CSE/STAT 416

K-Means Clustering

Pemi Nguyen
University of Washington
May 2, 2022

Slides by Hunter Schafer

Recap

For the past 5 weeks, we have covered different **supervised learning** algorithms

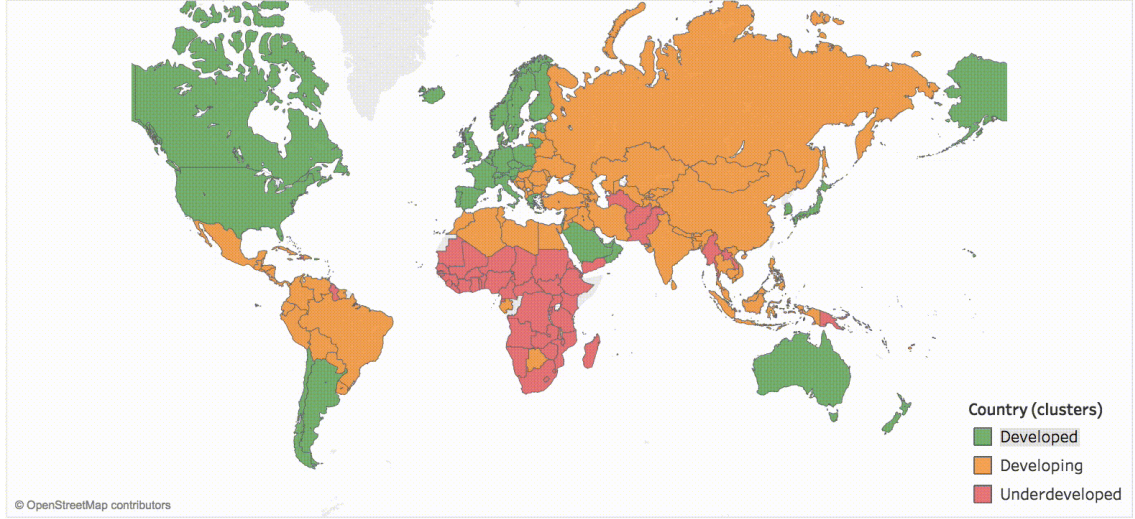
Now, we're going to explore **unsupervised learning** methods where don't have labels / outputs in your datasets anymore

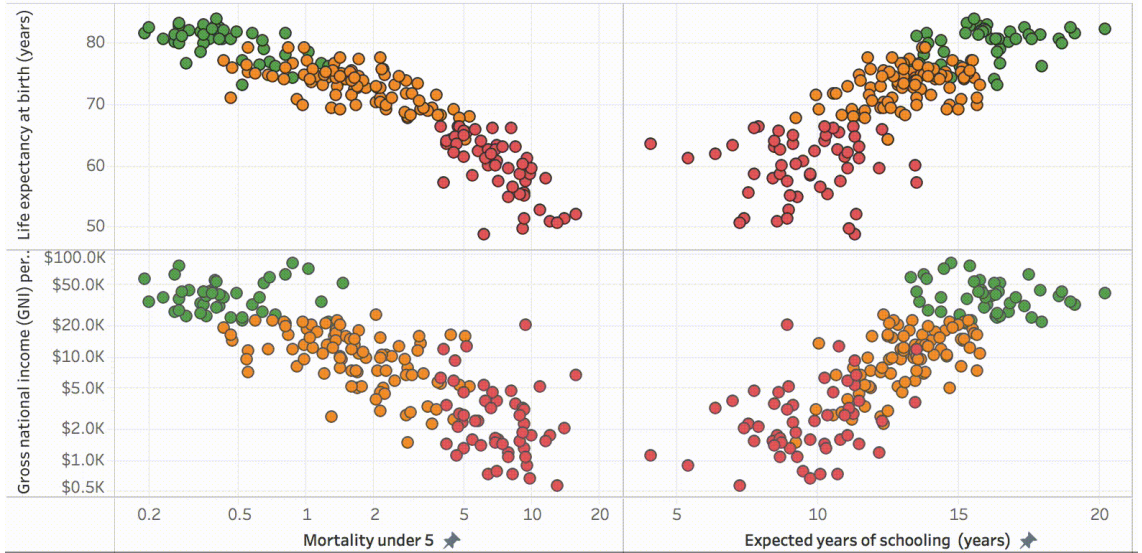
Clustering

SPORTS

WORLD NEWS

Clustering

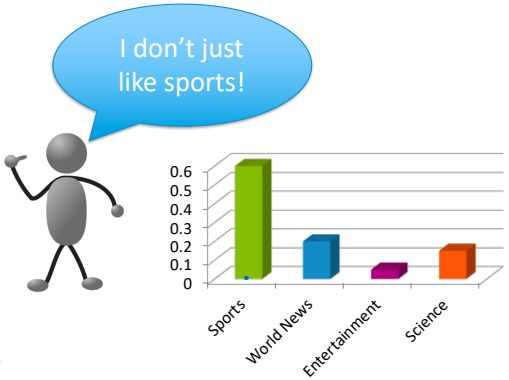

Country (clusters)
Developed
Developing
Underdeveloped
© OpenStreetMap contributors

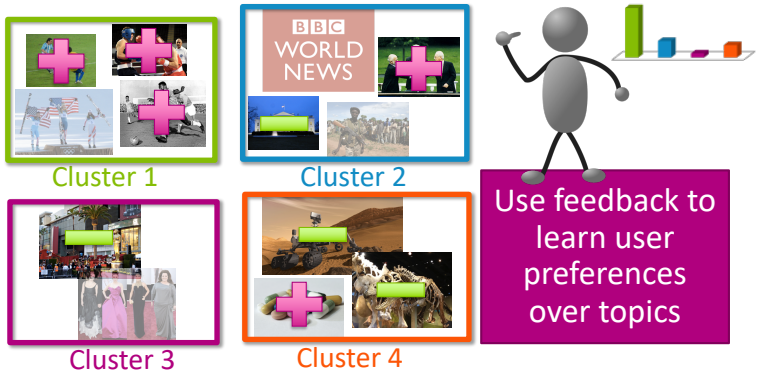

Life expectancy at birth (years)
80
70
60
50
Gross national income (GNI) per...
$100.0K
$50.0K
$20.0K
$10.0K
$5.0K
$2.0K
$1.0K
$0.5K
0.2
0.5
1
2
5
10
20
Mortality under 5
5
10
15
20
Expected years of schooling (years)

Recommending News

User preferences are important to learn, but can be challenging to do in practice.

- People have complicated preferences
- Topics aren't always clearly defined

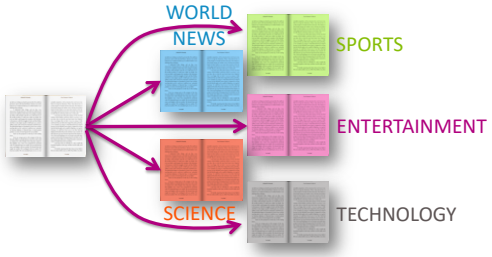




Labeled Data

What if the labels are known? Given labeled training data

Can do multi-class classification methods to predict label



Example of supervised learning

Unsupervised Learning

In many real world contexts, there aren't clearly defined labels so we won't be able to do classification

We will need to come up with methods that uncover structure from the (unlabeled) input data X.

Clustering is an automatic process of trying to find related groups within the given dataset.

$Input: x^{(1)}, x^{(2)}, \ldots, x^{(n)}$

$Output: y^{(1)}, y^{(2)}, \ldots, y^{(n)}$

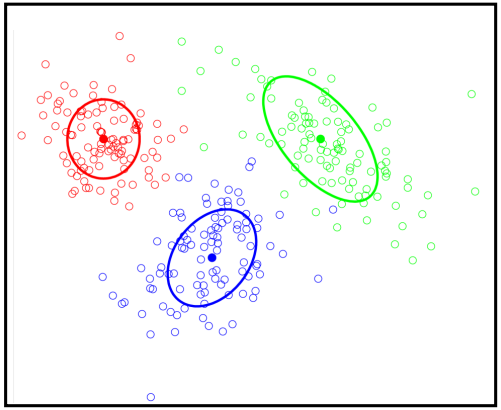
Define Clusters

In their simplest form, a **cluster** is defined by

- The location of its center (**centroid**)
- Shape and size of its **spread**

Clustering is the process of finding these clusters and **assigning** each example to a particular cluster.

- $x^{(i)}$ gets assigned $y^{(i)} \in [1, 2, \ldots, k]$
- Usually based on closest centroid



Will define some kind of objective function for a clustering that determines how good the assignments are

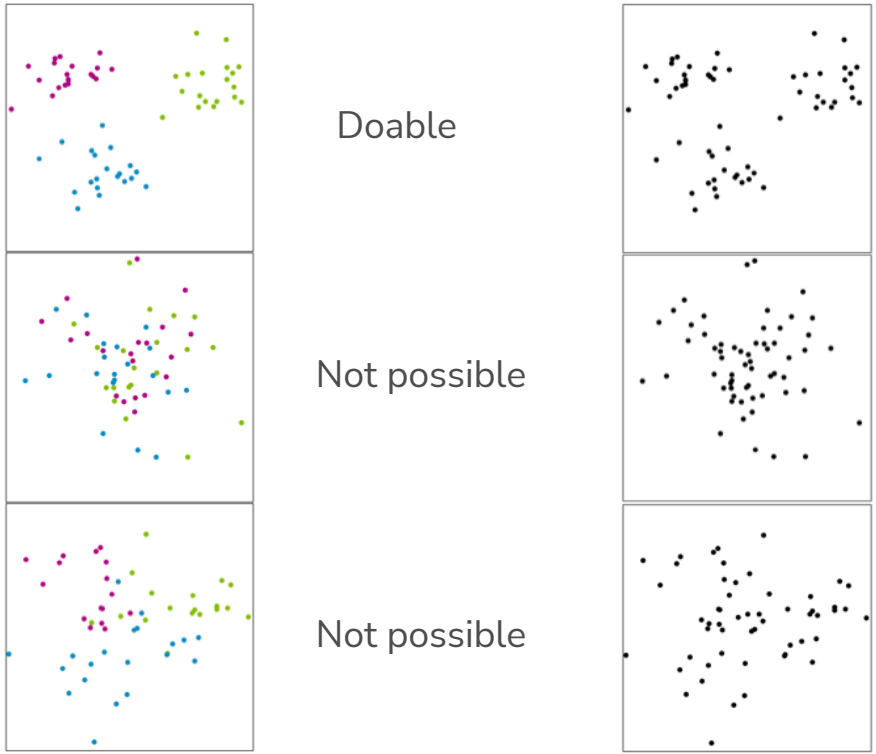
- Based on distance of assigned examples to each cluster.
- Close distance reflects strong similarity between datapoints.

When does this works for k means?

Clustering is easy when distance captures the clusters.

Ground Truth (not visible)

Given Data



Doable

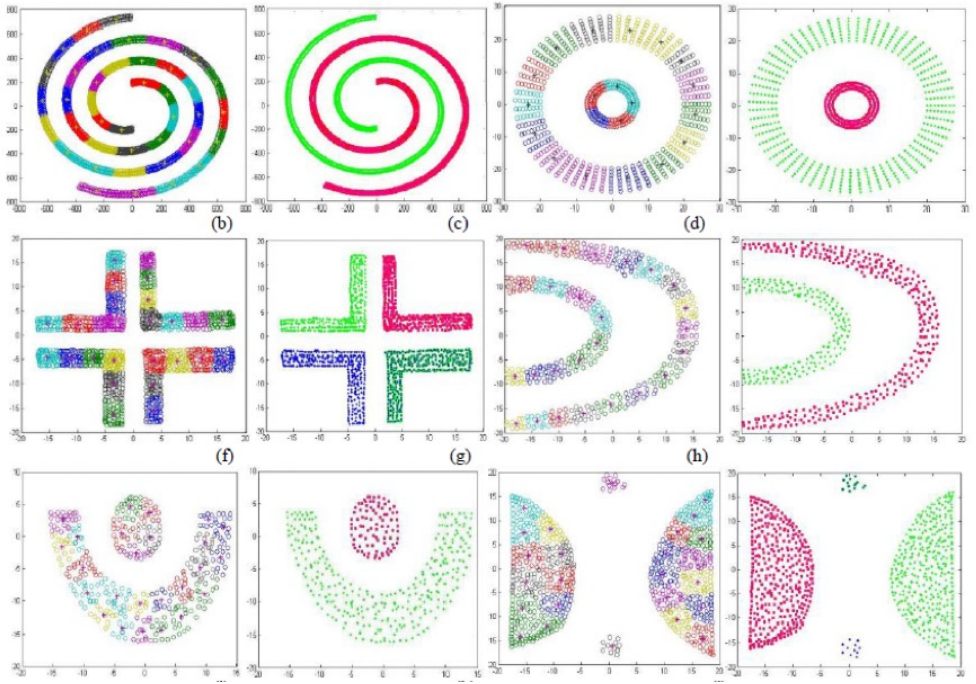
Not possible

Not possible

Clustering is not always straight-forward

There are many clusters that are harder to learn with this setup

Distance does not determine clusters



K-Means Clustering

K-Means Clustering Algorithm

We define the criterion of assigning point to a cluster based on **its distance.**

Shorter distance => Better Clustering

Algorithm
Given a dataset of n datapoints and a particular choice of k

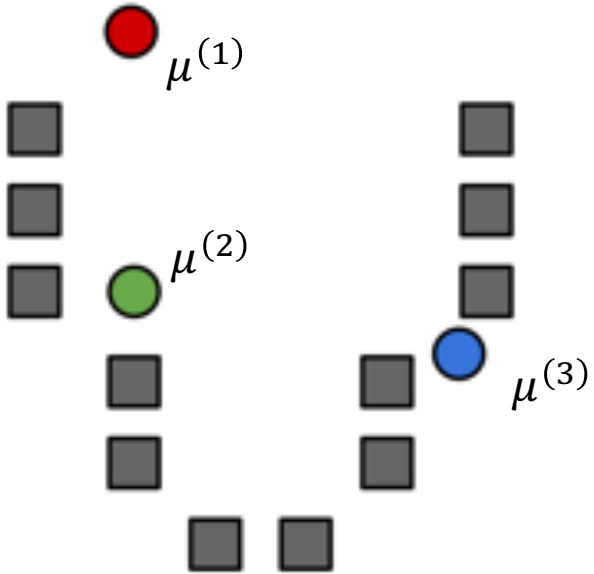
Step 0: Initialize cluster centroids randomly

Repeat until convergence:
- Step 1: Assign each example to its closest cluster centroid
- Step 2: Update the centroids to be the average of all the points assigned to that cluster

Step 0

Start by choosing the initial cluster centroids

- A common default choice is to choose centroids $\mu^{(1)}, \ldots, \mu^{(k)}$ randomly
- Will see later that there are smarter ways of initializing

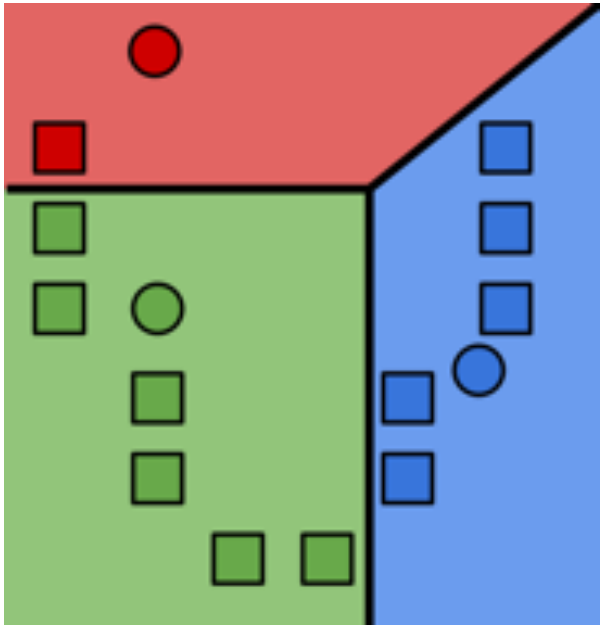


Step 1

Assign each example to its closest cluster centroid

For i = 1 to n

$$y^{(i)} \leftarrow \underset{j}{\operatorname{argmin}} \left|\left|\mu^{(j)} - x^{(i)}\right|\right|_2^2$$



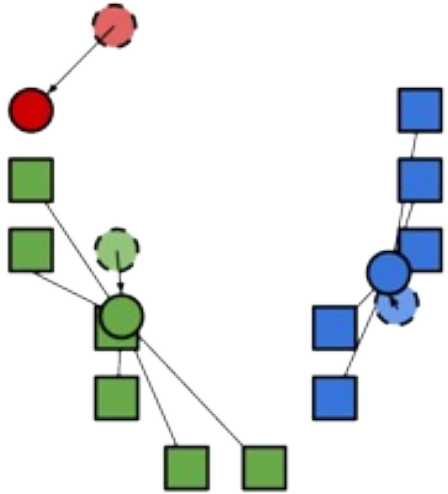
Step 2

Update the centroids to be the mean of all the points assigned to that cluster.

For j = 1 to k

$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}\, x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

Computes center of mass for cluster!

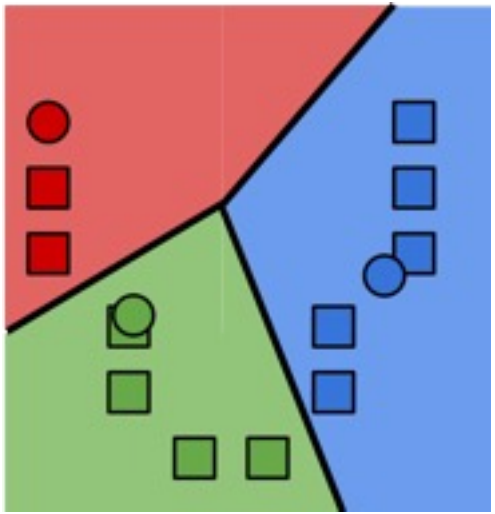


Repeat until convergence

Repeat Steps 1 and 2 until convergence

Stopping conditions

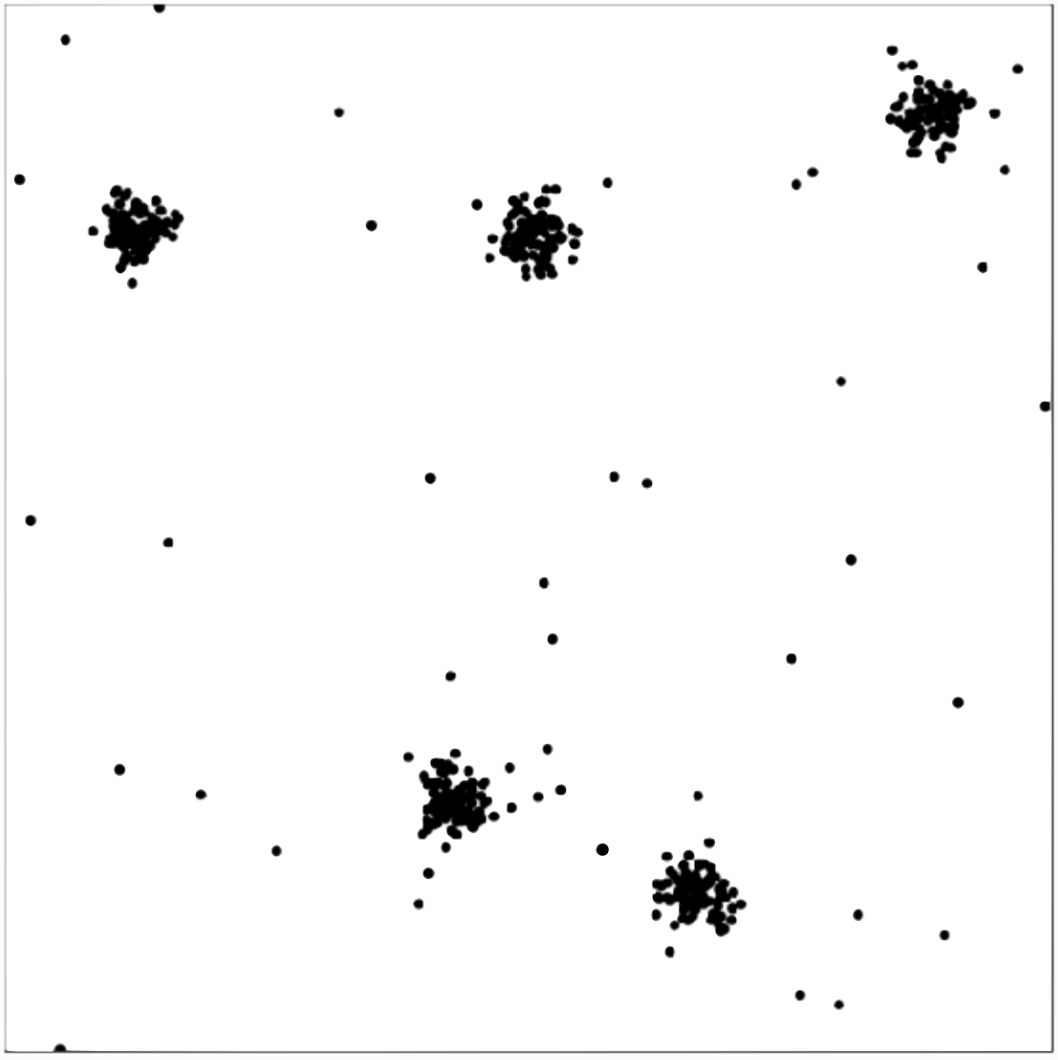
- Cluster assignments haven't changed
- Centroids haven't changed
- Some number of max iterations have been passed

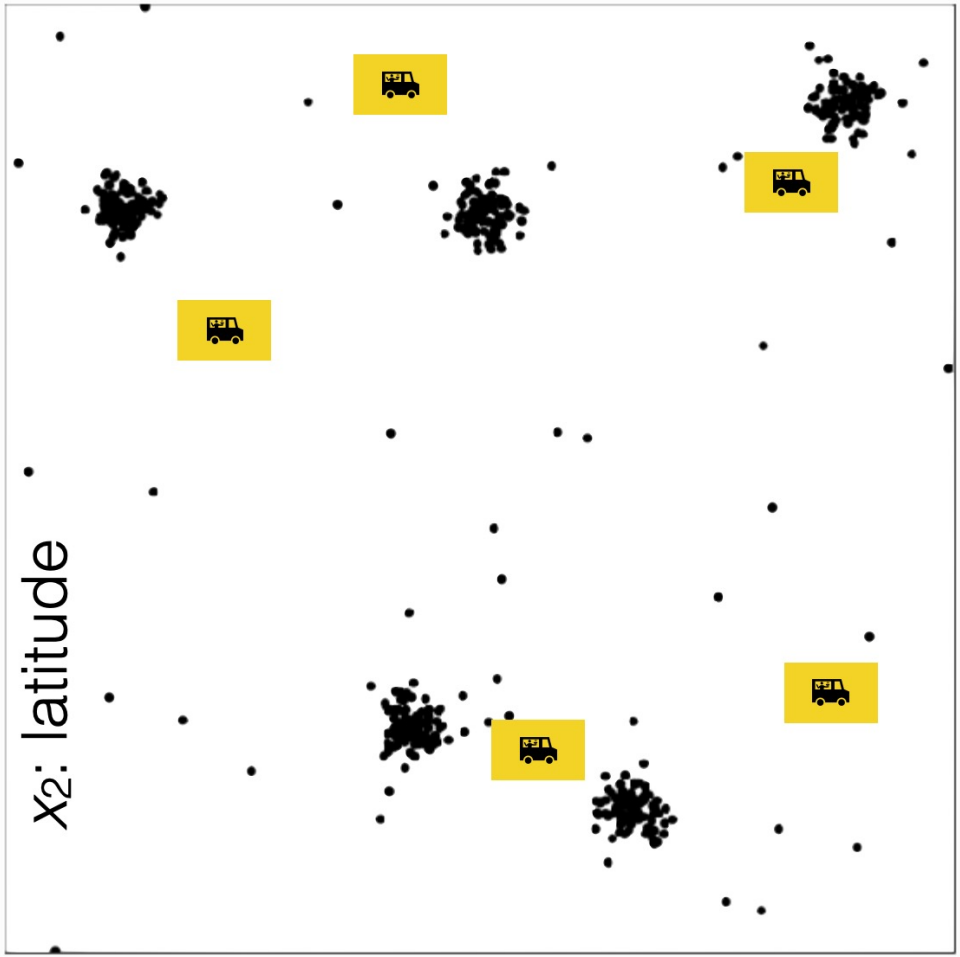


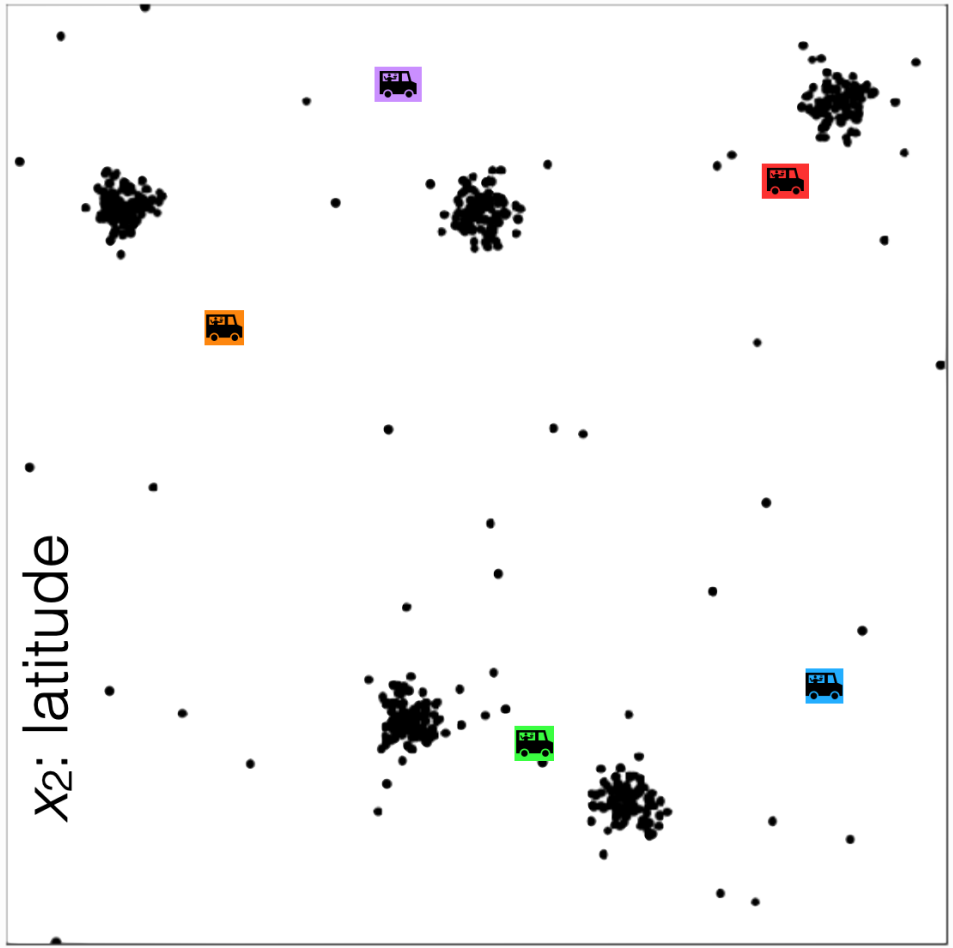
What will it converge to?

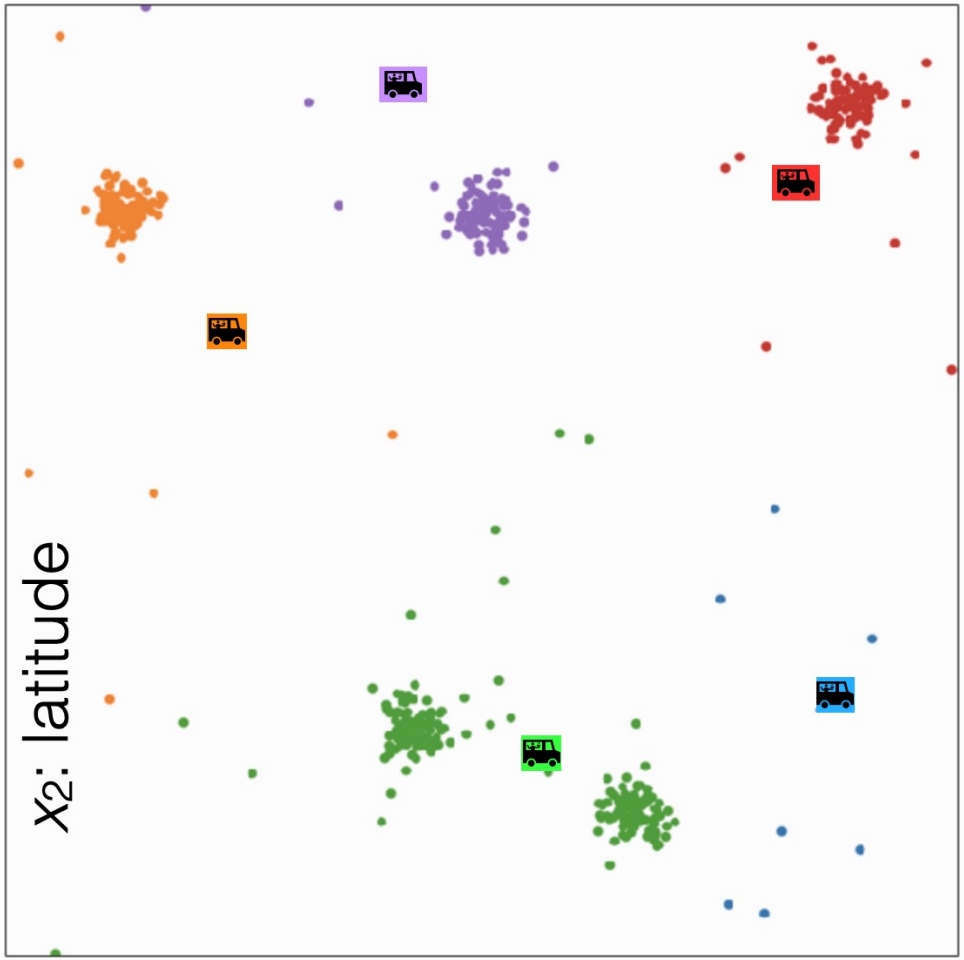
- Local Optima
- Global optima
- Neither

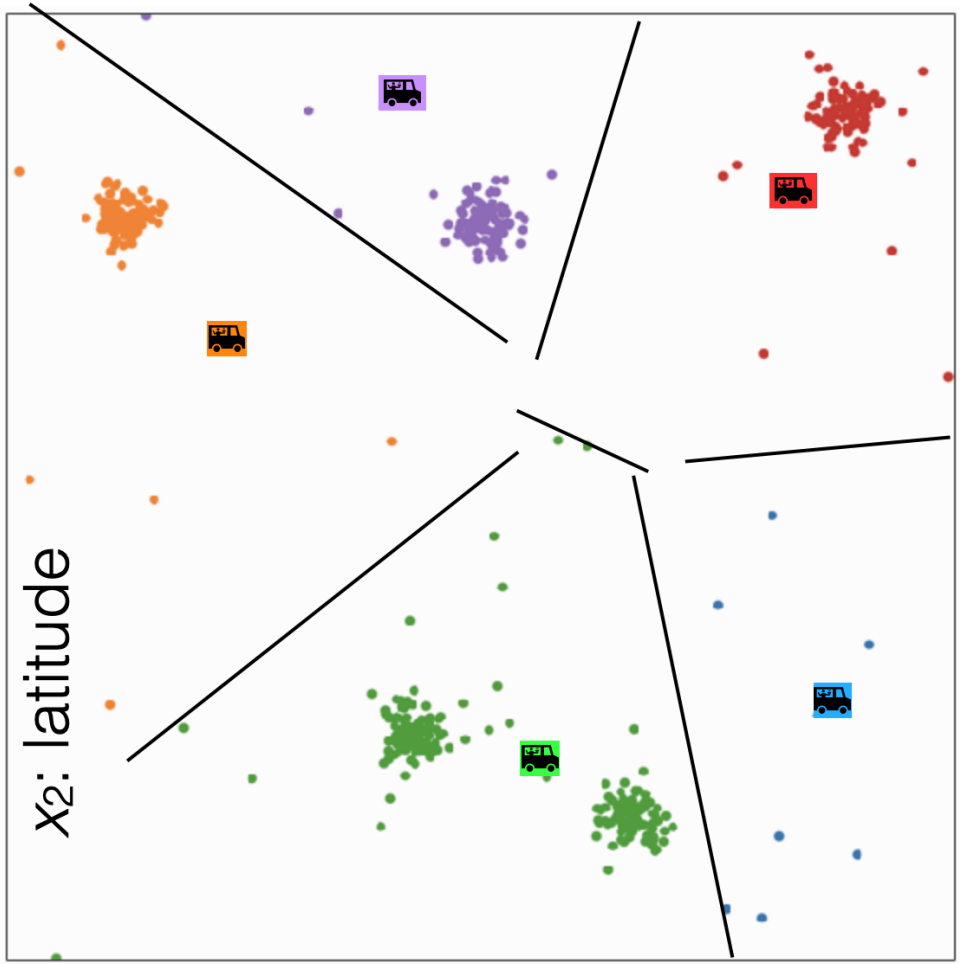
Visualization

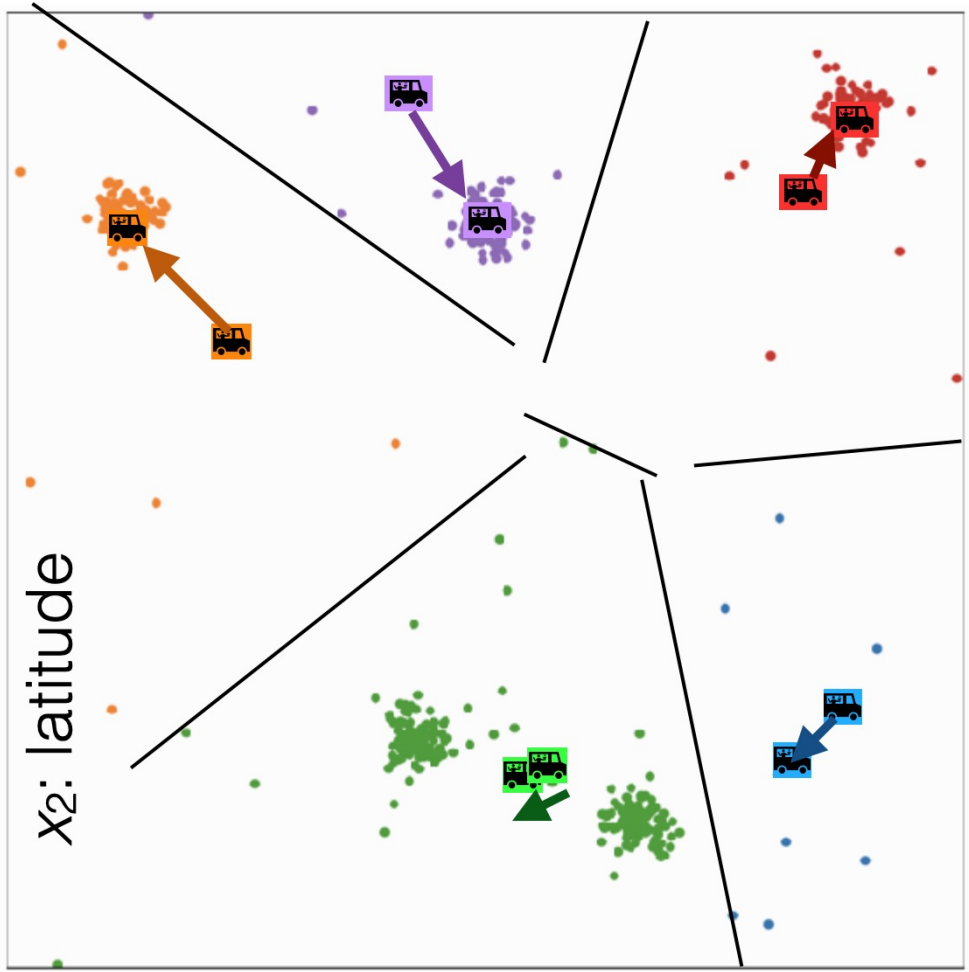


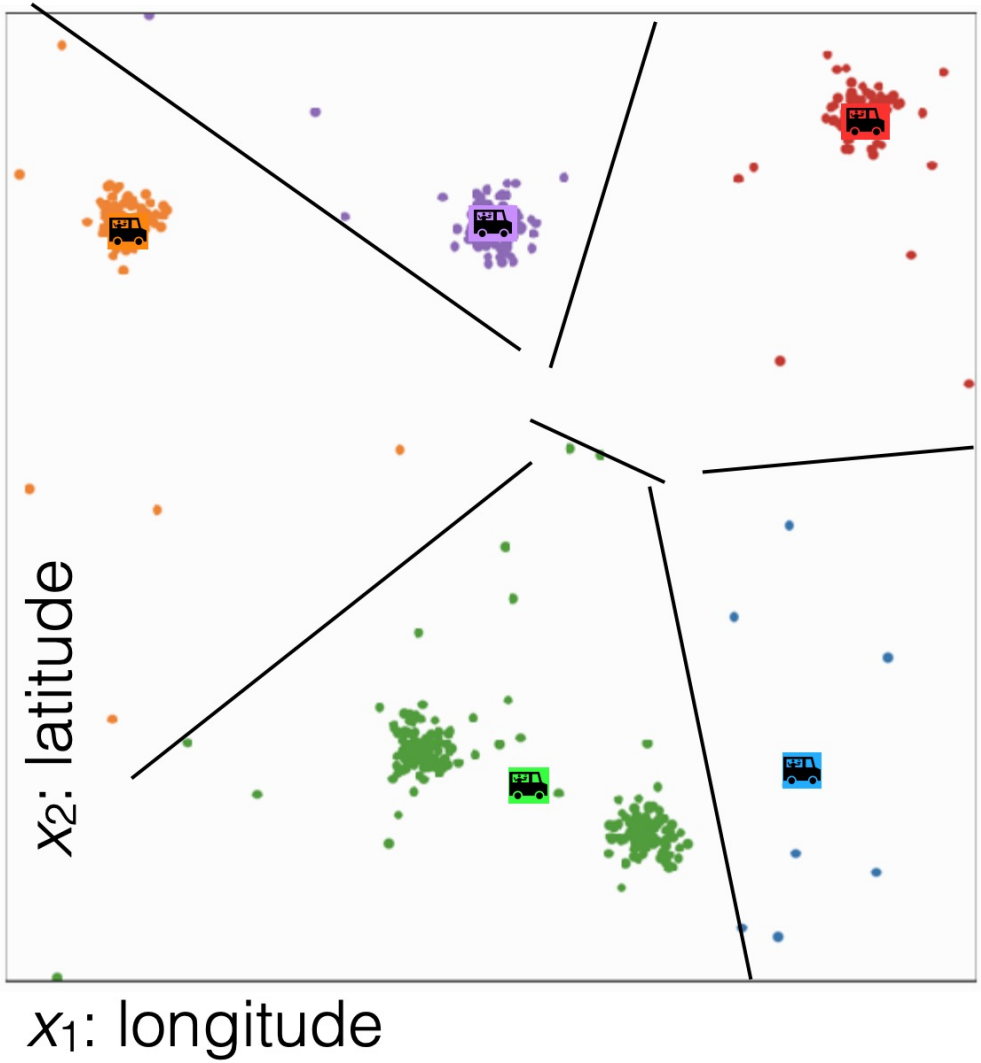

x_2: latitude
x_1: longitude

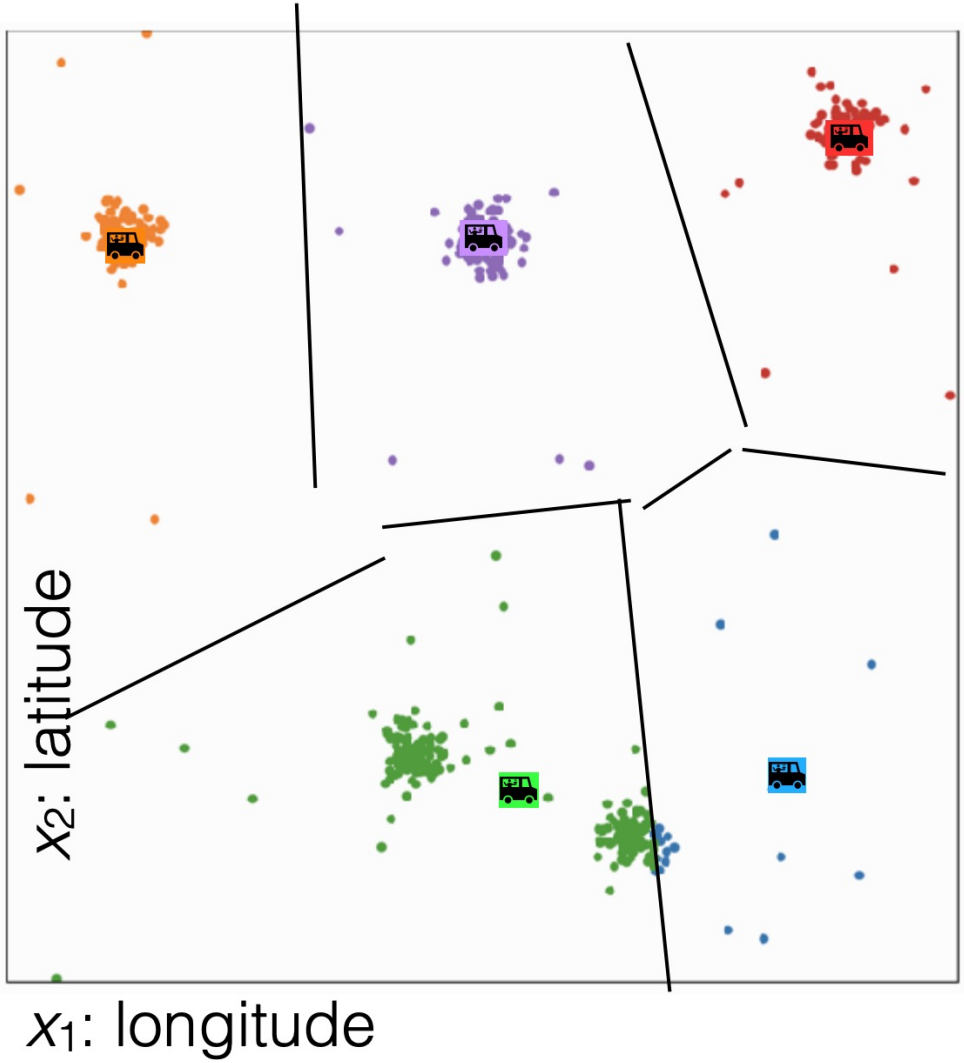

x_2: latitude
x_1: longitude

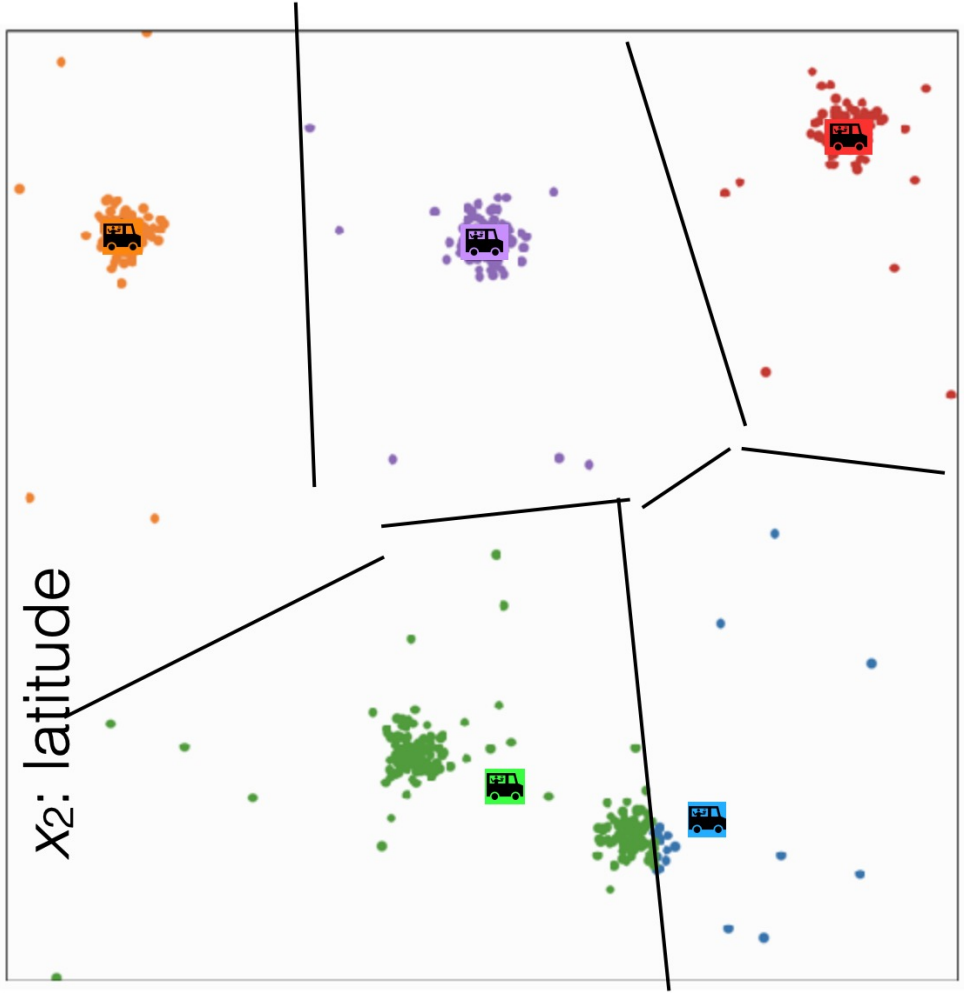

x_2: latitude
x_1: longitude

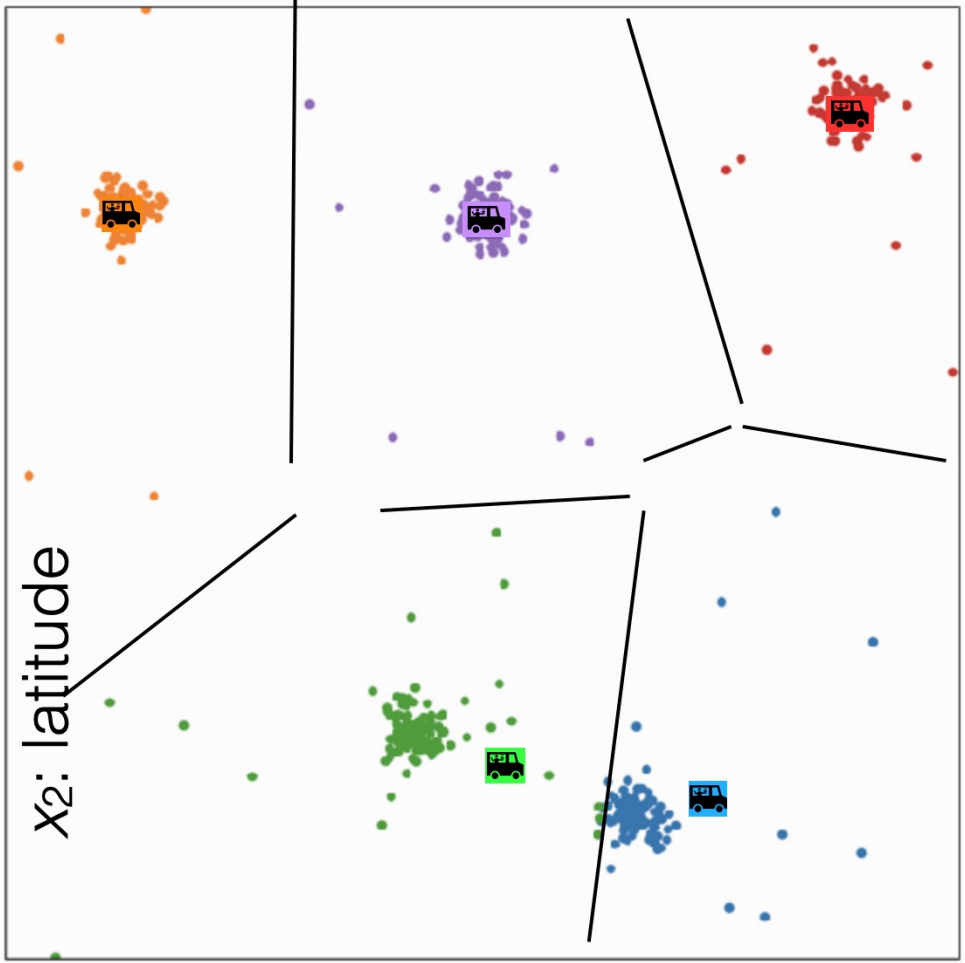

x_2: latitude
x_1: longitude

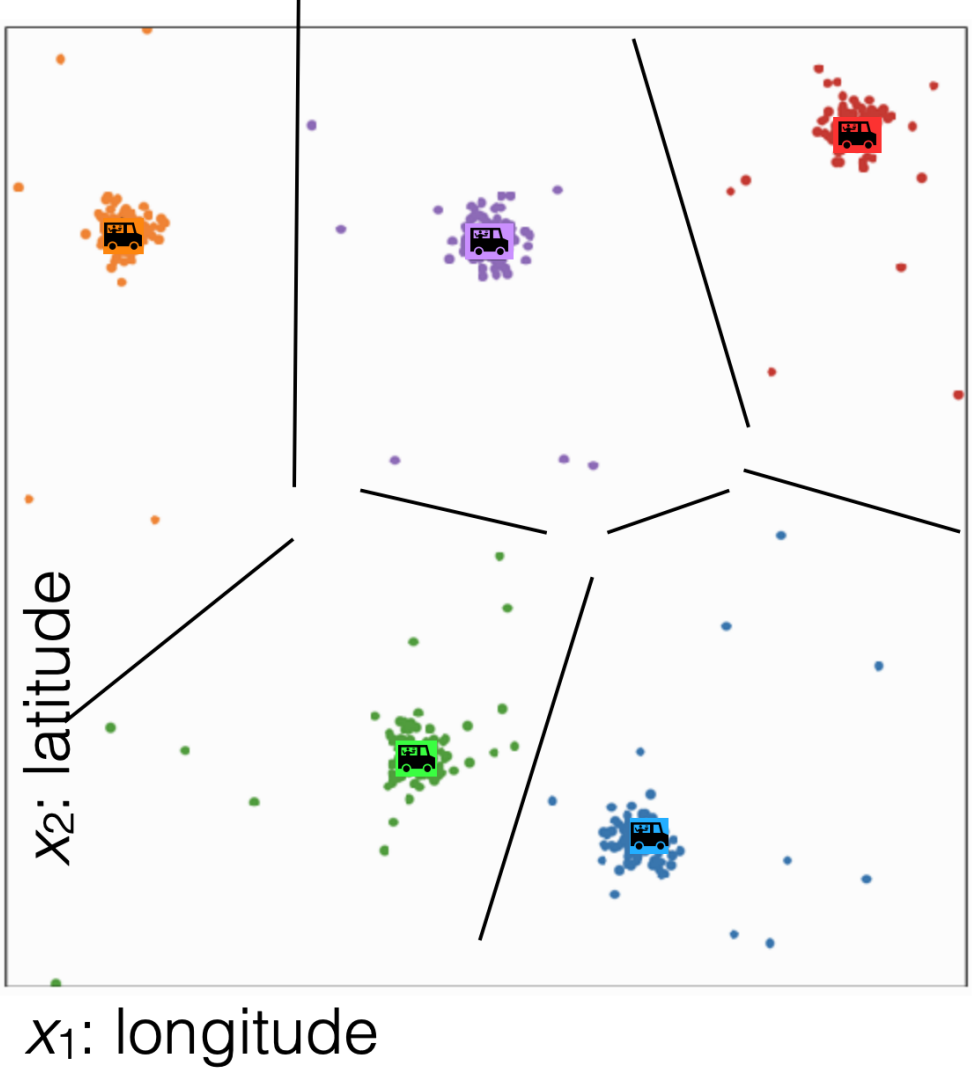

x_2: latitude
x_1: longitude

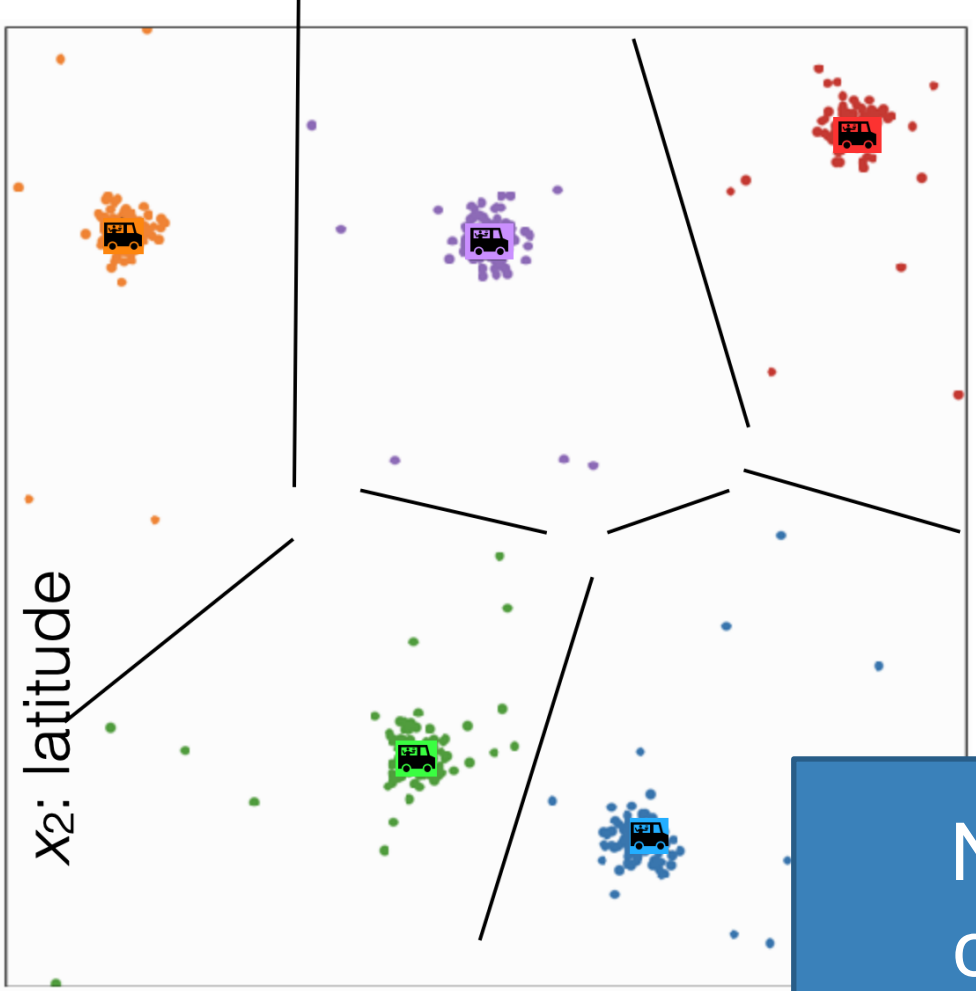

x_2: latitude
x_1: longitude


x_2: latitude
x_1: longitude


x_2: latitude
x_1: longitude

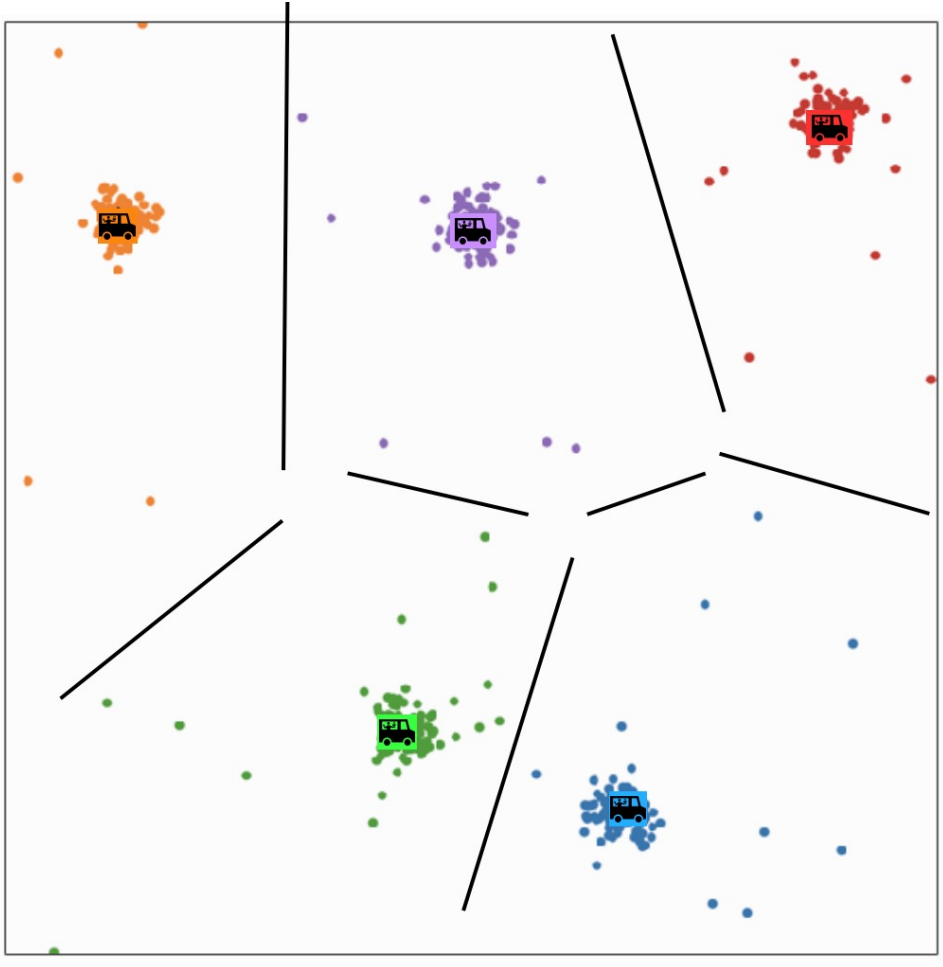

x_2: latitude
x_1: longitude

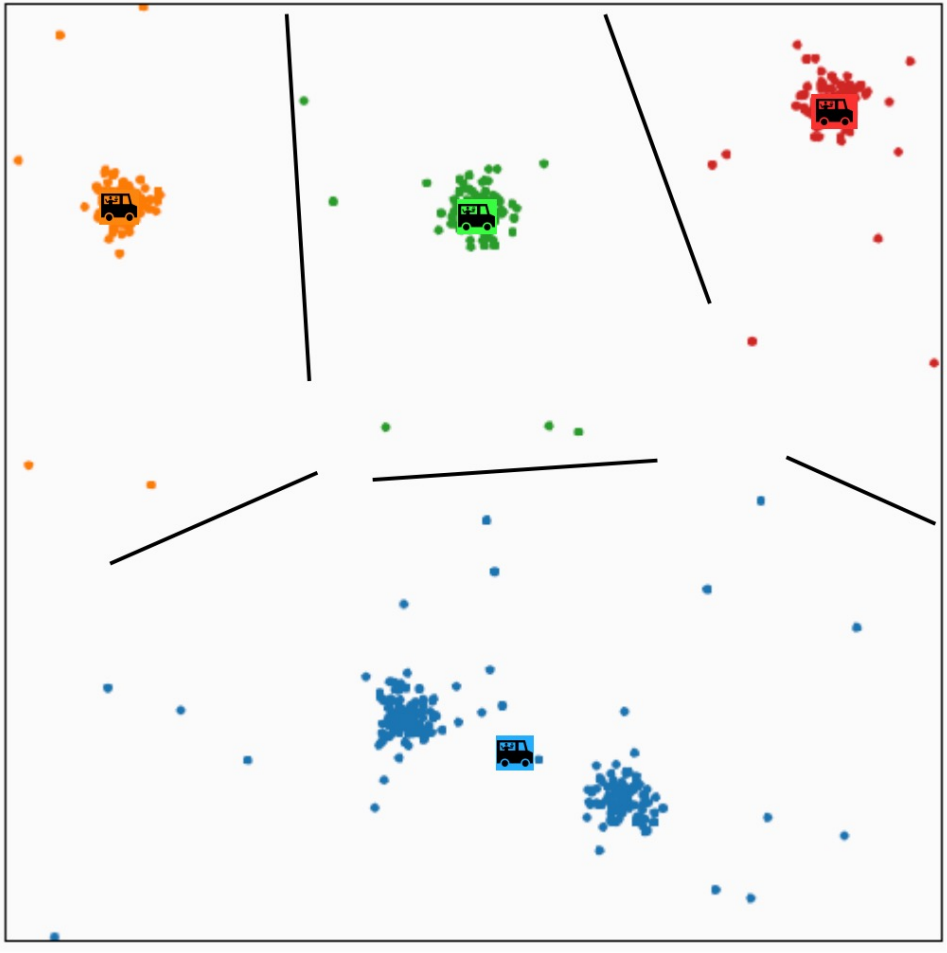

x_2: latitude
x_1: longitude



No more changes

Different k gives different results





Clustering vs Classification

Clustering looks like we assigned labels (by coloring or numbering different groups) but we didn't use any **labeled** data.

In clustering, the "labels" here don't have meaning. Permutating them gives the same result in k-means. To give meaning to the labels, human inputs are required

Classification learns from minimizing the error between a prediction and an actual label.

Clustering doesn't derive learning from labelled data.

Classification quality metrics (accuracy / loss) do not apply to clustering.

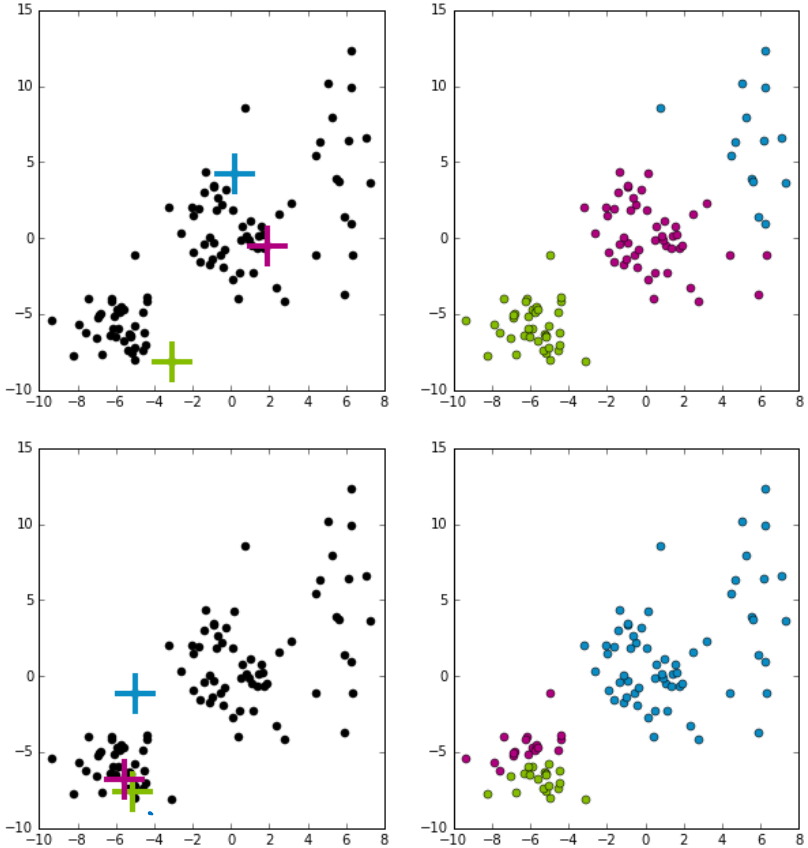
You may not be able to use validation set / k-fold to choose the best choice of k for clustering.

Effect of initialization

What does it mean for something to converge to a local optima?

Some initialization can be bad and affect the quality of clustering

Initialization will greatly impact results!



Smart Initializing w/ k-means++

Making sure the initialized centroids are “good” is critical to finding quality local optima. Our purely random approach was wasteful since it’s very possible that initial centroids start close together.

Idea: Try to select a set of points farther away from each other.

k-means++ does a slightly smarter random initialization

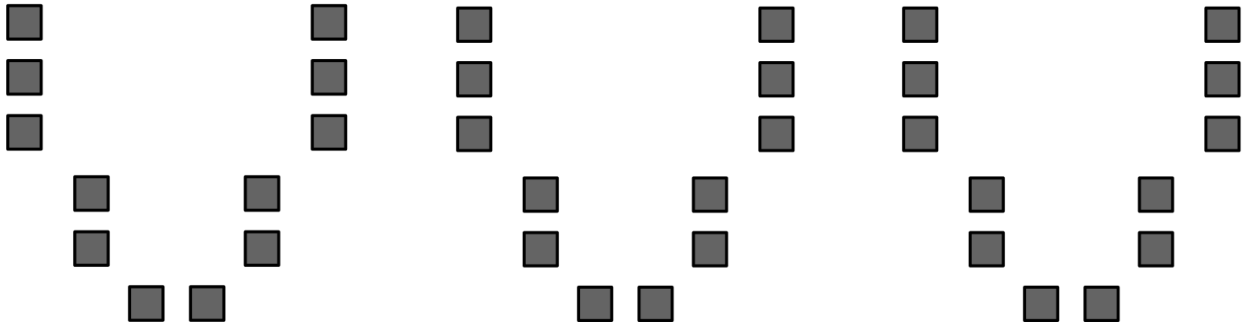
1. Choose first cluster $\mu^{(1)}$ from the data uniformly at random
2. For the current set of centroids $\mu^{(1)} \dots \mu^{(j)}$, compute the distance between each datapoint and its closest centroid $d_j\left(x^{(i)}\right)$
3. Choose a new centroid $\mu^{(j+1)}$ from the remaining data points with probability of $x^{(i)}$ being chosen proportional to $d_j\left(x^{(i)}\right)^2$
4. Repeat 2 and 3 until we have selected k centroids $\mu^{(1)} \dots \mu^{(k)}$

k-means++ Example

Start by picking a point at random

Then pick points proportional to their distances to their centroids

This tries to maximize the spread of the centroids!



k-means++ Pros / Cons

Pros

- Improves quality of local minima
- Faster convergence to local minima

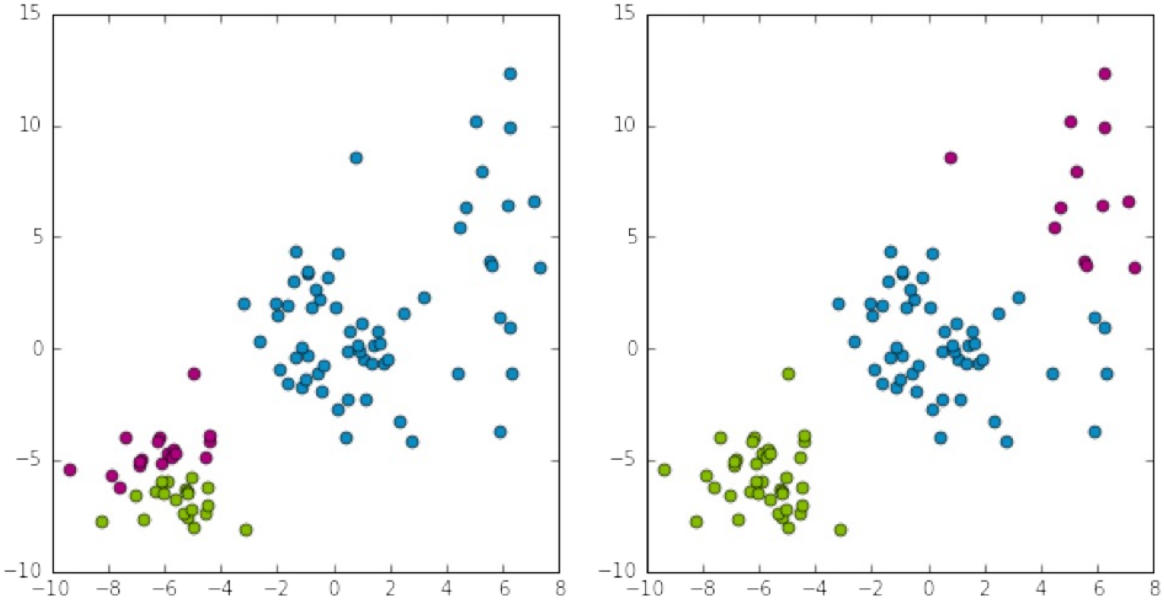
Cons

- Computationally more expensive at beginning when compared to simple random initialization

Assessing Performance

Which Cluster?

Which clustering would I prefer?



k-means is trying to optimize the **heterogeneity** objective

$$\underset{y,\mu}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} \left|\left|\mu^{(j)} - x^{(i)}\right|\right|_2^2$$

Objective function for k-means clustering

k-means is trying to minimize the (heterogeneity) objective

$$\underset{y,\mu}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} \left|\left|\mu^{(j)} - x^{(i)}\right|\right|_2^2$$

Step 0: Initialize cluster centers

Repeat until convergence:

- Step 1: Assign each example to its closest cluster centroid
- Step 2: Update the centroids to be the mean of all the points assigned to that cluster

The objective value will converge in finite time.

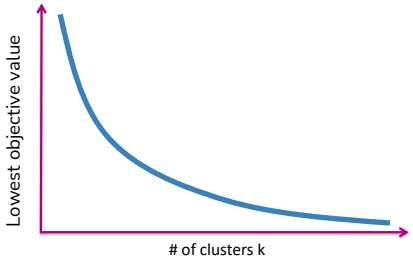
Poll Everywhere

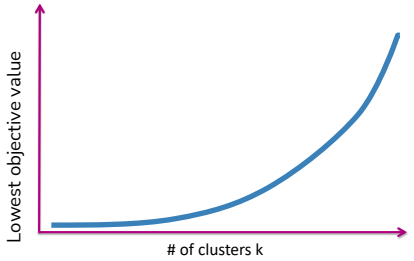
Think

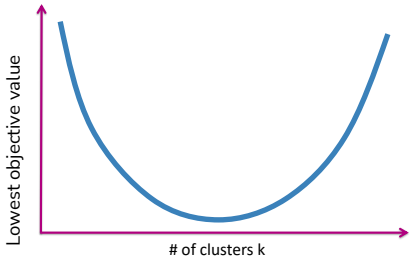
1 min

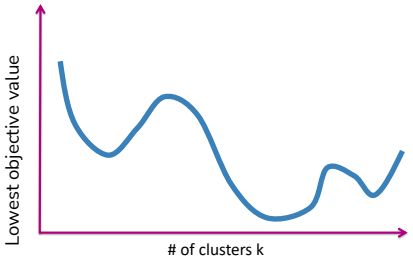
pollev.com/cs416

Consider trying k-means with different values of k. Which of the following graphs shows how the globally optimal heterogeneity changes for each value of k?









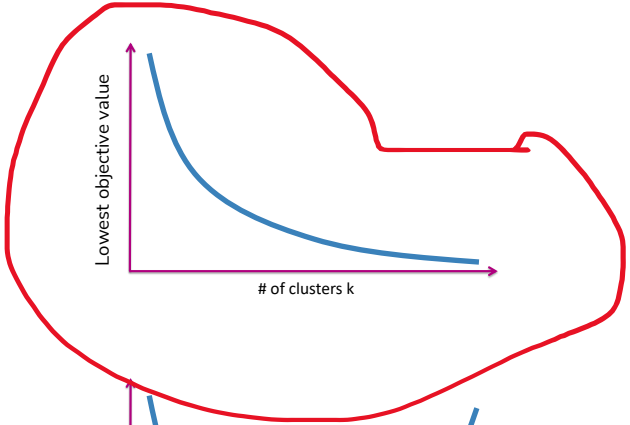
Poll Everywhere

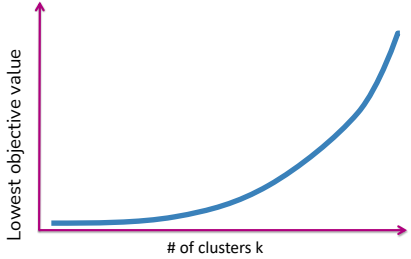
Think

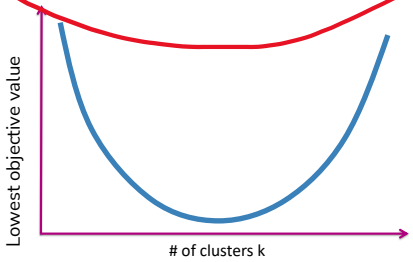
Discuss in groups for 2 mins

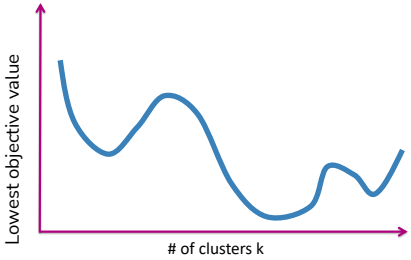
pollev.com/cs416

Consider trying k-means with different values of k. Which of the following graphs shows how the globally optimal heterogeneity changes for each value of k?









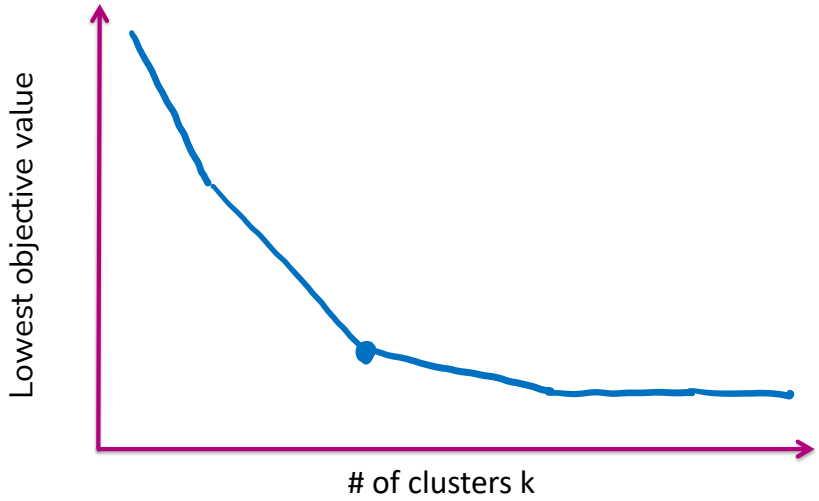
How to Choose k?

No right answer!

Human input is important. For example, if you cluster a lot of movies, you can pre-define the number of categories.

Practically, can choose k based on specific applications, such as based on cost-benefit tradeoff. The larger k is, the more costly it is.

In general, look for the “elbow” in the graph

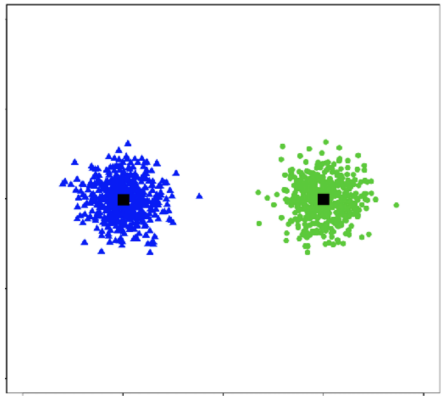


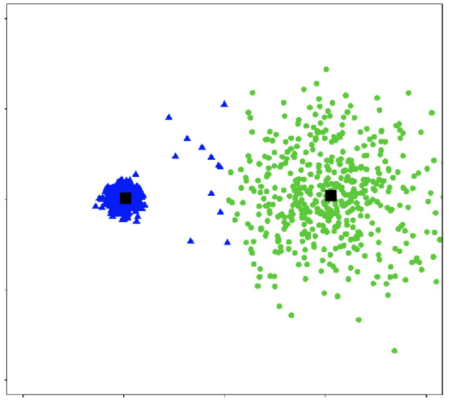
Other methods: BIC (Bayesian Information Criterion), Silhouette, etc.

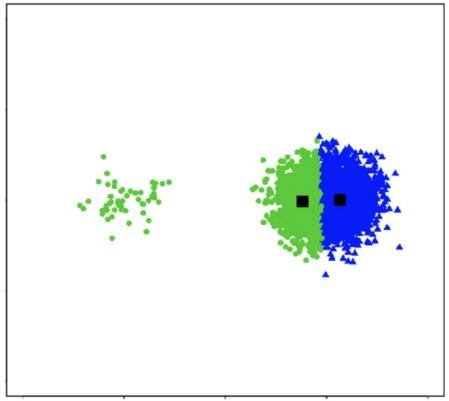
Note: You will usually have to run k-means multiple times for each k

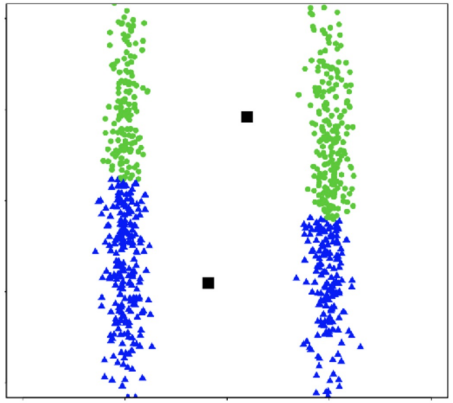
Cluster shape

- k-means works well for well-separated **hyper-spherical** clusters of the same size









Note about clustering

As we embark in the unsupervised learning task, a lot of previous concepts we have learned so far about model performance might not necessarily apply.

- Usually, we don't split train / test set for clustering problems, because there are no labels to measure how good a set of clusters are.
- Similarly, definitions like bias, variance, complexity bias-variance tradeoff might not work for unsupervised learning problems.
- The heterogeneity objective in clustering doesn't have the same meaning as training error in supervised learning tasks. There is no definition of accuracy for clustering either.

Recap

Differences between classification and clustering

What types of clusters can be formed by k-means

K-means algorithm

Convergence of k-means

How to choose k

Better initialization using k-means++