Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

Fall 2022

Solving the OLS Problem

Optimization in 1-D - Calculus Returns!

Now that we have our **problem** presented the way we **want**, we can figure out how to **optimize** our θ.

For now, we'll revert to **sum** notation, but we'll come back to our **matrices** later.

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

How do we **optimize** this? Let's just take **one data point**:

$$J(\theta) = (\theta^T x - y)^2 \tag{2}$$

And we'll start in 1D.

$$J(\theta) = (\theta x - y)^2 \tag{3}$$

If we treat θ like any ordinary **variable**, this is just a simple function! How would we find the **minimum**?

Using **calculus**! Anywhere there's a local **minimum**, we typically know the **derivative** is 0.

Assuming a "smooth" surface...

Note that we aren't taking $\frac{d}{dx}$: we want to change our **model**, not our **data**! So, since θ represents our **model**, we'll take $\frac{d}{d\theta}$.

$$J'(\theta) = 2x(\theta x - y) = 0 \tag{4}$$

We just find where the slope is 0, and solve for θ!

We technically need to prove whether this is minimum, maximum, or neither. For now, we'll assume we have a minimum.

$$\theta^* = \frac{y}{x} \tag{5}$$

Concept 1

If our function $J(\theta)$ has **one variable**, we can **explicitly** find **local minima** by solving for θ when the **derivative** is zero.

$$\frac{dJ}{d\theta} = 0$$

Then, you check each candidate using the second derivative to see if it is a minimum ($J''(\theta) > 0$). In this class we will often be able to ignore this step.

This concept is review from 18.01 (Single-variable calculus), but is worth repeating!

Using our sum

Now, we'll go back to having multiple data points we want to **average**:

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta x^{(i)} - y^{(i)}\right)^2 \tag{6}$$

We want to do the same optimization. Thankfully, derivatives are **linear**: addition and scalar multiplication are not affected!

Check the prerequisites chapter, chapter 0, for a full definition of linearity.

$$J'(\theta) = \frac{1}{n}\sum_{i=1}^{n} 2x^{(i)} \cdot (\theta x^{(i)} - y^{(i)}) = 0 \tag{7}$$

And we can **solve** this just the same way.

Optimizing for multiple variables

Now, the tricky part is working with **vectors**.

We'll ignore the averaging and $^{(i)}$ notation since that's easy to add on afterwards.

$$J(\theta) = (\theta^T x - y)^2 \tag{8}$$

We want to **optimize** this. In the **one-dimensional** case, we wanted to set the **derivative** of J to **zero**, using a single θ variable. Now, we have **multiple** variables θ_k to **change**.

Derivatives are all about **change** in variables, and our **change** $\Delta\theta$ is a **combination** of changing the different **components**, $\Delta\theta_k$.

$$\Delta\theta = \begin{bmatrix} \Delta\theta_0 \\ \Delta\theta_1 \\ \Delta\theta_2 \\ \vdots \\ \Delta\theta_d \end{bmatrix} \tag{9}$$

So, maybe it would be reasonable to just set **every** derivative to **zero**? It turns out, the answer is **yes**!

We can show this by using the **chain rule** definition:

$$\overbrace{\Delta\theta \frac{dJ}{d\theta}}^{\text{The change in } J \text{ from } \theta \text{ overall}} \approx \overbrace{\Delta\theta_0 \frac{\partial J}{\partial\theta_0} + \Delta\theta_1 \frac{\partial J}{\partial\theta_1} + \cdots + \Delta\theta_d \frac{\partial J}{\partial\theta_d}}^{\text{The change in } J \text{ from each } \theta_k \text{ term}} \tag{10}$$

So, if all the derivatives are zero, the **overall** derivative is zero.

This approximation formula becomes exact as the step size shrinks: we go from $\Delta\theta$ to $d\theta$.

Concept 2

If our function $J(\theta)$ has d **different variables** , we can **explicitly** find **local minima** by getting all of the **equations**

$$\frac{\partial J}{\partial \theta_0} = 0, \quad \frac{\partial J}{\partial \theta_1} = 0, \quad \frac{\partial J}{\partial \theta_2} = 0 ... \quad \frac{\partial J}{\partial \theta_d} = 0$$

Or in general,

$$\frac{\partial J}{\partial \theta_k} = 0 \quad \text{for all k in } \{0, 1, 2, ..., d\}$$

...And solving this **system of equations** for the **components** of θ.

The **solution** to this system of equations will be our **desired result**, θ^*.

Again, we ignore the second requirement of making sure this isn't a **maximum** or **saddle** point.

Gradient Notation

Writing it this way can be a **hassle**. So, we'll continue our tradition of using **matrix-based** notation to make our lives easier.

You may recognize these **component-wise** derivatives as part of the "**multivariable** version" of the derivative: the **gradient**.

Key Equation 3

The **gradient** of J with respect to θ is

$$\nabla_\theta J = \frac{\partial J}{\partial \theta} = \begin{bmatrix} \partial J / \partial \theta_0 \\ \vdots \\ \partial J / \partial \theta_d \end{bmatrix} \tag{11}$$

Note the subscript on the gradient! This emphasizes that our **space** is the components of θ, not the components of our data x.

For example, our previous approach boiled down to saying

$$\nabla_\theta J = \begin{bmatrix} \partial J / \partial \theta_0 \\ \vdots \\ \partial J / \partial \theta_d \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} = 0 \tag{12}$$

For our purposes, we will simply **represent** that **zero vector** with a single 0.

Concept 4

If our function $J(\theta)$ has a **vector variable**, we can **explicitly** find **local minima** by solving for θ when the **gradient** is **0**.

$$\nabla_\theta J = 0$$

This is the form we will be solving.

Ignoring the requirement from earlier! We're assuming it's a minimum.

Matrix Calculus

Taking derivatives of vectors falls under **vector calculus**. That would solve our **above** problem.

But, before, we showed that it's more **convenient** if we can store these instead as **matrices**: that way, we don't need the **sum**. Luckily, we can generalize our work with **matrix calculus**.

Below, we will use the alternative notation, to be consistent with the official notes.

$$J = \frac{1}{n}\left(\tilde{X}\theta - \tilde{Y}\right)^T\left(\tilde{X}\theta - \tilde{Y}\right) \tag{13}$$

We will not show here how to **find** these derivatives, but the important rules you need to compute our derivatives are in the **appendix**.

There's a document explaining vector derivatives coming soon!

Note that **matrix** derivatives often look **similar** to **traditional** derivatives, but they are **not the same**. Most often, making this mistake will result in **shape errors**.

Sometimes, you can guess a derivative by using the familiar rules and fixing shape errors with transposing/changing multiplication order. But be careful!

When we take our derivative, we get

$$\nabla_\theta J = \frac{2}{n}\tilde{X}^T\left(\tilde{X}\theta - \tilde{Y}\right) = 0 \tag{14}$$

From here, we just solve for θ just like in the official notes.

Key Equation 5

The **solution** for **OLS optimization** is

$$\theta = \underbrace{\left(\tilde{X}^T\tilde{X}\right)^{-1}}_{d\times d}\underbrace{\tilde{X}^T}_{d\times n}\underbrace{\tilde{Y}}_{n\times 1}$$

Or, in our **original** notation,

$$\theta = \underbrace{\left(XX^T\right)^{-1}}_{d\times d}\underbrace{X}_{d\times n}\underbrace{Y^T}_{n\times 1}$$

And we're done with OLS!