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 $\boldsymbol{x}_i$ by estimating a probability $P(y|\boldsymbol{x}_i)$ and comparing with a pre-defined threshold.
2. Recall the sigmoid function is defined as

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

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

3. The probability is thus represented by

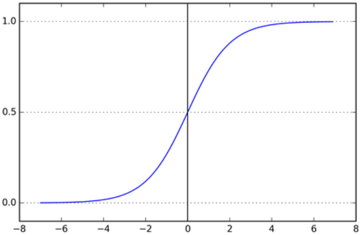
$$P(y|\boldsymbol{x}) = \begin{cases} \theta(\boldsymbol{w}^T\boldsymbol{x}) & \text{if } y = 1 \\ 1 - \theta(\boldsymbol{w}^T\boldsymbol{x}) & \text{if } y = -1. \end{cases}$$

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.

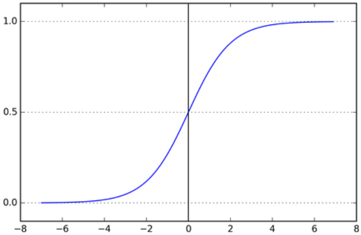
Properties of the Sigmoid Function



1. $0 < \theta(s) < 1, \forall s$
2. $\theta(-s) = 1 - \theta(s)$
3. $\theta(\cdot)$ is a monotonic function

Why are they important?

Properties of the Sigmoid Function



1. $0 < \theta(s) < 1, \forall s$
2. $\theta(-s) = 1 - \theta(s)$
3. $\theta(\cdot)$ is a monotonic function

Why are they important?

1. Probabilistic interpretation
2. Compact representation
3. Linear model, why?

Is logistic regression a linear model? Why?

Multi-Class Logistic Regression - Prediction

1. 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.
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

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

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 $\boldsymbol{x}$.

Softmax

1. 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

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

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 $\boldsymbol{v} = [v_1, v_2, \cdots, v_K]^T \in \mathbb{R}^K$,

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

Softmax

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

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

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 $K=2$

Training with 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

Only for one data sample

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

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.

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.

Loss Function

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

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

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

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

I[yi = c] - This is a indicator function to identify the correct class yi

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

$$\begin{aligned}
P(y = k|\boldsymbol{x}; \boldsymbol{w}') &= \frac{e^{(\boldsymbol{w}_k - \boldsymbol{u})^T \boldsymbol{x}}}{\sum_{i=1}^K e^{(\boldsymbol{w}_i - \boldsymbol{u})^T \boldsymbol{x}}} && (9)\\
&= \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x}} e^{-\boldsymbol{u}^T \boldsymbol{x}}}{\sum_{i=1}^K e^{\boldsymbol{w}_i^T \boldsymbol{x}} e^{-\boldsymbol{u}^T \boldsymbol{x}}} && (10)\\
&= \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x}} e^{-\boldsymbol{u}^T \boldsymbol{x}}}{(\sum_{i=1}^K e^{\boldsymbol{w}_i^T \boldsymbol{x}}) e^{-\boldsymbol{u}^T \boldsymbol{x}}} && (11)\\
&= \frac{e^{(\boldsymbol{w}_k)^T \boldsymbol{x}}}{\sum_{i=1}^K e^{(\boldsymbol{w}_i)^T \boldsymbol{x}}} && (12)\\
&= P(y = k|\boldsymbol{x}; \boldsymbol{w}), && (13)
\end{aligned}$$

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.

Proof

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

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

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

$$= \begin{bmatrix} \frac{1}{1+e^{-\hat{\boldsymbol{w}}^T\boldsymbol{x}}} \\ \frac{e^{-\hat{\boldsymbol{w}}^T\boldsymbol{x}}}{1+e^{-\hat{\boldsymbol{w}}^T\boldsymbol{x}}} \end{bmatrix} \tag{17}$$

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

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

Equivalence of Loss Function

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 $\boldsymbol{w}$ 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 $K = 2$.

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.

Derivative of Loss Function

The notes (Logistic Regression: From Binary to Multi-Class) contain details on derivative of cross entropy loss function, which is necessary for your homework. All you need are:

1. Univariate calculus
2. Chain rule
3. $\frac{\partial(\mathbf{w}^\mathsf{T}\mathbf{b})}{\partial \mathbf{w}} = \mathbf{b}$

THANKS!