CSCI 5525 Machine Learning Fall 2019

Lecture 5: Logistic Regression

Feb 10 2020

Lecturer: Steven Wu *Scribe: Steven Wu*

Last lecture, we give several convex surrogate loss functions to replace the zero-one loss function, which is NP-hard to optimize. Now let us look into one of the examples, logistic loss: given parameter $\mathbf{w}$ and example $(x_i, y_i) \in \mathbb{R}^d \times \{\pm 1\}$, the logistic loss of $\mathbf{w}$ on example (x_i, y_i) is defined as

$$\ln\left(1 + \exp(-y_i \mathbf{w}^\intercal x_i)\right)$$

This loss function is used in logistic regression. We will introduce the statistical model behind logistic regression, and show that the ERM problem for logistic regression is the same as the relevant maximum likelihood estimation (MLE) problem.

1 MLE Derivation

For this derivation it is more convenient to have $\mathcal{Y} = \{0, 1\}$. Note that for any label $y_i \in \{0, 1\}$, we also have the "signed" version of the label $2y_i - 1 \in \{-1, 1\}$. Recall that in general supervised learning setting, the learner receive examples $(x_1, y_1), \ldots, (x_n, y_n)$ drawn iid from some distribution P over labeled examples. We will make the following parametric assumption on P:

$$y_i \mid x_i \sim \text{Bern}(\sigma(\mathbf{w}^\intercal x_i))$$

where Bern denotes the Bernoulli distribution, and σ is the *logistic function* defined as follows

$$\sigma(z) = \frac{1}{1 + \exp(-z)} = \frac{\exp(z)}{1 + \exp(z)}$$

See Figure 1 for a visualization of the logistic function. In general, the logistic function is a useful function to convert real values into probabilities (in the range of $(0, 1)$). If $\mathbf{w}^\intercal x$ increases, then $\sigma(\mathbf{w}^\intercal x)$ also increases, and so does the probability of $Y = 1$.

Recall that MLE procedure finds a model parameter to maximize

$$P(\text{observed data} \mid \text{model paramter})$$

Under logistic regression model, this means finding a weight vector $\mathbf{w}$ that maximize the conditional probability:

$$P(y_1, x_1 \ldots, x_n, y_n \mid \mathbf{w})$$

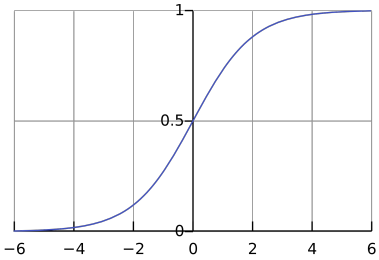

Figure 1: Logistic Function σ. Observe that $\sigma(z) > 1/2$ if and only if $z > 0$, and $\sigma(z)+\sigma(-z) = 1$.

Recall in the MLE derivation for linear regression, we simplified the maximization problem as follows:

$$\begin{aligned}
\mathbf{w} &= \underset{\mathbf{w}}{\operatorname{argmax}}\, P(y_1, x_1, ..., y_n, x_n|\mathbf{w}) \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i, x_i|\mathbf{w}) && \text{(Independence)} \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i|x_i, \mathbf{w})P(x_i|\mathbf{w}) \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i|x_i, \mathbf{w})P(x_i) && (x_i \text{ is independent of } \mathbf{w}) \\
&= \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i|x_i, \mathbf{w}) && (P(x_i) \text{ does not depend on } \mathbf{w})
\end{aligned}$$

This means finding a weight vector $\mathbf{w}$ that maximize the conditional probability (and hence the phrase maximum likelihood estimation):

$$\prod_{i=1}^{n} \sigma(\mathbf{w}^\intercal x_i)^{y_i}(1 - \sigma(\mathbf{w}^\intercal x_i))^{1-y_i}$$

Equivalently, we would like to find the $\mathbf{w}$ to maximize the log likelihood:

$$\begin{aligned}
&\ln \prod_{i=1}^{n} \sigma(\mathbf{w}^\intercal x_i)^{y_i}(1 - \sigma(\mathbf{w}^\intercal x_i))^{1-y_i} \\
&= \sum_{i=1}^{n} (y_i \ln(\sigma(\mathbf{w}^\intercal x_i)) + (1 - y_i) \ln(1 - \sigma(\mathbf{w}^\intercal x_i))) \\
&= -\sum_{i=1}^{n} (y_i \ln(1 + \exp(-\mathbf{w}^\intercal x_i)) + (1 - y_i) \ln(1 + \exp(\mathbf{w}^\intercal x_i))) && \text{(Plugging in } \sigma) \\
&= -\sum_{i=1}^{n} (\ln(1 + \exp(-(2y_i - 1)\mathbf{w}^\intercal x_i)))
\end{aligned}$$

Note that the last step is essentially a change of variable by switching the labels to our old labels $2y_i - 1 \in \{\pm 1\}$. Therefore, maximizing the log-likelihood is exactly minimizing the following

$$\sum_{i=1}^{n} \ln(1 + \exp(-(2y_i - 1)\mathbf{w}^\intercal x_i))$$

This is exactly the ERM problem for logistic regression. Thus, the ERM problem in logistic regression is also the MLE problem under the statistical model we describe above.

Solution To find the values of the parameters at minimum, we can try to find solutions for

$$\nabla_{\mathbf{w}} \sum_{i=1}^{n} \ln(1 + \exp(-y_i \mathbf{w}^\intercal x_i)) = 0$$

This equation has no closed form solution, so we will use gradient descent on the negative log likelihood $\ell(\mathbf{w}) = \sum_{i=1}^{n} \log(1 + \exp(-y_i \mathbf{w}^\intercal x_i))$.

MAP Estimate Similar to the MAP estimation for linear regression, we can also have a MAP estimate for logistic regression. In the MAP estimate, we assume $\mathbf{w}$ is drawn from a prior belief distribution, which is often the multivariate Gaussian distribution

$$\mathbf{w} \sim \mathcal{N}(\vec{0}, \sigma^2 I)$$

Our goal in MAP is to find the most likely model parameters given the data, i.e., the parameters that maximaize the posterior:

$$P(\mathbf{w} \mid x_1, y_1, \ldots, x_n, y_n) \propto P(y_1, \ldots, y_n \mid x_1, \ldots, x_n, \mathbf{w})\ P(\mathbf{w}) \qquad (\propto \text{ means proportional to})$$

One can show (maybe in a homework problem) that

$$\begin{aligned} \widehat{\mathbf{w}}_{MAP} &= \operatorname*{argmax}_{\mathbf{w}} \ln\ (P(y_1, \ldots, y_n \mid x_1, \ldots, x_n, \mathbf{w}) P(\mathbf{w})) \\ &= \operatorname*{argmin}_{\mathbf{w}} \sum_{i=1}^{n} \ln(1 + e^{-(2y_i - 1)\mathbf{w}^T \vec{x}}) + \lambda \mathbf{w}^\top \mathbf{w} \end{aligned}$$

where $\lambda = \frac{1}{2\sigma^2}$. This also corresponds to the regularized logistic regression with ℓ_2 regularization. This optimization problem also has no closed-form solutions, so we will use gradient descent to optimize the regularized loss function.

2 Multiclass Classification

Now we extend these ideas to multiclass classification with $\mathcal{Y} = \{1, \ldots, K\}$.

To define a linear predictor in this setting, let us consider a linear *score* function $f\colon \mathbb{R}^d \to \mathbb{R}^k$ such that $f(x) = W^\intercal x$ with a matrix $W \in \mathbb{R}^{d\times K}$. Intuively, for each example x, the j-th coordinate of $f(x)$, denoted $f(x)_j$, is a score that measures how "good" the j-th label is for this feature x. Analogously, in logistic regression $\mathbf{w}^\intercal x$ essentially provides a score for the label 1, and the score for label 0 is always 0.

To make predictions based on the scores, we will turn score vector $f(x)$ into probability distributions over the K labels. We will write the probability simplex over K labels as

$$\Delta_K = \{v \in \mathbb{R}^K_{\geq 0}\colon \sum_i p_i = 1\}$$

In logistic regression, this is done via the logistic function. For multiclass, we can use the *multinomial logit model* and define a probability vector $\hat{f}(x) \in \Delta_K$ such that each coordinate j satisfies:

$$\hat{f}(x)_j \propto \exp(f(x)_j)$$

By normalization, we have

$$\hat{f}(x)_j = \frac{\exp(f(x)_j)}{\sum_{j'=1}^K \exp(f(x)_{j'})}$$

Now we will define a new loss function to measure the prediction quality of $\hat{f}$.

Cross-entropoy. Given two probability vectors $p, q \in \Delta_K$, the cross-entropy of p and q is

$$H(p,q) = -\sum_{i=1}^K p_i \ln q_i$$

In the special case when $p = q$, we have $H(p,q)$ as the entropy of p, denoted $H(p)$, since

$$H(p,q) = -\sum_{i=1}^K p_i \ln q_i = \underbrace{H(p)}_{\text{Entropy}} + \underbrace{\text{KL}(p,q)}_{\text{KL Divergence}}$$

where the KL divergence term goes to 0 with $p = q$.

To use the cross-entropy as a loss function, we need to encode the true label y_i also as a probability vector. We can do that by rewriting each label y as $\tilde{y} = e_y$ (the standard basis vector) for any $y \in \{1, \ldots, K\}$. Then given any encoded label $\tilde{y}$ (from its true label y) and real-valued score vector $f(x) \in \mathbb{R}^K$ (along with its induced probabilistic prediction $\hat{f}(x) \in \Delta_K$), we can

define the the cross-entropy loss as follows:

$$\begin{aligned}
\ell_{\text{ce}}(\tilde{y}, f(x)) &= H(\tilde{y}, \hat{y}) \\
&= -\sum_{j=1}^{K} \tilde{y}_j \ln \left(\frac{\exp(f(x)_j)}{\sum_{j=1}^{K} \exp(f(x)_j)} \right) \\
&= -\ln \left(\frac{\exp(f(x)_y)}{\sum_{j=1}^{k} \exp(f(x)_j)} \right) \\
&= -f(x)_y + \ln \sum_{j=1}^{K} \exp(f(x)_j)
\end{aligned}$$