w}^T\boldsymbol{x}_n}) \\
&= \arg\min_{\boldsymbol{w}} \frac{1}{N}\sum_{n=1}^{N} \ln\frac{1}{\theta(y_n\boldsymbol{w}^T\boldsymbol{x}_n)} \\
&= \arg\min_{\boldsymbol{w}} \frac{1}{N}\sum_{n=1}^{N} \ln\frac{1}{P(y_n|\boldsymbol{x}_n)} \\
&= \arg\min_{\boldsymbol{w}} \frac{1}{N}\sum_{n=1}^{N} I[y_n=+1]\ln\frac{1}{P(y_n|\boldsymbol{x}_n)} + I[y_n=-1]\ln\frac{1}{P(y_n|\boldsymbol{x}_n)} \\
&= \arg\min_{\boldsymbol{w}} \frac{1}{N}\sum_{n=1}^{N} I[y_n=+1]\ln\frac{1}{h(\boldsymbol{x}_n)} + I[y_n=-1]\ln\frac{1}{1-h(\boldsymbol{x}_n)} \\
&= \arg\min_{\boldsymbol{w}} p\log\frac{1}{q} + (1-p)\log\frac{1}{1-q} \\
&= \arg\min_{\boldsymbol{w}} H(\{p, 1-p\}, \{q, 1-q\})
\end{aligned}$$

where $p = I[y_n = +1]$ and $q = h(\boldsymbol{x}_n)$. This completes the proof. □

The equivalence between logistic regression loss and the cross-entropy loss, as proved above, shows that we always obtain identical weights $\boldsymbol{w}$ by minimizing the two losses. The equivalence between the losses, together with the equivalence between sigmoid and softmax, leads to the conclusion that the binary logistic regression is a particular case of multi-class logistic regression when $K = 2$.

5 Derivative of multi-class LR

To optimize the multi-class LR by gradient descent, we now derive the derivative of softmax and cross entropy. The derivative of the loss function can thus be obtained by the chain rule.

5.1 Derivative of softmax

Let p_i denotes the i-th element of softmax$(\boldsymbol{a})$. Then for $j = i$, we have

$$
\begin{aligned}
\frac{\partial p_i}{\partial a_j} = \frac{\partial p_i}{\partial a_i} =& \frac{\partial \frac{e^{a_i}}{\sum_{k=1}^{K} e^{a_k}}}{\partial a_i} && (23) \\
=& \frac{e^{a_i} \sum_{k=1}^{K} e^{a_k} - e^{2a_i}}{(\sum_{k=1}^{K} e^{a_k})^2} && (24) \\
=& \frac{e^{a_i}}{\sum_{k=1}^{K} e^{a_k}} \cdot \frac{\sum_{k=1}^{K} e^{a_k} - e^{a_i}}{\sum_{k=1}^{K} e^{a_k}} && (25) \\
=& p_i(1 - p_i) && (26) \\
=& p_i(1 - p_j) && (27)
\end{aligned}
$$

And for $j \neq i$,

$$
\begin{aligned}
\frac{\partial p_i}{\partial a_j} =& \frac{\partial \frac{e^{a_i}}{\sum_{k=1}^{K} e^{a_k}}}{\partial a_j} && (28) \\
=& \frac{0 - e^{a_i} e^{a_j}}{(\sum_{k=1}^{K} e^{a_k})^2} && (29) \\
=& - \frac{e^{a_i}}{\sum_{k=1}^{K} e^{a_k}} \cdot \frac{e^{a_j}}{\sum_{k=1}^{K} e^{a_k}} && (30) \\
=& - p_i p_j && (31)
\end{aligned}
$$

If we unify the two cases with the Kronecker delta, we will have

$$\frac{\partial p_i}{\partial a_j} = p_i(\delta_{ij} - p_j),$$

where

$$\delta_{ij} = \begin{cases} 1 & if\ i = j \\ 0 & if\ i \neq j \end{cases}$$

5.2 Derivative of cross entropy loss with softmax

The Cross Entropy Loss is given by:

$$L = -\sum_i y_i log(p_i)$$

where $p_i = \text{softmax}_i(\boldsymbol{a}) = \frac{e^{a_i}}{\sum_{k=1}^{K} e^{a_k}}$ and y_i denotes the i-th element of the one-hot vector. The derivative of cross entropy is

$$
\begin{aligned}
\frac{\partial L}{\partial a_k} =& -\sum_i y_i \frac{\partial \log(p_i)}{\partial a_k} && (32) \\
=& -\sum_i y_i \frac{\partial \log(p_i)}{\partial p_i} \cdot \frac{\partial p_i}{\partial a_k} && (33) \\
=& -\sum_i y_i \frac{1}{p_i} \cdot \frac{\partial p_i}{\partial a_k} && (34) \\
=& -\sum_i y_i \frac{1}{p_i} \cdot p_i(\delta_{ki} - p_k) && (35) \\
=& - y_k(1 - p_k) + \sum_{i \neq k} y_i p_k && (36) \\
=& p_k \sum_{i=1}^{K} y_i - y_k && (37) \\
=& p_k - y_k && (38)
\end{aligned}
$$

Note that here we use the fact that $\sum_{i=1}^{K} y_i = 1$.

Acknowledgements

This work was supported in part by National Science Foundation grants IIS-1908220, IIS-1908198, IIS-1908166, DBI-1147134, DBI-1922969, DBI-1661289, CHE-1738305, National Institutes of Health grant 1R21NS102828, and Defense Advanced Research Projects Agency grant N66001-17-2-4031.