DS 5220, Lecture 5: Logistic Regression Using Gradient Descent

September 20, 2024

Here, we'll look at the logistic regression model step-by-step and describe a gradient descent algorithm to solve the regression model. Suppose we have an input set $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where every x_i is a p-dimensional feature vector, and y_i is a binary label between $+1$ or -1.

In the logit model, we want to know what's the probability that a given x has a certain label, in this case: $\Pr[y = +1 \mid x]$ and $\Pr[y = -1| \mid x]$. We'll assume that the probabilities follow the logistic function. For example, suppose the true label is $+1$, then we want $\Pr[y = +1 \mid x]$ to be as close to 1 as possible. Using the logistic function, we may represent this as:

$$\Pr\left[y = +1 \mid x\right] = \frac{\exp\left(\beta_0 + \sum_{i=1}^{p} \beta_i x_i\right)}{1 + \exp\left(\beta_0 + \sum_{i=1}^{p} \beta_i x_i\right)}. \tag{1}$$

The logistic loss (or log loss) is the negative log-likelihood of the above probability, which is

$$\begin{aligned} -\log \Pr\left[y = +1 \mid x\right] &= -\log \frac{\exp\left(\beta_0 + \sum_{i=1}^{p} \beta_i x_i\right)}{1 + \exp\left(\beta_0 + \sum_{i=1}^{p} \beta_i x_i\right)} \\ &= \log \frac{1 + \exp\left(\beta_0 + \sum_{i=1}^{p} \beta_i x_i\right)}{\exp\left(\beta_0 + \sum_{i=1}^{p} \beta_i x_i\right)} \\ &= \log\left(1 + \exp\left(-\beta_0 - \sum_{i=1}^{p} \beta_i x_i\right)\right) \end{aligned}$$

At the other extreme, when the true label is -1, we want the probability of (1) to be as low as possible. Instead, the log loss becomes

$$\log\left(1 + \exp\left(\beta_0 + \sum_{i=1}^{p} \beta_i x_i\right)\right).$$

Taken together, we may write the averaged log-loss in the training set as

$$\hat{L}(\beta) = \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \cdot \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}\right)\right)\right),$$

where $x_{i,j}$ is the j-the entry of x_i.

Unlike the least squares problem, logistic regression does not permit a closed-form solution. One way to solve this regression problem is using an optimization algorithm such as gradient descent. We need to compute the gradient of the loss, $\nabla\hat{L}(\beta)$. Then, we set a step size parameter η_t (usually between 0 and 1), for $t = 1, 2, \ldots, T$. With the gradient, we can update β as follows:

$$\beta^{(t)} \leftarrow \beta^{(t-1)} - \eta_t \cdot \nabla\hat{L}(\beta^{(t-1)}),$$

for $t = 1, 2, \ldots, T$.

Recall that the gradient is a vector that includes the entry-wise partial derivative of $\hat{L}$. Let's look at one entry as an example. For a particular (x_i, y_i), let's look at the partial derivative of the log-loss over β_j:

$$\frac{\partial \log\Big(1 + \exp\big(-y_i \cdot (\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j})\big)\Big)}{\partial \beta_j} = \frac{\exp\Big(-y_i \cdot (\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j})\Big)}{1 + \exp\Big(-y_i \cdot (\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j})\Big)} \times (-y_i x_{i,j}),$$

for any $j = 1, 2, \ldots, p$. As for β_0, the partial derivative is similar:

$$\frac{\partial \log\Big(1 + \exp\big(-y_i \cdot (\beta_0 + \sum_{j=1}^{p} \beta_i x_{i,j})\big)\Big)}{\partial \beta_0} = \frac{\exp\Big(-y_i \cdot \Big(\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}\Big)\Big)}{1 + \exp\Big(-y_i \cdot \Big(\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}\Big)\Big)} \times (-y_i).$$

Taken together, we have obtained the gradient of $\hat{L}$.

Lastly, we'll show that the log loss, $\ell(x) = \log(1 + \exp(-x))$ is a convex function. Recall that a function is convex if and only if $\ell''(x) \geq 0$, or equivalently, $\alpha\ell(x) + (1-\alpha)\ell(y) \geq \ell(\alpha x + (1-\alpha)y)$.

$$\ell'(x) = -\frac{\exp(-x)}{1 + \exp(-x)} = \frac{-1}{1 + \exp(x)},$$
$$\ell''(x) = \frac{\exp(x)}{(1 + \exp(x))^2} > 0.$$

With a bit more calculation, one could show that for minimizing a convex function, the gradient descent algorithm (starting from a random initialization) will eventually converge to a global minimizer that is approximately optimal for minimizing $\hat{L}(\beta)$.