Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

Fall 2022

Application to Regression

One nice thing about **gradient descent** is that it is **easy** to switch the kind of problem you're applying it to: all you need is your **parameters**(s) θ, and a function to optimize, J.

From there, you can just **compute** the gradient.

Ordinary Least Squares

Our **loss** function is

$$J(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^{n} \left((\theta^T x^{(i)} + \theta_0) - y^{(i)} \right)^2 \tag{1}$$

Or, in **matrix** terms,

Including the appended row of 1's from before.

$$J = \frac{1}{n} \left(\tilde{X}\theta - \tilde{Y} \right)^T \left(\tilde{X}\theta - \tilde{Y} \right)$$

Our gradient, according to **matrix derivative** rules, is

$$\nabla_\theta J(\theta) = \frac{2}{n} \tilde{X}^T \left(\tilde{X}\theta - \tilde{Y} \right) \tag{2}$$

Before, we set it equal to **zero**. But here, we can instead take **steps** towards the solution, using **gradient descent**.

We could use the **matrix** form, but sometimes it's easier to use a **sum**. Fortunately, derivatives are easy with a sum. If so, here's **another** way to write it:

$$\nabla_\theta J(\theta) = \frac{2}{n} \sum_{i=1}^{n} \left(\theta^T x^{(i)} - y^{(i)} \right) x^{(i)} \tag{3}$$

Either way, we use gradient descent **normally**:

Remember that θ_{old} is an **input** to the gradient, not multiplied by it!

$$\theta_{new} = \theta_{old} - \eta \nabla_\theta J(\theta_{old})$$

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

$$\theta^{(t)} = \theta^{(t-1)} - \eta \nabla_\theta J(\theta^{(t-1)})$$

Ridge Regression

Ridge regression is similar.

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\underbrace{(\theta^T x^{(i)} + \theta_0)}_{\text{guess}} - \underbrace{y^{(i)}}_{\text{answer}}\right)^2 + \underbrace{\lambda\|\theta\|^2}_{\text{Regularizer}}$$

However, we have to treat θ_0 as **separate** from our other data points, because of **regularization**: remember that it **doesn't** apply to θ_0.

For θ:

$$\nabla_\theta J_{\text{ridge}}(\theta, \theta_0) = \frac{2}{n}\sum_{i=1}^{n}\left((\theta^T x^{(i)} + \theta_0) - y^{(i)}\right)x^{(i)} + 2\lambda\theta \tag{4}$$

For θ_0:

$$\frac{\partial J_{\text{ridge}}(\theta, \theta_0)}{\partial \theta_0} = \frac{2}{n}\sum_{i=1}^{n}\left((\theta^T x^{(i)} + \theta_0) - y^{(i)}\right) \tag{5}$$

Notice that we used a **gradient** for our vector θ, but since θ_0 is a single variable, we just used a **simple derivative**!

Concept 1

The **gradient** $\frac{dJ}{d\theta}$ must have the **same shape as** θ: this shape-matching is why we can easily **subtract** it during gradient descent.

$$\underbrace{\theta_{new}}_{(d\times 1)} = \underbrace{\theta_{old}}_{(d\times 1)} - \eta \underbrace{\nabla_\theta J(\theta_{old})}_{(d\times 1)}$$