Explanatory Notes for 6.390

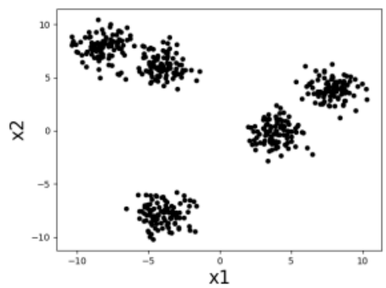
Shaunticlair Ruiz (Current TA)

Fall 2022

Initializing the k-means algorithm

Now that we have our **clusters**, **means**, and a **loss** function for evaluating them, we can begin looking for a better **clustering**.

We'll start out with a **dataset** we want to cluster: we'll use the one from the **beginning** of the chapter:

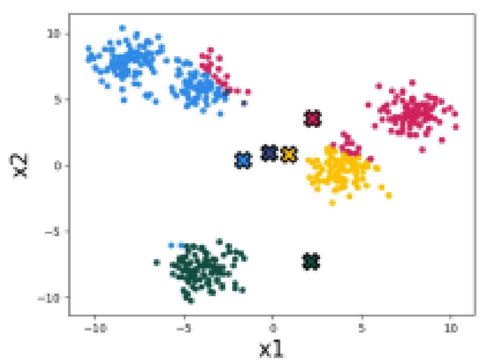


We could cluster this visually, but we want our machine to be able to do it for us.

First, we need to decide on our **number** of clusters. When you can't **visualize** it, this can be **difficult** - how many is too many or too few?

But, for now, we'll **ignore** that problem, and say $k = 5$.

Let's **randomly** assign our initial cluster means, and assign each point to the **closest** cluster:



1) Initial assignments

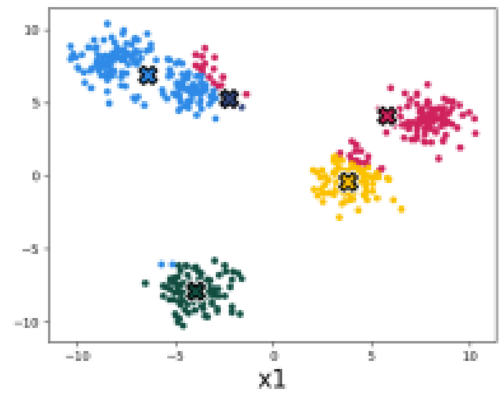
This is our starting point for the algorithm.

First step: moving our cluster means

As we mentioned before, these points aren't **actually** the average of their cluster: you can tell that by looking at it.

We want to **minimize** the variation in our cluster: that's why we're using the mean.

So, let's fix this: we'll take the **average** of all the points in each **cluster**, and **move** the cluster mean to that position.



2) Update means

And now, our cluster means are closer to all our data points!

Concept 1

One way **minimize** the **distance** between the **cluster mean** and its **data points** is:

- Take the **average** of all the points in the cluster, and **reassign** the cluster mean to that average.

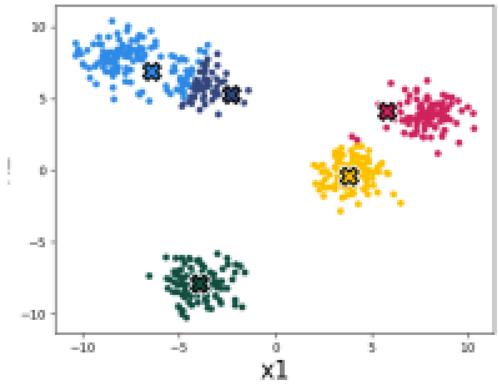
Second step: Reassign data points

We've **improved** our model by moving the cluster mean.

The problem is, we originally **assigned** every point to the **closest** cluster mean.

If the cluster means **move**, then some points might be closer to a **different** cluster now!

If so, we can **improve** our clustering further by reassigning points to the cluster they're **closest** to!



3) Update assignments

Concept 2

Another way **minimize** the **distance** between the **cluster mean** and its **data points** is:

- After the **means** have been **moved**, **reassign** the **data points** to whichever mean is **closest**.

The cycle continues

But wait - now that we've changed the points in each cluster, our cluster mean might not be the **true** average!

So, we can, again, improve our loss by taking the average of each cluster, and moving the cluster mean.

This creates a cycle that continues until we **converge** on our final answer.

Concept 3

Together, of our steps for **improving** our clusters create a **cycle** of **optimization**:

- **Moving** our cluster mean **changes** which point should go in each cluster.
- **Reassigning** points to different clusters **changes** our cluster mean.

The k-means algorithm

These two steps make up the **bulk** of our algorithm:

Definition 4

The k-**means algorithm** uses the following steps:

- First, we **randomly** choose our **initial** cluster means.

Then, we **cycle** through the following two steps:

- **Reassign points** to the cluster mean they're closest to.
- **Move** each **cluster mean** to the average of all the points in that cluster.

Until our clusters means **stop** changing.

When we run our algorithm on the above dataset, we get:

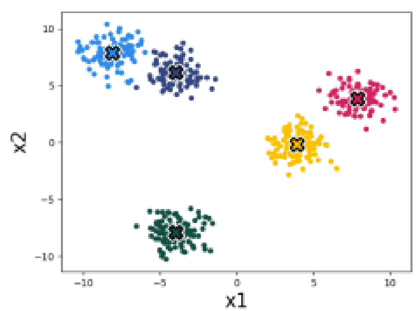


Figure 6.3: Converged result.

Note that, our cycle **works** because changing **either** cluster mean or point assignments allows you to further **improve** the **other** step.

So, if we're **not** changing one of them, the other one won't **change** either: the cycle is **broken**, and we can **stop**.

This is our termination condition!

Another nice fact: it can be shown that this algorithm does **converge** to a local minimum!

Concept 5

The k-**means algorithm** is guaranteed to **converge** to a **local minimum**.

Pseudocode

K-MEANS$(k, \tau, \{x^{(i)}\}_{i=1}^{n})$

$\mu, y =$ randinit #Random initialization
for $t = 1$ **to** τ #Begin cycling

  $y_{\texttt{old}} = y$ #Keep track of last step

  for $i = 1$ **to** n
    $y^{(i)} = \arg\min_j \left\|x^{(i)} - \mu^{(j)}\right\|_2^2$ #Reassign data point to closest mean

  for $j = 1$ **to** k
    $\mu^{(j)} = \frac{1}{N_j} \sum_{i=1}^{n} \mathbb{1}(y^{(i)} = j) x^{(i)}$ #Move cluster mean to average of cluster

  if $\mathbb{1}(y = y_{\texttt{old}})$
    break #If nothing has changed, then the cycle is done. Terminate

return μ, y