CSCI 5525 Machine Learning Fall 2019

Lecture 5: Linear Classification (Part 2)

Sep 17 2019

Lecturer: Steven Wu *Scribe: Steven Wu*

1 Logistic Regression

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—logisitic loss—a loss function 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.

MLE Derivation for logistic regression For now, we consider $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{0, 1\}$. 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. Here we will make certain parametric assumption on P: we assume the conditional probability function has the following form

$$Y \mid X = x \sim \text{Bern}(\sigma(\mathbf{w}^\intercal x))$$

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, which is a useful function to convert real values into probabilities (in the range of $(0, 1)$). Obviously, if $\mathbf{w}^\intercal x$ increases, then $\sigma(\mathbf{w}^\intercal x)$ also increases, and so does the probability of $Y = 1$.

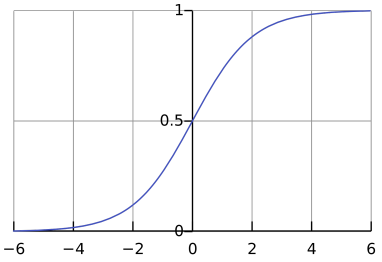

Figure 1: Logistic Function σ. Observe that $\sigma(z) > 1/2$ if and only if $z > 0$, and $\sigma(z) + \sigma(-z) = 1$.

Under this statistical model, we would like to find a weight vector $\mathbf{w}$ that maximize the conditional probability (and hence the phrase maximum likelihood estimation):

$$\mathbb{P}_P(y_1, \dots, y_n \mid x_1, \dots, x_n, \mathbf{w}) = \prod_{i=1}^{n} \sigma(\mathbf{w}^\intercal x_i)^{y_i}(1 - \sigma(\mathbf{w}^\intercal x_i))^{1-y_i}$$

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

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

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.

In binary classification, the predictor $f^* \colon \mathcal{X} \to \mathcal{Y}$ with the smallest risk (that is, error rate) $\mathcal{R}(f) = \mathbb{P}_P[f(X) \neq Y]$ is called *Bayes optimal predictor*. In particular, f^* uses the following prediction rule: for any $x \in \mathcal{X}$:

$$f^*(x) = \mathbf{1}[\mathbb{P}_P(Y = 1 \mid X = x) \geq 1/2].$$

More generally, for multi-class classification with $\mathcal{Y} = \{1, \dots, K\}$, the Bayes optimal predictor predicts:

$$f^*(x) = \arg\max_{y \in \mathcal{Y}} \mathbb{P}_P(Y = y \mid X = x)$$

Now let us go into multiclass classification.

2 Multiclass Classification

How do we extend the ideas in logistic regression to multiclass classification with $\mathcal{Y} = \{1, \dots, K\}$? For starter, 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 generalize the idea of logistic regression, we would like to turn score vectors $f(x)$ 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:

$$\hat{f}(x)_j \propto \exp(f(x)_j) \qquad (\propto \text{ reads "proportional to"})$$

By normalization, we have

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

Now how do we measure the accuracy of a probablistic predictor like $\hat{f}$? Let us introduce a new loss function.

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

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

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 $y \in \{1, \dots, K\}$. 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:

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

Relationship to margin maximization. Recall that in binary classification, the margin is defined as $y(\mathbf{w}^\intercal x)$, where $y_i \in \{\pm 1\}$. For multiclass, we can extend the idea of margin to the following:

$$f(x)_y - \max_{j \neq y} f(x)_j$$

which measures the degree to which f is correct. How does the margin relate to the cross-entropy loss? First, observe that $\ln \sum_{j=1}^{K} \exp(f(x)_j) \approx \max_j f(x)_j$, so

$$\ell_{\text{ce}}(\tilde{y}_i, f(x)) \approx -f(x)_y + \max_j f(x)_j$$

Thus, minimizing cross-entropy loss is implicitly maximizing margins.