Logistic Regression: From Binary to Multi-Class

Shuiwang Ji
Department of Computer Science & Engineering
Texas A&M University
The binary LR predicts the label $y_i \in \{-1, +1\}$ for a given sample $x_i$ by estimating a probability $P(y|x_i)$ and comparing with a pre-defined threshold.

Recall the sigmoid function is defined as

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

(1)

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

The probability is thus represented by

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

This can also be expressed compactly as

$$P(y|x) = \theta(yw^T x),$$

(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?
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

\[
h_w(x) = \begin{bmatrix} P(y = 1|x; w) \\ P(y = 2|x; w) \\ \vdots \\ P(y = K|x; w) \end{bmatrix}
\]  

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 \( x \).
As a result, in the multi-class LR, we compute $K$ linear signals by the dot product between the input $x$ and $K$ independent weight vectors $w_k, k = 1, \cdots, K$ as

$$
\begin{bmatrix}
    w_1^T x \\
    w_2^T x \\
    \vdots \\
    w_K^T x
\end{bmatrix}.
$$

(4)

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).

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

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

(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

$$h_w(x) = \begin{bmatrix} P(y = 1|x; w) \\ P(y = 2|x; w) \\ \vdots \\ P(y = K|x; w) \end{bmatrix} = \frac{1}{\sum_{k=1}^{K} e^{w_k^T x}} \begin{bmatrix} e^{w_1^T x} \\ e^{w_2^T x} \\ \vdots \\ e^{w_K^T x} \end{bmatrix}. \tag{6}$$

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 $K = 2$. 
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

\[ H(P, Q) = - \sum_{i=1}^{K} p_i \log(q_i), \]  

where \( P = (p_1, \ldots, p_K) \) and \( Q = (q_1, \ldots, q_K) \) 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.
Now the loss for a training sample \( x \) in class \( c \) is given by

\[
\text{loss}(x, y; w) = H(y, \hat{y}) = - \sum_k y_k \log \hat{y}_k = - \log \hat{y}_c = - \log \frac{\exp^{w^T_c x}}{\sum_{k=1}^K \exp^{w^T_k x}}
\]

where \( y \) denotes the one-hot vector and \( \hat{y} \) is the predicted distribution \( h(x_i) \). And the loss on all samples \((X_i, Y_i)_{i=1}^N\) is

\[
\text{loss}(X, Y; w) = - \sum_{i=1}^N \sum_{c=1}^K I[y_i = c] \log \frac{\exp^{w^T_c x_i}}{\sum_{k=1}^K \exp^{w^T_k x_i}}
\]

(8)
The softmax function in multi-class LR has an invariance property when shifting the parameters. Given the weights $\mathbf{w} = (\mathbf{w}_1, \cdots, \mathbf{w}_K)$, suppose we subtract the same vector $\mathbf{u}$ from each of the $K$ weight vectors, the outputs of softmax function will remain the same.
Proof

To prove this, let us denote \( w' = \{w'_i\}^K_{i=1} \), where \( w'_i = w_i - u \). We have

\[
P(y = k|x; w') = \frac{e^{(w_k-u)^T x}}{\sum_{i=1}^{K} e^{(w_i-u)^T x}}
\]

\[
= \frac{e^{w_k^T x} e^{-u^T x}}{\sum_{i=1}^{K} e^{w_i^T x} e^{-u^T x}}
\]

\[
= \frac{e^{w_k^T x} e^{-u^T x}}{(\sum_{i=1}^{K} e^{w_i^T x}) e^{-u^T x}}
\]

\[
= \frac{e^{(w_k)^T x}}{\sum_{i=1}^{K} e^{(w_i)^T x}}
\]

\[
= P(y = k|x; w),
\]

which completes the proof.
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

\[ h_w(x) = \frac{1}{e^{w_1^T x} + e^{w_2^T x}} \begin{bmatrix} e^{w_1^T x} \\ e^{w_2^T x} \end{bmatrix} \]  

(14)

\[ = \frac{1}{e^{(w_1-w_1)^T x} + e^{(w_2-w_1)^T x}} \begin{bmatrix} e^{(w_1-w_1)^T x} \\ e^{(w_2-w_1)^T x} \end{bmatrix} \]  

(15)

\[ = \begin{bmatrix} \frac{1}{1+e^{(w_2-w_1)^T x}} \\ \frac{e^{(w_2-w_1)^T x}}{1+e^{(w_2-w_1)^T x}} \end{bmatrix} \]  

(16)

\[ = \begin{bmatrix} \frac{1}{1+e^{-\hat{w}^T x}} \\ \frac{e^{-\hat{w}^T x}}{1+e^{-\hat{w}^T x}} \end{bmatrix} \]  

(17)

\[ = \begin{bmatrix} \frac{1}{1+e^{-\hat{w}^T x}} \\ \frac{1}{1+e^{-\hat{w}^T x}} \end{bmatrix} = \begin{bmatrix} h_{\hat{w}}(x) \\ 1 - h_{\hat{w}}(x) \end{bmatrix}, \]  

(18)

where \( \hat{w} = w_1 - w_2 \). This completes the proof.
Now we show that minimizing the logistic regression loss is equivalent to minimizing the cross-entropy loss with binary outcomes.

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 $K = 2$. 
Proof

\[ \arg \min_w E_{in}(w) = \arg \min_w \frac{1}{N} \sum_{n=1}^{N} \ln \left( 1 + e^{-y_n w^T x_n} \right) \]

\[ = \arg \min_w \frac{1}{N} \sum_{n=1}^{N} \ln \frac{1}{\theta(y_n w^T x_n)} \]

\[ = \arg \min_w \frac{1}{N} \sum_{n=1}^{N} \ln \frac{1}{P(y_n | x_n)} \]

\[ = \arg \min_w \frac{1}{N} \sum_{n=1}^{N} [y_n = +1] \ln \frac{1}{P(y_n | x_n)} + [y_n = -1] \ln \frac{1}{P(y_n | x_n)} \]

\[ = \arg \min_w \frac{1}{N} \sum_{n=1}^{N} [y_n = +1] \ln \frac{1}{h(x_n)} + [y_n = -1] \ln \frac{1}{1 - h(x_n)} \]

\[ = \arg \min_p p \log \frac{1}{q} + (1 - p) \log \frac{1}{1 - q} \]

\[ = \arg \min_w H(\{p, 1 - p\}, \{q, 1 - q\}) \]

where \( p = I[y_n = +1] \) and \( q = h(x_n) \). This completes the proof.
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. \[
\frac{\partial (w^T b)}{\partial w} = b
\]
THANKS!