Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

Fall 2022

Gradient Descent for Logistic Regression (WIP)

Summary

Now, we have developed all the tool we need to do binary classification with LLC:

- A **linear** model that lets us **combine** our variables,

$$u(x) = \theta^T x + \theta_0 \tag{1}$$

- A **logistic** model that lets us get the **probability** of a classification,

$$\sigma(u) = \frac{1}{1 + e^{-u}} \tag{2}$$

- A **threshold value** we use to determine how to **classify** our data,

$$h(x; \theta) = \begin{cases} +1 & \text{if } \sigma(u(x)) > \sigma_{thresh} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

- A **loss function** NLL we use to **evaluate** our model performance:

$$\mathcal{L}_{nll}(g^{(i)}, \quad y^{(i)}) = -\left(y^{(i)} \log g^{(i)} + \left(1 - y^{(i)}\right) \log\left(1 - g^{(i)}\right)\right)$$

- And an **objective function** we can **optimize**:

$$J_{lr}(\theta, \theta_0; \mathcal{D}) = \lambda \|\theta\|^2 + \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{nll}(g^{(i)}, \quad y^{(i)}) \tag{4}$$

We have everything we need to do optimization.

The problem: Gradient Descent

We want to do **gradient descent** to minimize J_{lr}

$$R(\theta) + J_{lr}(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{nll}(g^{(i)}, \quad y^{(i)}) \tag{5}$$

We want repeatedly **adjust** our model $\Theta = (\theta, \theta_0)$ to improve J_{lr}. To do that, we want the gradients for θ and θ_0. Let's start with θ.

$$\nabla_\theta J_{lr} = \frac{\partial J_{lr}}{\partial \theta} \tag{6}$$

First, J_{lr} has **two** terms, so we'll separate them.

$$\nabla_\theta J_{lr} = \frac{\partial R}{\partial \theta} + \frac{1}{n}\sum_{i=1}^{n} \frac{\partial \mathcal{L}_{NLL}}{\partial \theta}(g^{(i)}, \quad y^{(i)}) \tag{7}$$

The regularization term is pretty easy, because we did it last chapter:

$$\frac{\partial R}{\partial \theta} = 2\lambda\theta \tag{8}$$

But what about our first term?

Getting the gradient: Chain Rule

Now, we just need to do

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \theta}(g, y) \tag{9}$$

With our $\mathcal{L}_{NLL}$ term, we run into an issue: how do we take the **derivative**? The function is very, very deeply **nested**. In our case:

x **affects** $u(x)$. $u(x)$ **affects** $\sigma(u)$. $\sigma(u) = g$ **affects** $\mathcal{L}_{NLL}(g, y)$, which finally **affects** $J(\theta, \theta_0)$.

How do we represent this **chain** of functions? With the **chain rule**:

$$\frac{\partial A}{\partial C} = \frac{\partial A}{\partial B} \cdot \frac{\partial B}{\partial C} \tag{10}$$

So, we'll build up a **chain rule** for our needs. We'll use $g = \sigma(u)$.

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \theta} = \frac{\partial \mathcal{L}_{NLL}}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial \theta} \tag{11}$$

Sigma contains u, so we'll use that instead:

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \theta} = \frac{\partial \mathcal{L}_{NLL}}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial u} \cdot \frac{\partial u}{\partial \theta} \tag{12}$$

This is our full **chain rule**!

Key Equation 1

The **gradient** of **NLL** can be calculated using the **chain rule**:

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \theta} = \frac{\partial \mathcal{L}_{NLL}}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial u} \cdot \frac{\partial u}{\partial \theta} \tag{13}$$

Getting our individual derivatives

We can take the derivative of each of these objects. First, let's look at $\mathcal{L}_{NLL}$

$$\mathcal{L}_{nll}(\sigma, y) = -\Big(y \log \sigma + (1-y)\log(1-\sigma)\Big)$$

And we'll use $\frac{d}{dx}\log(x) = \frac{1}{x}$

$$\boxed{\frac{\partial \mathcal{L}_{NLL}}{\partial \sigma} = -\left(\frac{y}{\sigma} - \frac{1-y}{1-\sigma}\right)} \tag{14}$$

Now, we look at $\sigma(u)$:

$$\sigma(u) = \frac{1}{1+e^{-u}} \tag{15}$$

If we take the derivative, we can get:

$$\frac{\partial \sigma}{\partial u} = \frac{-e^{-u}}{\Big(1+e^{-u}\Big)^2} \tag{16}$$

Which we can rewrite, conveniently, as

Try this yourself if you're curious!

$$\boxed{\frac{\partial \sigma}{\partial u} = \sigma(1-\sigma)} \tag{17}$$

Finally, our last derivative:

$$u = \theta^T x + \theta_0 \tag{18}$$

$$\boxed{\frac{\partial u}{\partial \theta} = x} \tag{19}$$

Simplifying our chain rule

So, now, we can put together our chain rule:

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \theta} = \frac{\partial \mathcal{L}_{NLL}}{\partial \sigma} \cdot \frac{\partial \sigma}{\partial u} \cdot \frac{\partial u}{\partial \theta} \tag{20}$$

Plug in the derivatives:

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \theta} = -\left(\frac{y}{\sigma} - \frac{1-y}{1-\sigma}\right) \cdot \sigma(1-\sigma) \cdot x \tag{21}$$

Simplify:

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \theta} = \Big((1-y)\sigma - y(1-\sigma)\Big) \cdot x \tag{22}$$

And finally, we sum the terms. We can do the θ_0 gradient at the same time: the only difference is that $\frac{\partial u}{\partial \theta_0} = 1$, instead of x.

Key Equation 2

The **gradients** of NLL for gradient descent are

$$\nabla_\theta \mathcal{L}_{NLL} = (\sigma - y)x$$

$$\frac{\partial \mathcal{L}_{NLL}}{\partial \theta_0} = (\sigma - y)$$

We can plug this into J_{lr}:

One comment we didn't make: remember that $R(\theta)$ won't show up in the θ_0 derivative!

$$\nabla_\theta J_{lr} = \frac{1}{n}\sum_{i=1}^{n}\left(\left(g^{(i)} - y^{(i)}\right)x^{(i)}\right) + 2\lambda\theta \tag{23}$$

$$\frac{\partial J_{lr}}{\partial \theta_0} = \frac{1}{n}\sum_{i=1}^{n}\left(g^{(i)} - y^{(i)}\right) \tag{24}$$

We can use this to do **gradient descent**!

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta J_{lr}(\theta_{old}) \tag{25}$$

In $\theta^{(t)}$ notation:

$$\theta^{(t)} = \theta^{(t-1)} - \eta\left(\nabla_\theta J_{lr}(\theta^{(t-1)})\right) \tag{26}$$

$$\theta_0^{(t)} = \theta_0^{(t-1)} - \eta\left(\frac{\partial J_{lr}(\theta^{(t-1)})}{\partial \theta_0}\right) \tag{27}$$

This also corresponds to some basic math within Neural Networks, which we will return to **later** in the course.