Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

Fall 2022

Using gradient descent: minimizing μ

We can also use **gradient descent** to solve this problem!

We want to **minimize** our loss $\mathcal{L}$, and we do this by **adjusting** our cluster means $\mu^{(j)}$ until they're in the best position.

Concept 1

We can solve the **k-means problem** using **gradient descent**!

So, we want to **optimize** $\mathcal{L}$ using μ:

$$\mathcal{L}(\mu) = \sum_{i=1}^{n} \mathbb{1}(y^{(i)} = j) \left\| x^{(i)} - \right\|^2 \tag{1}$$

Rather than dealing with the indicator function $1(\cdot)$, we could instead just consider whichever μ is closest: **minimum** distance.

$$\overbrace{\min_j}^{\text{Minimizing}} \overbrace{\left\| x^{(i)} - \right\|^2}^{\text{distance}} \tag{2}$$

This **automatically** assigns every point to the closest **cluster** before we get our loss! So, all we need to worry about is μ_j.

Notation 2

Instead of using an **indicator function**, we can represent **cluster assignment** another way: using the **function** $\min_j$.

It can give **minimum distance** from $x^{(i)}$ to one of the cluster means: it picks the **closest** mean.

This **automatically** assigns the point to the **closest** cluster, making our job easier.

$$\mathcal{L}(\mu) = \sum_{i=1}^{n} \overbrace{\min_j}^{\text{Nearest cluster}} \left\| x^{(i)} - \right\|^2 \tag{3}$$

Now, we can do gradient descent using $\frac{\partial L(\mu)}{\partial \mu}$.

We move our means until they're minima!

$L(\mu)$ is **mostly** smooth, except when the cluster assignment of a **point** changes. So, it's usually smooth **enough** to do gradient descent.

Getting labels

Once we've finished gradient descent, and we've minimized our loss, we can get our **labels**.

min gives the **output** value that we get by minimizing. In this case, average **squared distance** from the cluster mean.

Meanwhile, arg min gives us the **input** value that gives us the minimum output. In this case, the **cluster** that gives the minimum distance.

So, arg min gives us the cluster closest to each point: that's our **label**!

We can use this notation to get our **labels**.

Notation 3

After **optimizing** μ, our **labels** are given by:

$$y^{(i)} = \arg\min_j \left\| x^{(i)} - \mu^{(j)} \right\|^2 \tag{4}$$

Using gradient descent can give us a **local** minimum, but our surface is not fully **convex**: so, we don't necessarily get a **global** minimum.

Even though individual terms of squared distance may be convex, adding min terms may not be convex.