

Logistic Regression: From Binary to Multi-Class

Shuiwang Ji
Department of Computer Science & Engineering
Texas A&M University

Binary Logistic Regression

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

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

where $s \in \mathbb{R}$ and θ denotes the sigmoid function.

- 3 The probability is thus represented by

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

This can also be expressed compactly as

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

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.

Multi-Class Logistic Regression

- 1 In the multi-class cases there are more than two classes, i.e., $y_i \in \{1, 2, \dots, K\}$ ($i = 1, \dots, N$), where K is the number of classes and N is the number of samples.
- 2 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

$$\mathbf{h}_w(\mathbf{x}) = \begin{bmatrix} P(y = 1 | \mathbf{x}; w) \\ P(y = 2 | \mathbf{x}; w) \\ \dots \\ P(y = K | \mathbf{x}; w) \end{bmatrix} \quad (3)$$

- 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 \mathbf{x} .

- ① As a result, in the multi-class LR, we compute K linear signals by the dot product between the input \mathbf{x} and K **independent** weight vectors \mathbf{w}_k , $k = 1, \dots, K$ as

$$\begin{bmatrix} \mathbf{w}_1^T \mathbf{x} \\ \mathbf{w}_2^T \mathbf{x} \\ \vdots \\ \mathbf{w}_K^T \mathbf{x} \end{bmatrix}. \quad (4)$$

- ② We then need 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 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 $\mathbf{v} = [v_1, v_2, \dots, v_K]^T \in \mathbb{R}^K$,

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

- ① 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

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

- ② 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 $K = 2$

Cross Entropy

- 1 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.
- 2 The cross entropy is defined by

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

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

- 3 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.

- ① Now the loss for a training sample \mathbf{x} in class c is given by

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

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

$$\text{loss}(\mathbf{X}, \mathbf{Y}; \mathbf{w}) = - \sum_{i=1}^N \sum_{k=1}^K I[y_i = k] \log \frac{e^{\mathbf{w}_k^T \mathbf{x}_i}}{\sum_{k=1}^K e^{\mathbf{w}_k^T \mathbf{x}_i}} \quad (8)$$

Shift-invariance in Parameters

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

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

$$P(y = k | \mathbf{x}; \mathbf{w}') = \frac{e^{(\mathbf{w}_k - \mathbf{u})^T \mathbf{x}}}{\sum_{i=1}^K e^{(\mathbf{w}_i - \mathbf{u})^T \mathbf{x}}} \quad (9)$$

$$= \frac{e^{\mathbf{w}_k^T \mathbf{x}} e^{-\mathbf{u}^T \mathbf{x}}}{\sum_{i=1}^K e^{\mathbf{w}_i^T \mathbf{x}} e^{-\mathbf{u}^T \mathbf{x}}} \quad (10)$$

$$= \frac{e^{\mathbf{w}_k^T \mathbf{x}} e^{-\mathbf{u}^T \mathbf{x}}}{(\sum_{i=1}^K e^{\mathbf{w}_i^T \mathbf{x}}) e^{-\mathbf{u}^T \mathbf{x}}} \quad (11)$$

$$= \frac{e^{(\mathbf{w}_k)^T \mathbf{x}}}{\sum_{i=1}^K e^{(\mathbf{w}_i)^T \mathbf{x}}} \quad (12)$$

$$= P(y = k | \mathbf{x}; \mathbf{w}), \quad (13)$$

which completes the proof.

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.

$$\mathbf{h}_{\mathbf{w}}(\mathbf{x}) = \frac{1}{e^{\mathbf{w}_1^T \mathbf{x}} + e^{\mathbf{w}_2^T \mathbf{x}}} \begin{bmatrix} e^{\mathbf{w}_1^T \mathbf{x}} \\ e^{\mathbf{w}_2^T \mathbf{x}} \end{bmatrix} \quad (14)$$

$$= \frac{1}{e^{(\mathbf{w}_1 - \mathbf{w}_1)^T \mathbf{x}} + e^{(\mathbf{w}_2 - \mathbf{w}_1)^T \mathbf{x}}} \begin{bmatrix} e^{(\mathbf{w}_1 - \mathbf{w}_1)^T \mathbf{x}} \\ e^{(\mathbf{w}_2 - \mathbf{w}_1)^T \mathbf{x}} \end{bmatrix} \quad (15)$$

$$= \begin{bmatrix} \frac{1}{1 + e^{(\mathbf{w}_2 - \mathbf{w}_1)^T \mathbf{x}}} \\ \frac{e^{(\mathbf{w}_2 - \mathbf{w}_1)^T \mathbf{x}}}{1 + e^{(\mathbf{w}_2 - \mathbf{w}_1)^T \mathbf{x}}} \end{bmatrix} \quad (16)$$

$$= \begin{bmatrix} \frac{1}{1 + e^{-\hat{\mathbf{w}}^T \mathbf{x}}} \\ \frac{e^{-\hat{\mathbf{w}}^T \mathbf{x}}}{1 + e^{-\hat{\mathbf{w}}^T \mathbf{x}}} \end{bmatrix} \quad (17)$$

$$= \begin{bmatrix} \frac{1}{1 + e^{-\hat{\mathbf{w}}^T \mathbf{x}}} \\ 1 - \frac{1}{1 + e^{-\hat{\mathbf{w}}^T \mathbf{x}}} \end{bmatrix} = \begin{bmatrix} h_{\hat{\mathbf{w}}}(\mathbf{x}) \\ 1 - h_{\hat{\mathbf{w}}}(\mathbf{x}) \end{bmatrix}, \quad (18)$$

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

- 1 Now we show that minimizing the logistic regression loss is equivalent to minimizing the cross-entropy loss with binary outcomes.
- 2 The equivalence between logistic regression loss and the cross-entropy loss, as shown below, proves that we always obtain identical weights \mathbf{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 $K = 2$.

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

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

THANKS!