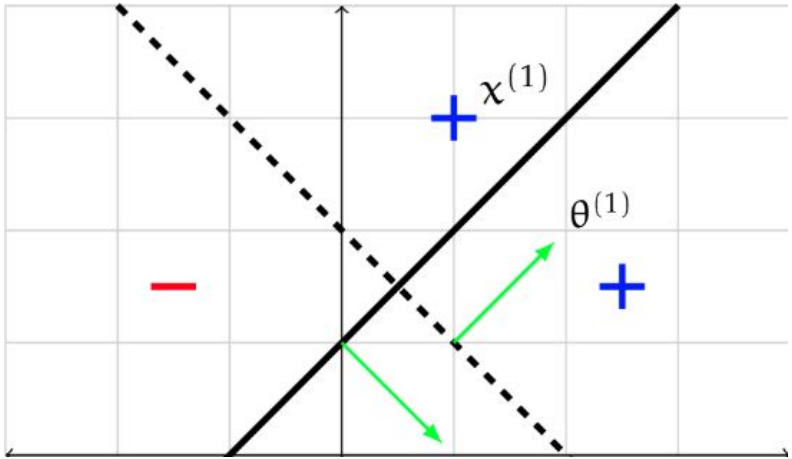
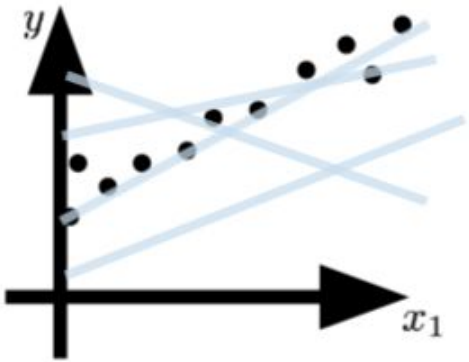
Introduction to Machine Learning

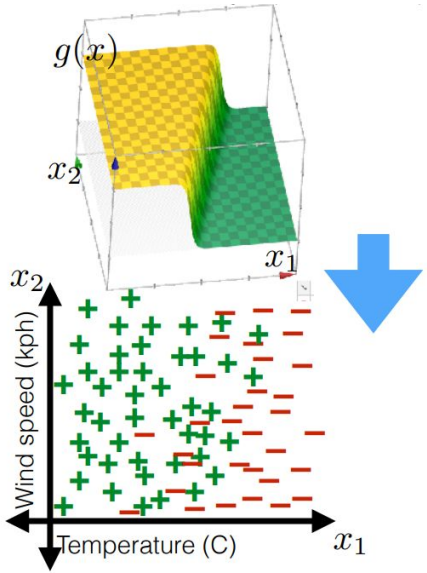


Clustering

Goals of Supervised Learning

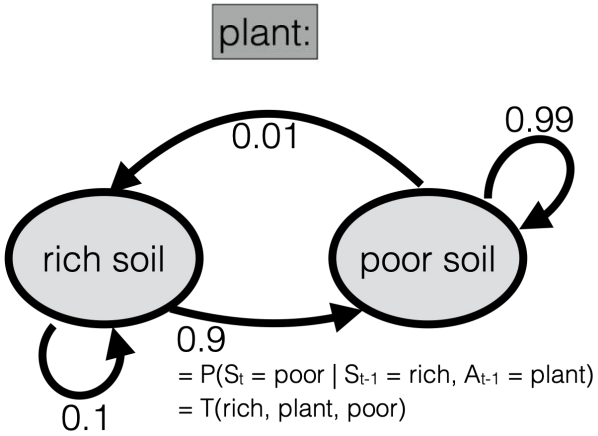
Learn a hypothesis h from labeled datase$D = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$ that has low error on unseen data.

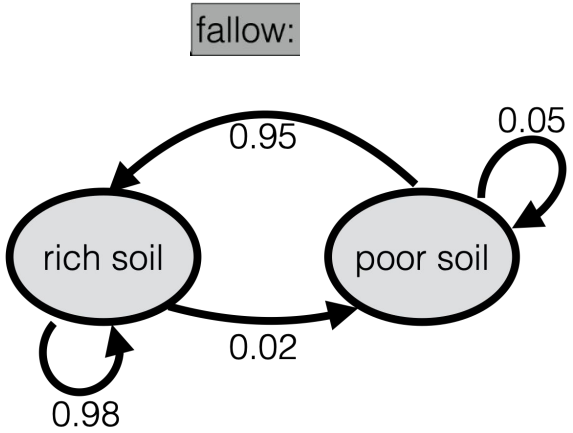




Goals of Reinforcement Learning

Find a "policy" $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes reward in an environment.

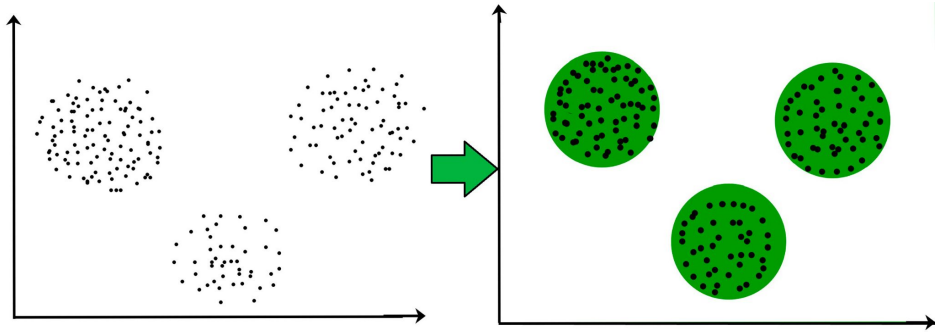


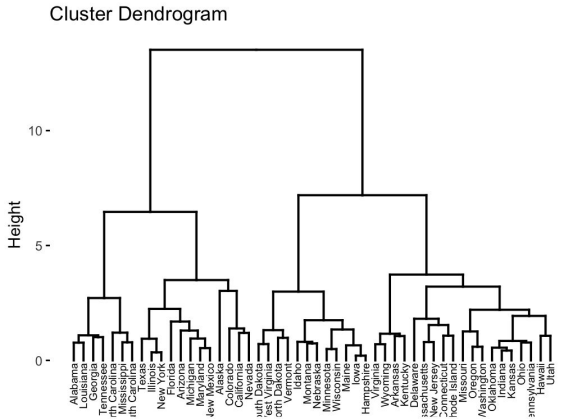


R(rich, plant)=100
R(poor, plant)=10
R(rich, fallow)=0
R(poor, fallow)=0

Goals of Unsupervised Learning?

Clustering

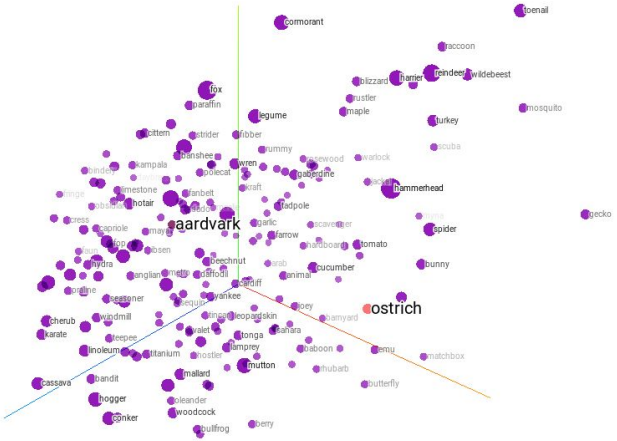


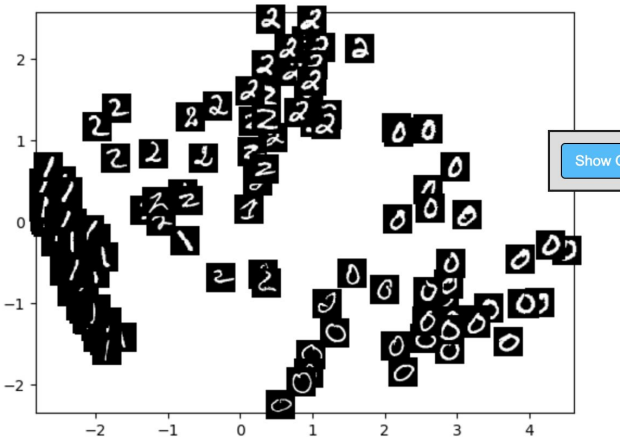


https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/

Goals of Unsupervised Learning?

Representation Learning





(Lab 7)

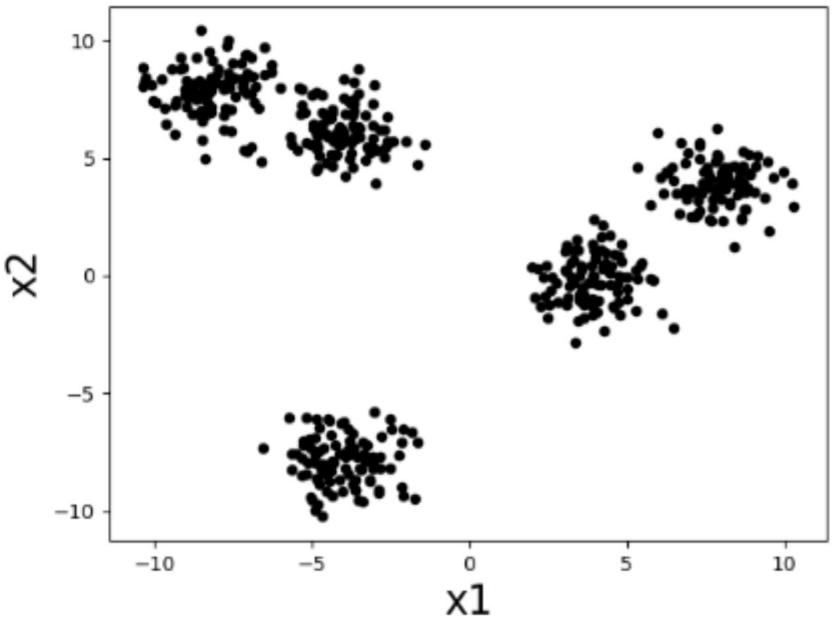
The ML Landscape

- **Supervised Learning**: Train a model that performs well (e.g., high accuracy) on unseen data.
- **Reinforcement Learning:** Learn a policy that maximizes expected reward in some environment.
- **Unsupervised Learning:**
    - Extract useful insights from data
    - Learn features for downstream tasks

Clustering

- Find a mapping from each data point to a cluster.

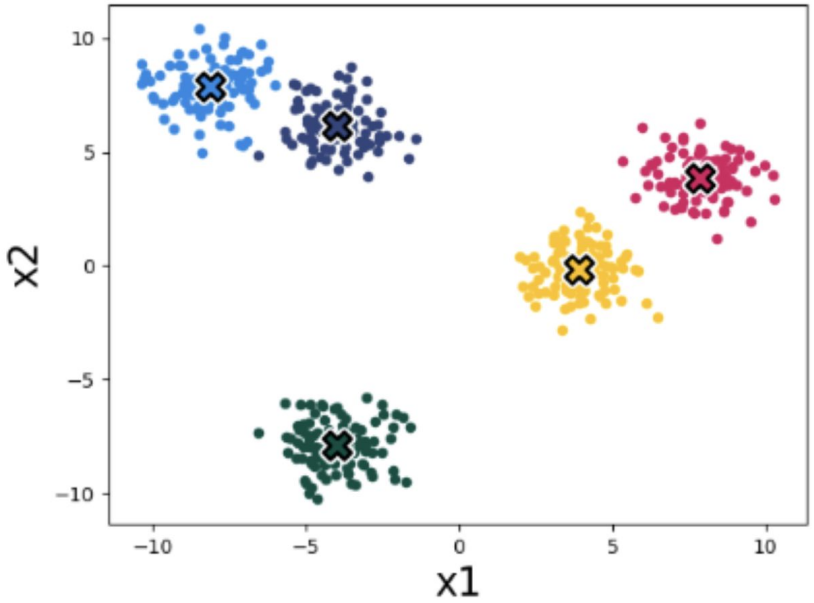
- Modeling choices:
  - How many clusters?
  - How do we define “close”?
  - How do we know if we have succeeded?



K-Means Clustering Objective

$$\arg\min_{\mu,y} \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{1}\{y^{(i)} = j\} \|x^{(i)} - \mu^{(j)}\|_2^2$$

Find K cluster assignments and cluster means such that across all data points, the squared Euclidean distance between the data point and the cluster mean of its assigned cluster is minimized.



K-Means Algorithm

Input:

1. Data points $\{x^{(i)}\}_{i=1}^{n}$
2. Number of clusters k
3. Number of iterations

Output:

1. An assignment of each point to a cluster.
2. A “centroid” of each cluster with which to assign new points.

K-Means Algorithm

Input:

1. Data points $\{x^{(i)}\}_{i=1}^{n}$
2. Number of clusters k
3. Number of iterations

Output:

1. An assignment of each point to a cluster.
2. A “centroid” of each cluster with which to assign new points.

k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

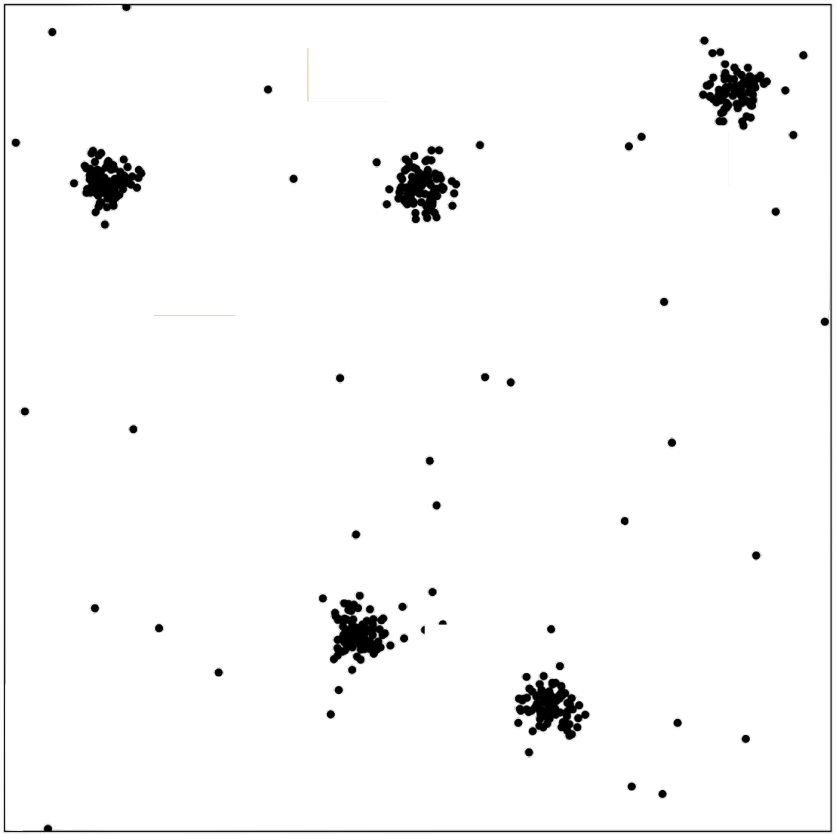
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

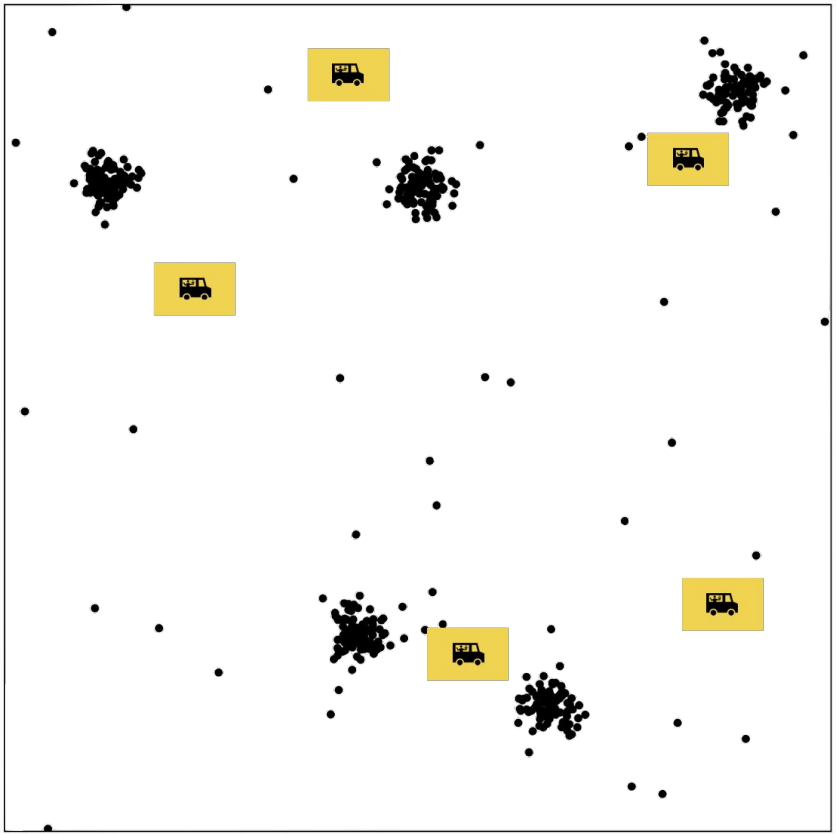
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



K = 5

k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

  $y_{\text{old}} = y$

  for i = 1 to n

    $y^{(i)} = \arg\min_j \|x^{(i)} - \mu^{(j)}\|_2^2$

  for j = 1 to k

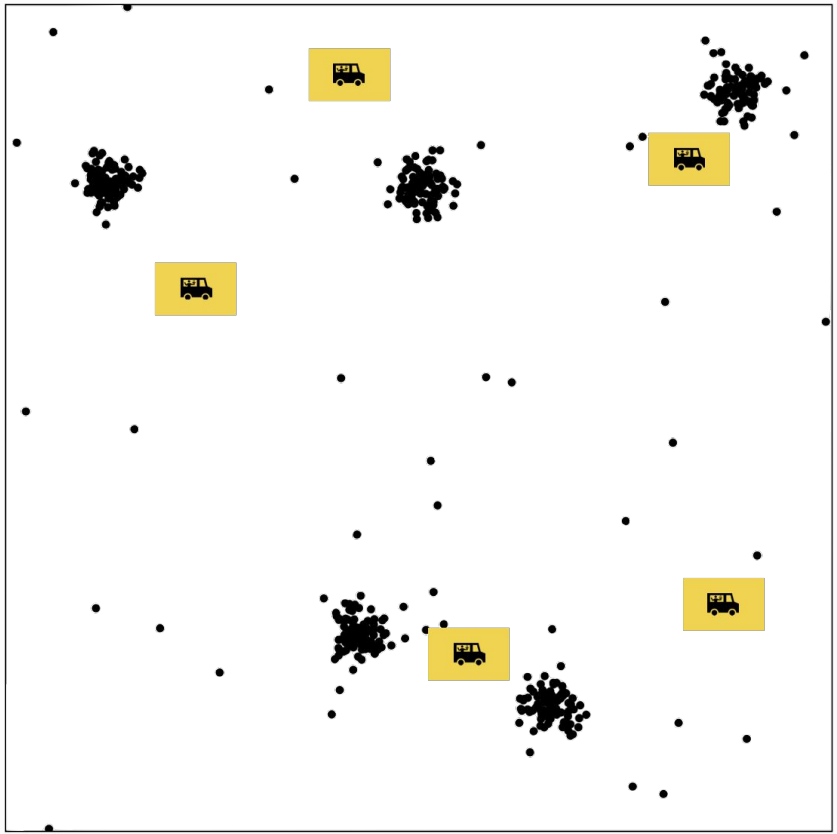
    $\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$

  if $y = y_{\text{old}}$

    break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



K = 5

k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$\|^2_2$

o k

$= j\}x^{(i)}$

$i) = j\}$

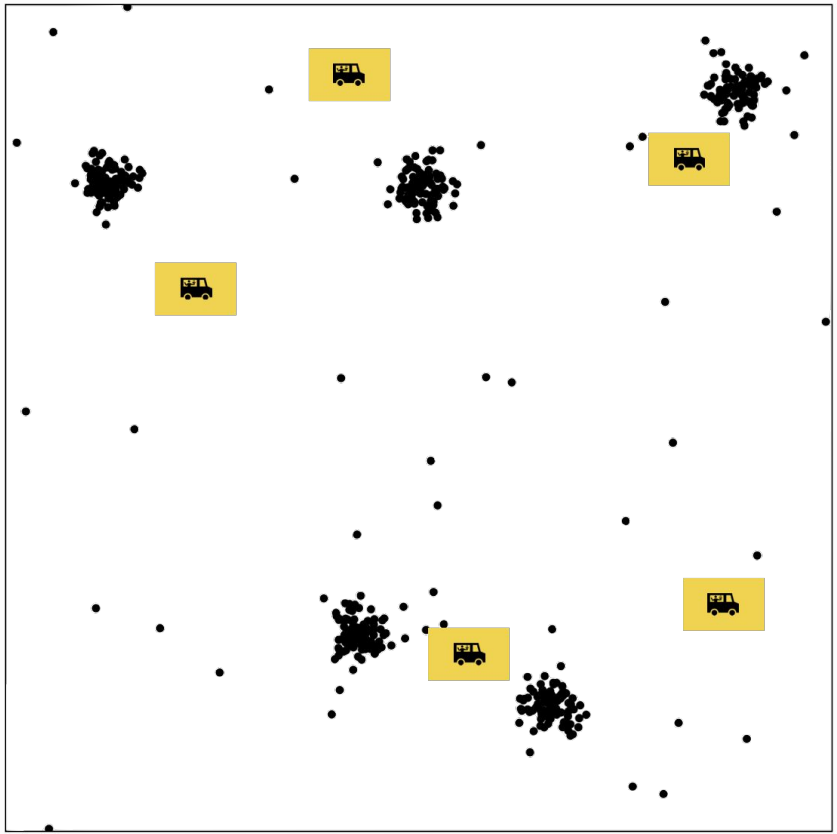
break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

Some options:

1. Choose *k* data points uniformly at random, *without* replacement
2. Choose uniformly at random within the span of the data

K-Means Algorithm



k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

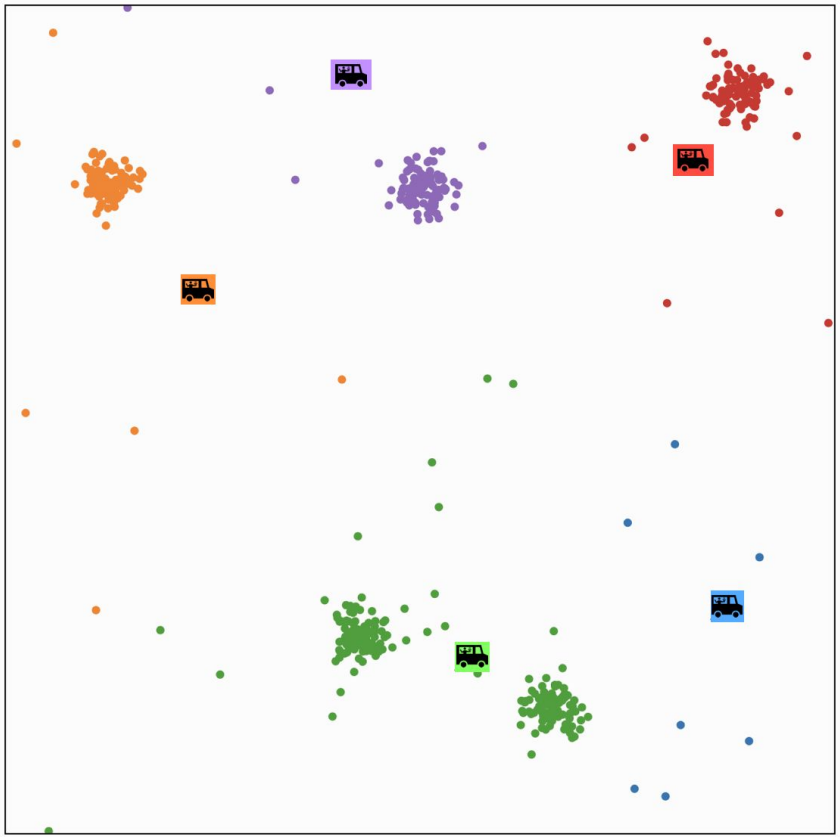
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

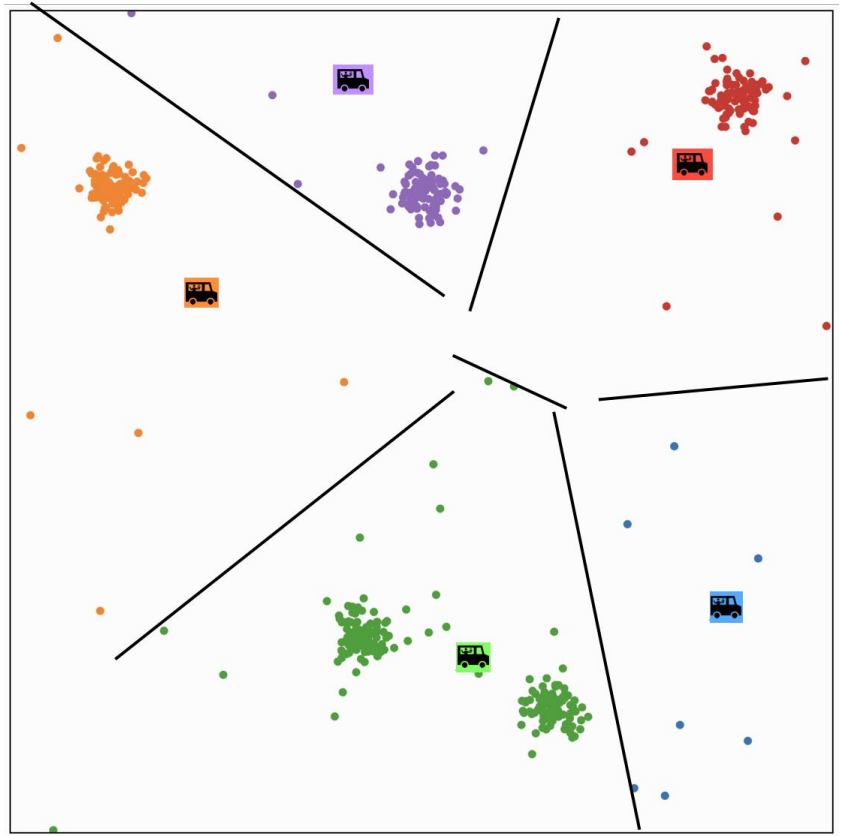
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_j \|x^{(i)} - \mu^{(j)}\|_2^2$$

Assign each data point to closest "centroid"

for j = 1 to k

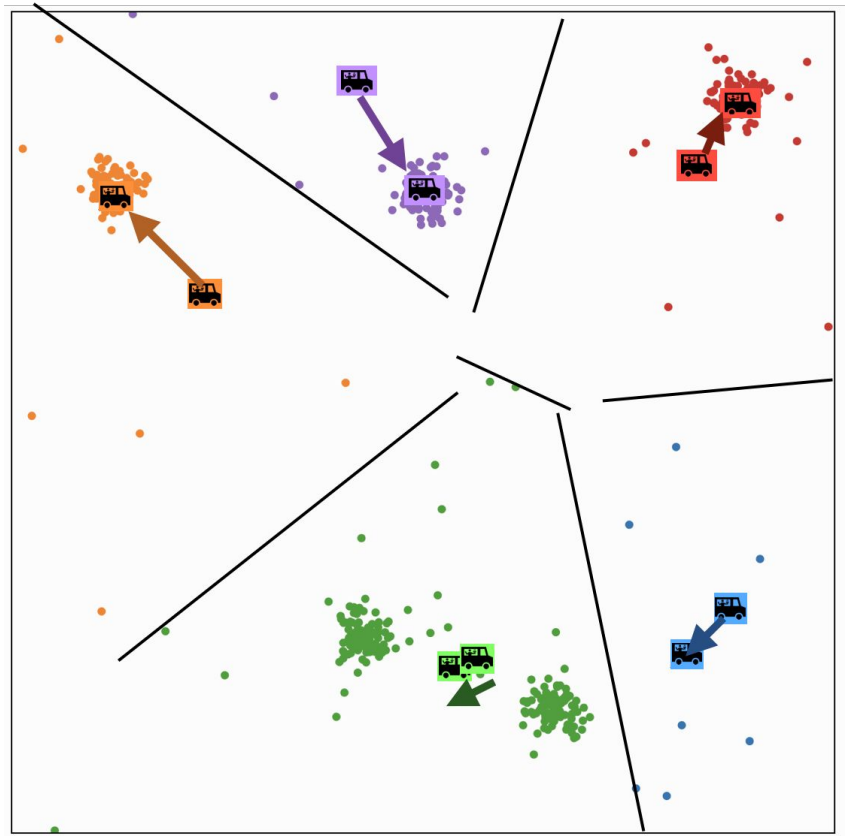
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



Update centroid as the average of all the points assigned to it

k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_j \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

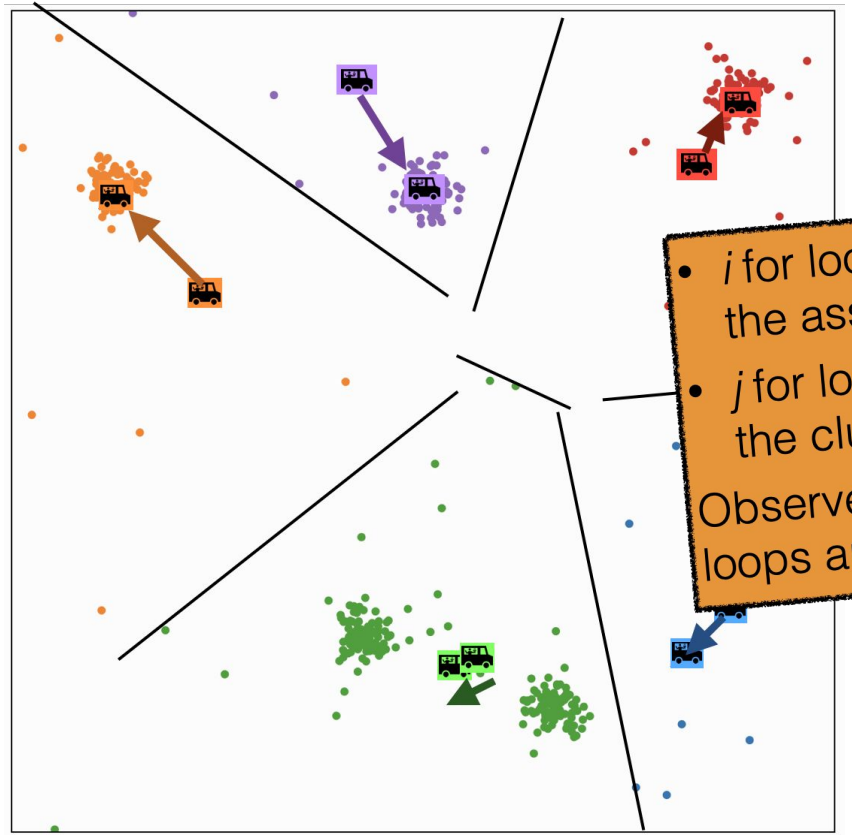
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



- *i* for loop: update the assignments
- *j* for loop: update the cluster centers

Observe: these two loops are *not* nested

k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_j \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

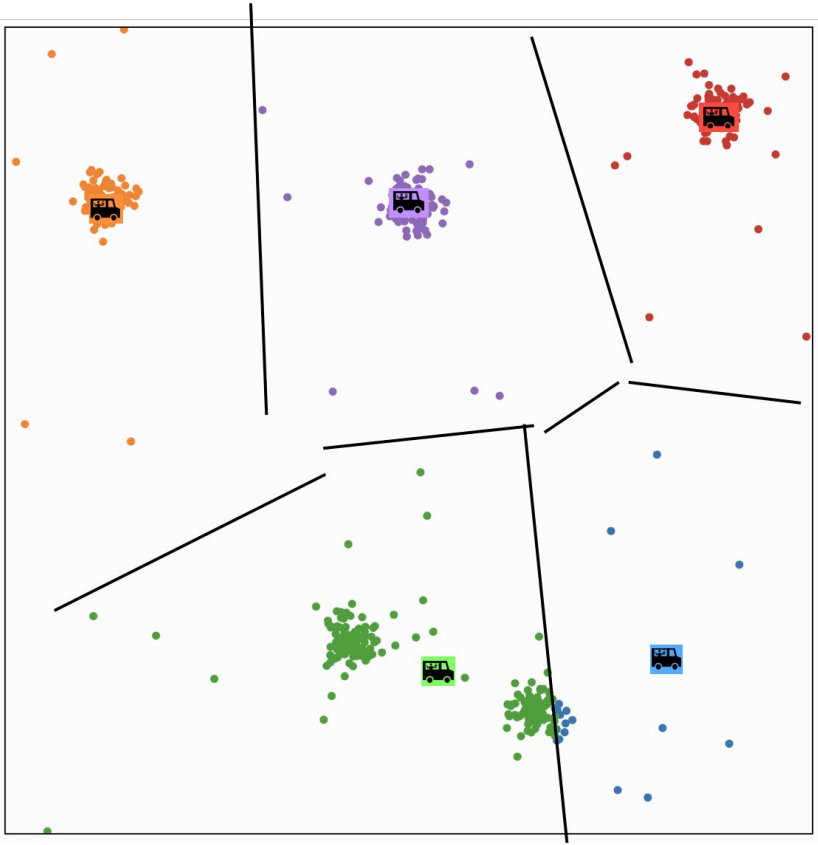
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



Iteration 2

k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_j \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

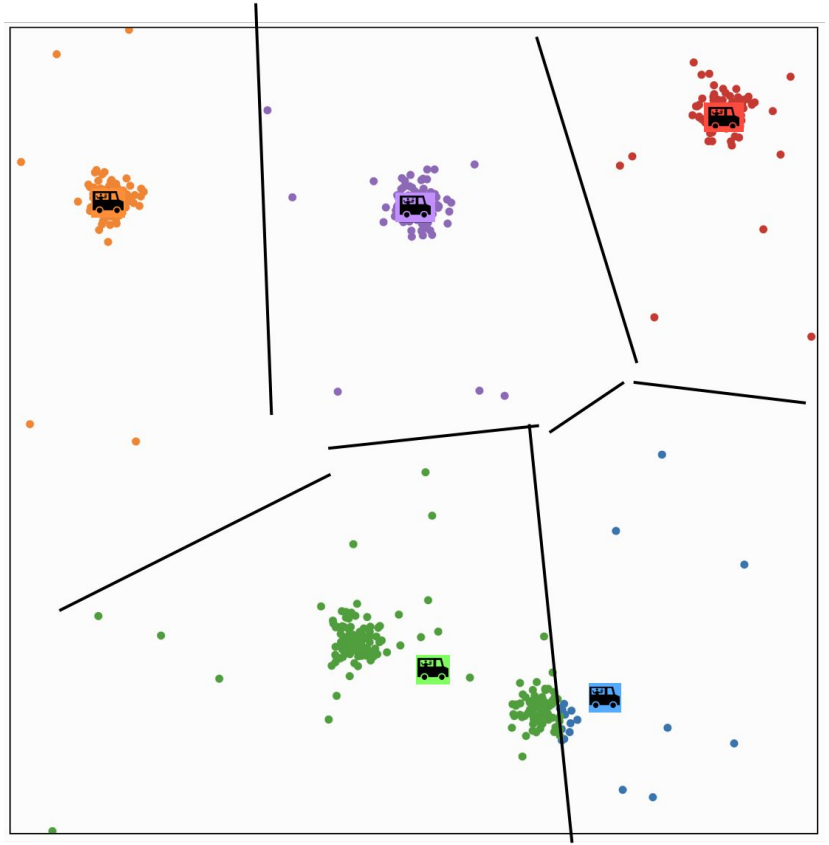
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



Iteration 2

k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

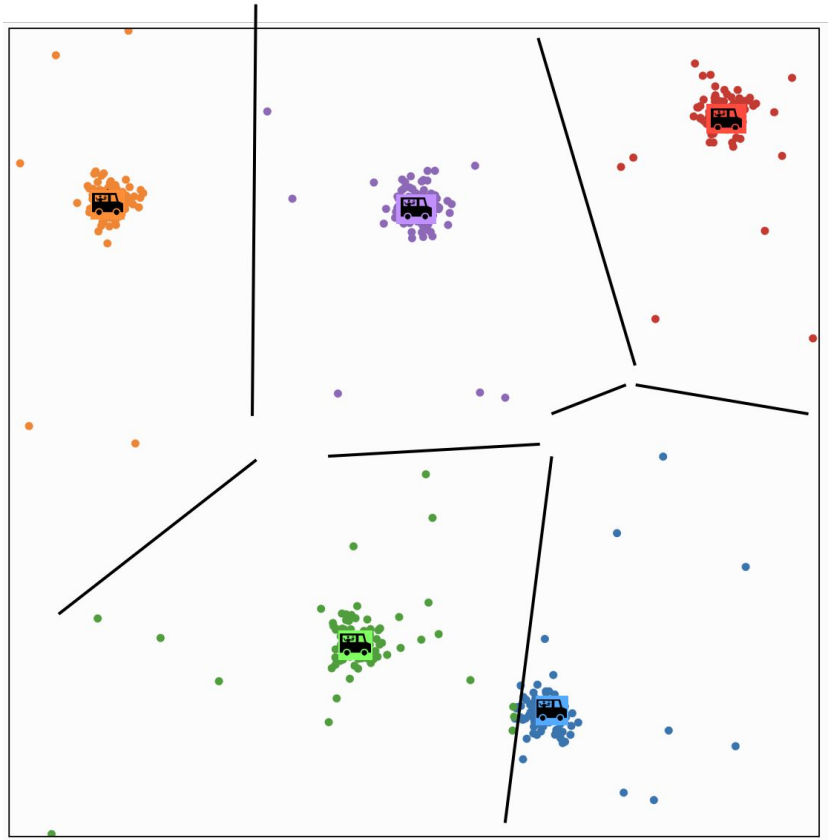
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

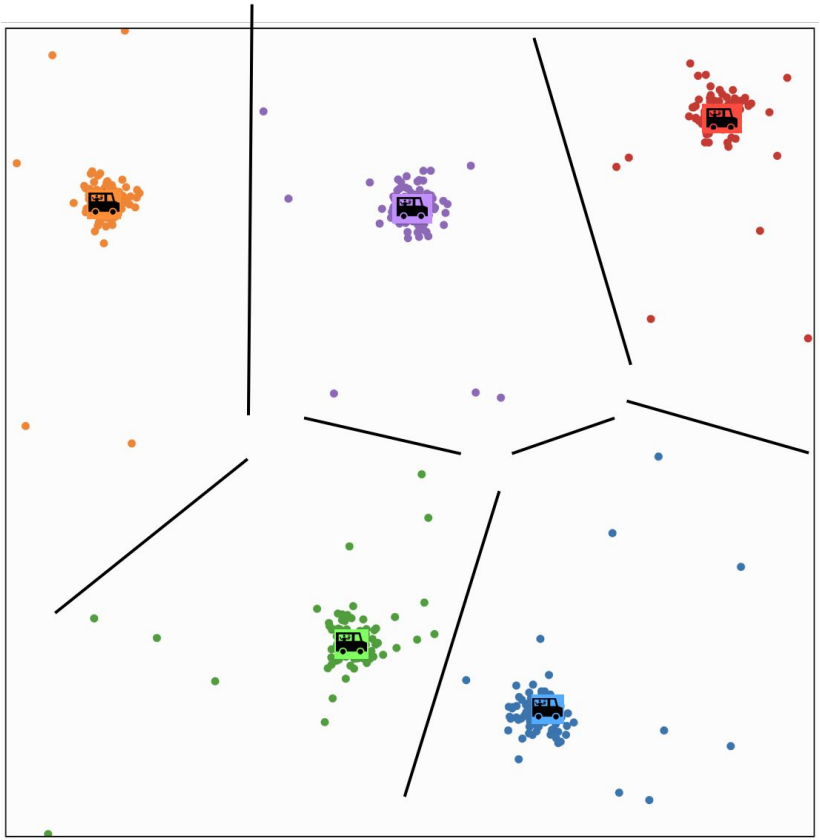
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

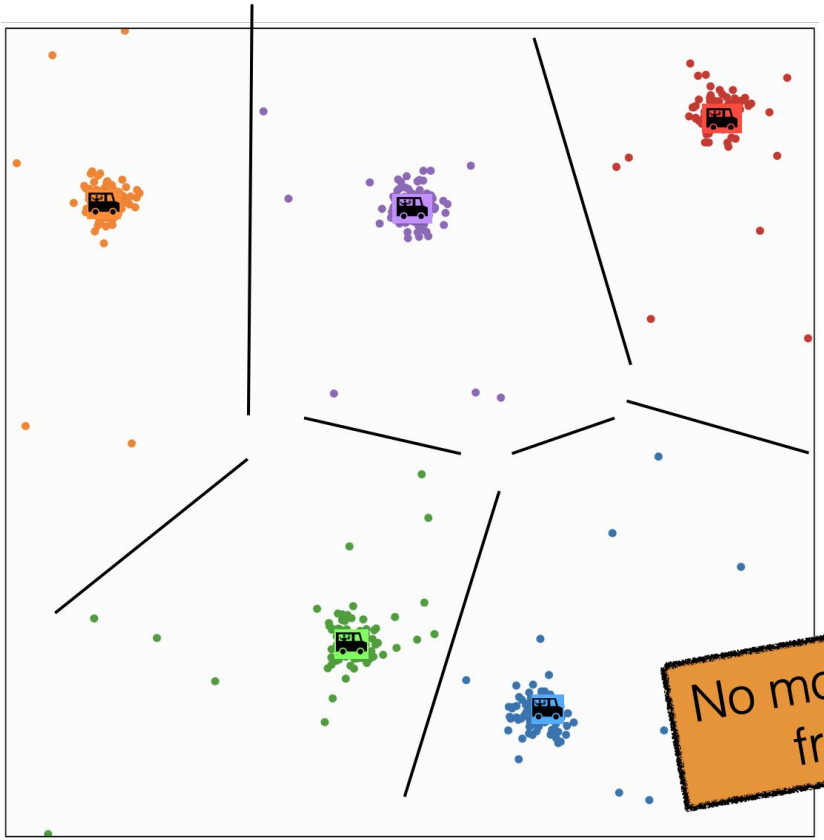
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$$y^{(i)} = \arg\min_{j} \|x^{(i)} - \mu^{(j)}\|_2^2$$

for j = 1 to k

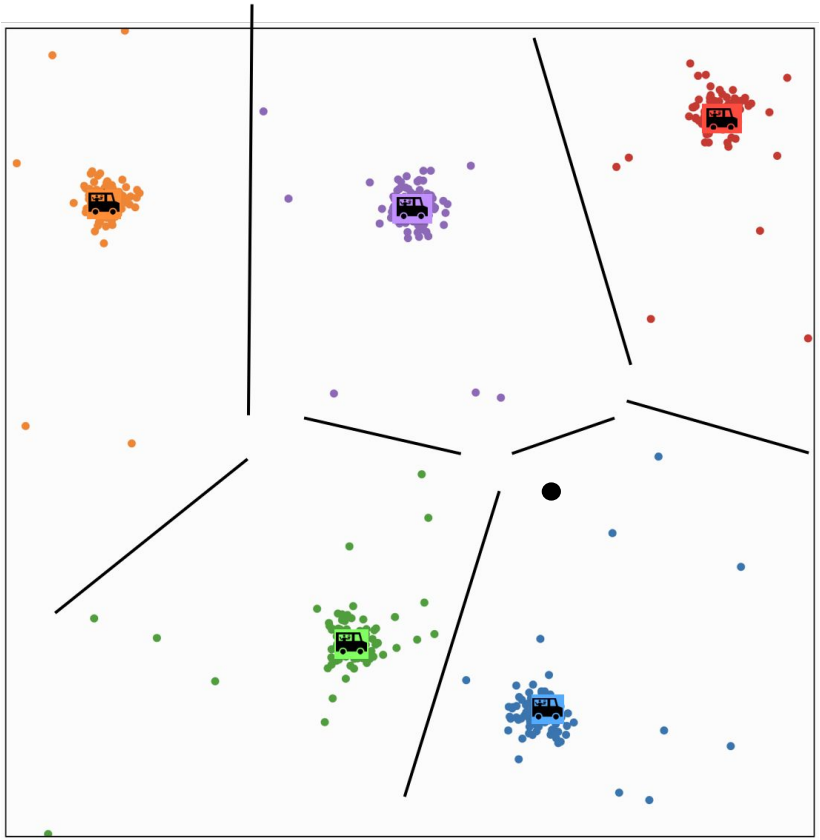
$$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$$

if $y = y_{\text{old}}$

break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



k-means(k, τ)

Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

for t = 1 to τ

$y_{\text{old}} = y$

for i = 1 to n

$y^{(i)} = \arg\min_j \|x^{(i)} - \mu^{(j)}\|_2^2$

for j = 1 to k

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

if $y = y_{\text{old}}$

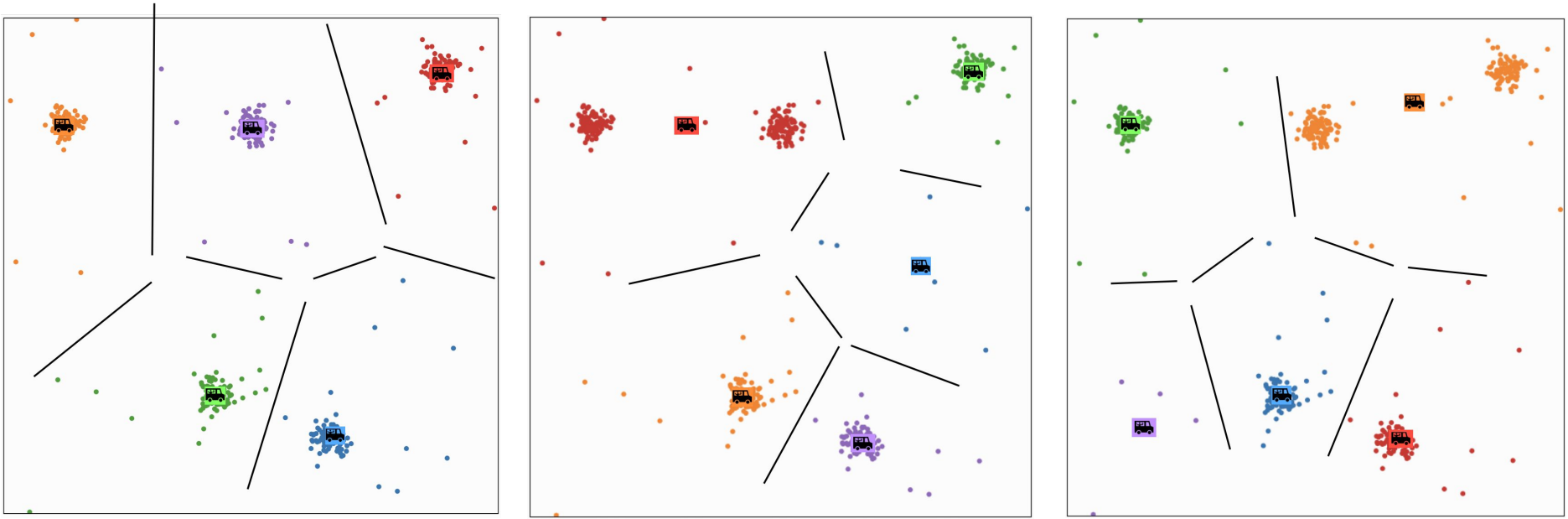
break

return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

Use centroid to cluster new data

K-Means Algorithm

K-means is sensitive to initialization!

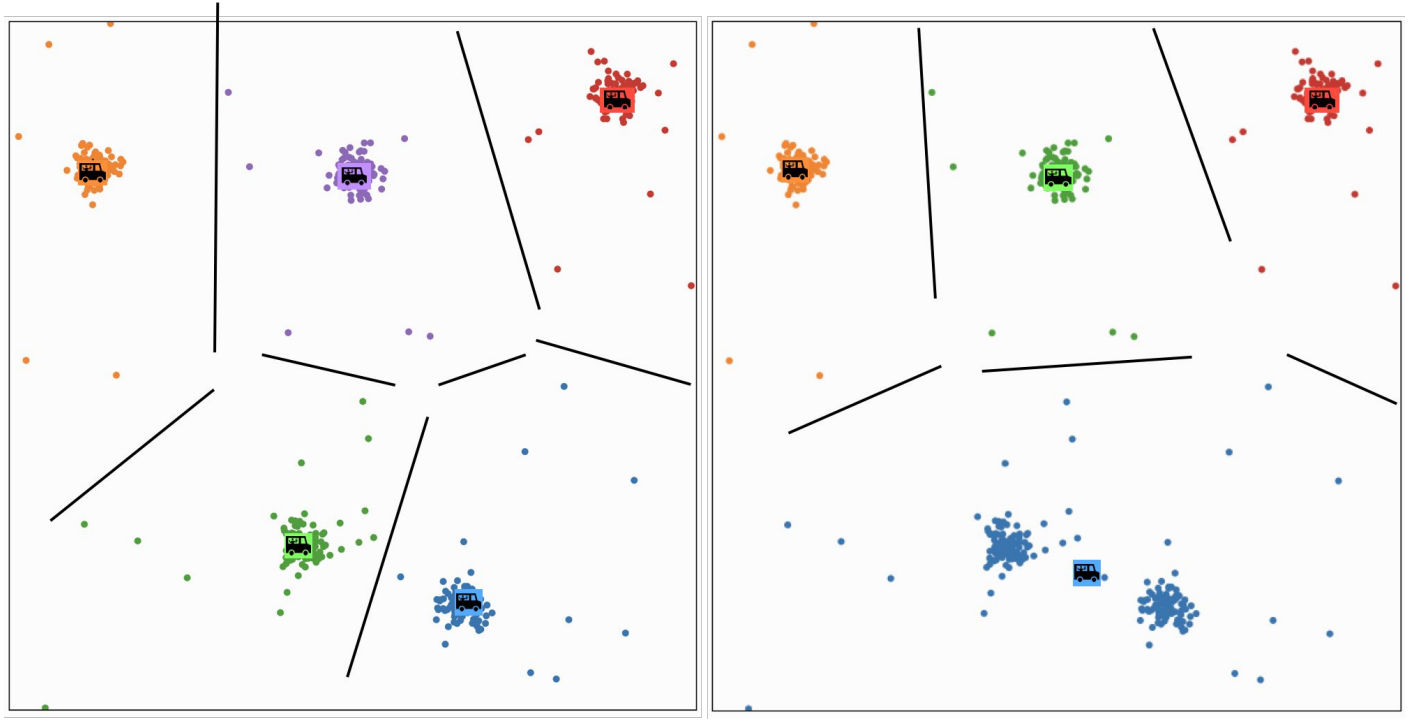


K-means is guaranteed to converge with iterations, but not necessarily to the global minimum.

K-Means Algorithm

K-means is sensitive to initialization!
And number of clusters

Why can't we just increase K?



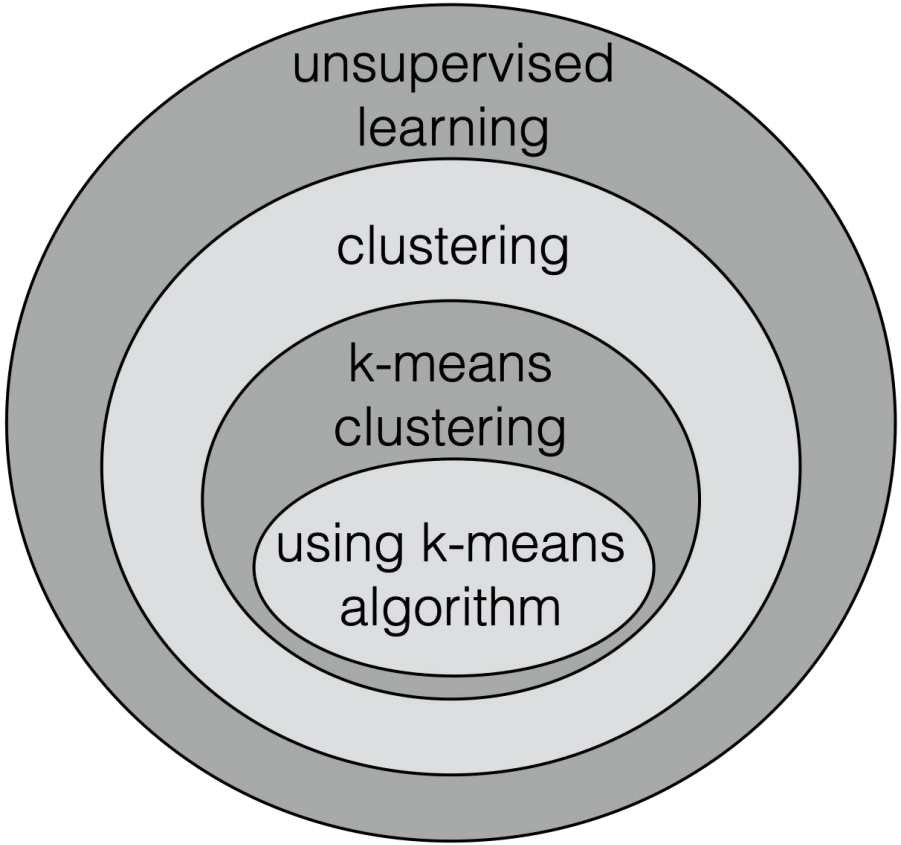
K-Means Algorithm

K-means is sensitive to initialization!
And number of clusters
And choice of distance metric

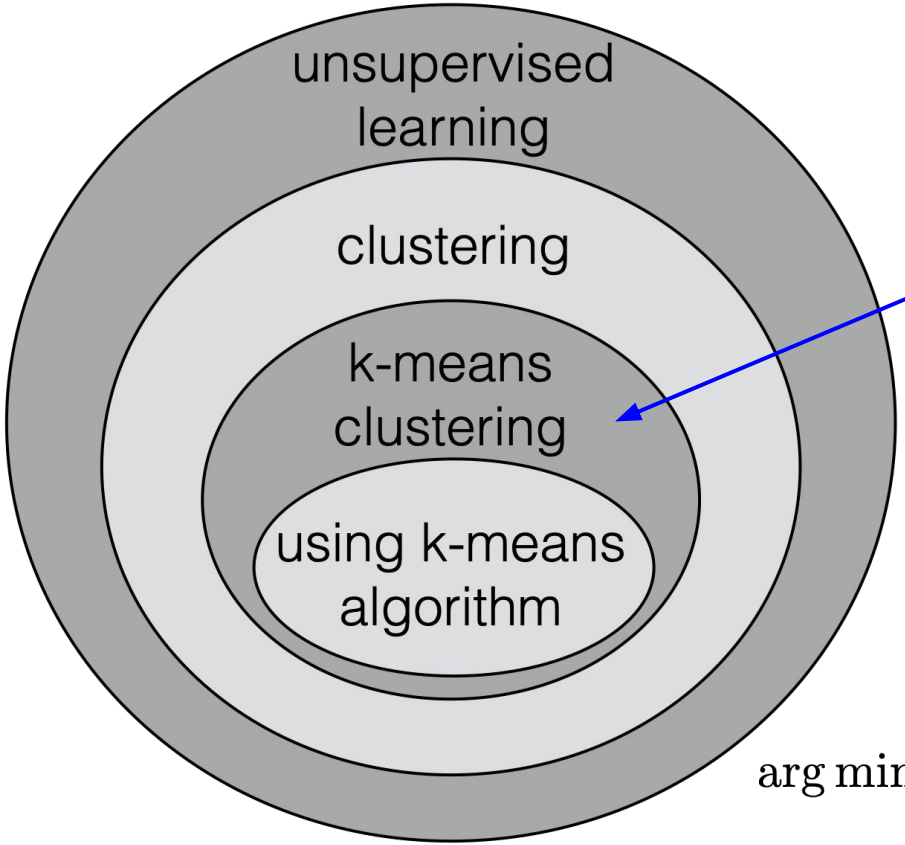
What are some issues with this “distance”?

k-means(k, τ)
Init $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$
for t = 1 to τ
$y_{\text{old}} = y$
for i = 1 to n
$y^{(i)} = \arg\min_j \|x^{(i)} - \mu^{(j)}\|_2^2$
for j = 1 to k
$\mu^{(j)} = \frac{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbf{1}\{y^{(i)} = j\}}$
if $y = y_{\text{old}}$
break
return $\{\mu^{(j)}\}_{j=1}^{k}, \{y^{(i)}\}_{i=1}^{n}$

K-Means Algorithm



K-Means Algorithm



Can also just do gradient descent!

$$L(\mu) = \sum_{i=1}^{n} \min_{j} \left\| x^{(i)} - \mu^{(j)} \right\|_2^2$$

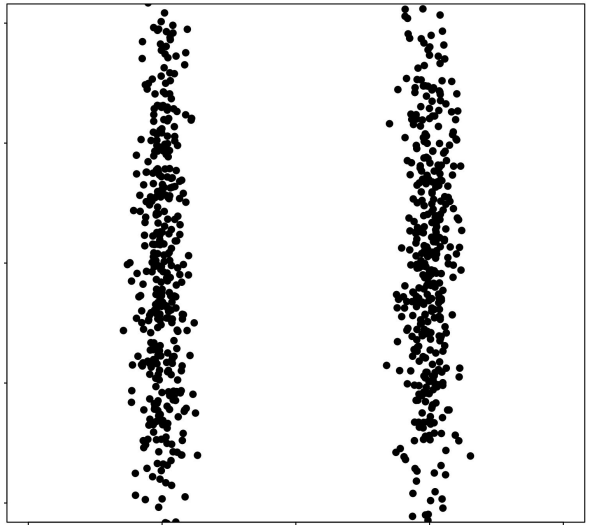
K-means **algorithm** is just one way of optimizing the K-means **objective**

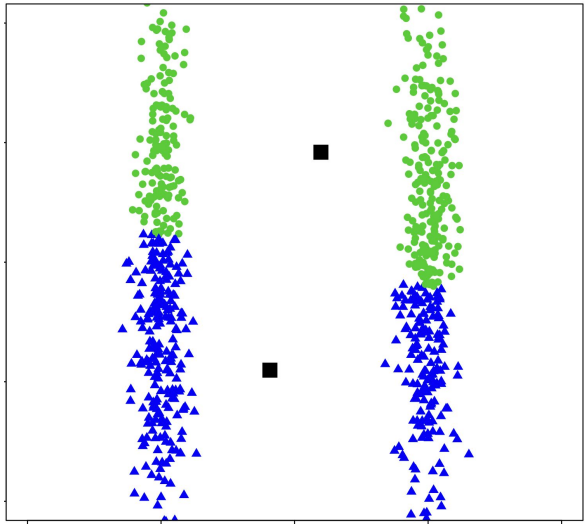
$$\arg\min_{\mu,y} \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{1}\{y^{(i)} = j\} \|x^{(i)} - \mu^{(j)}\|_2^2$$

When to use K-Means?

K-means works well when:

- Data “circular”

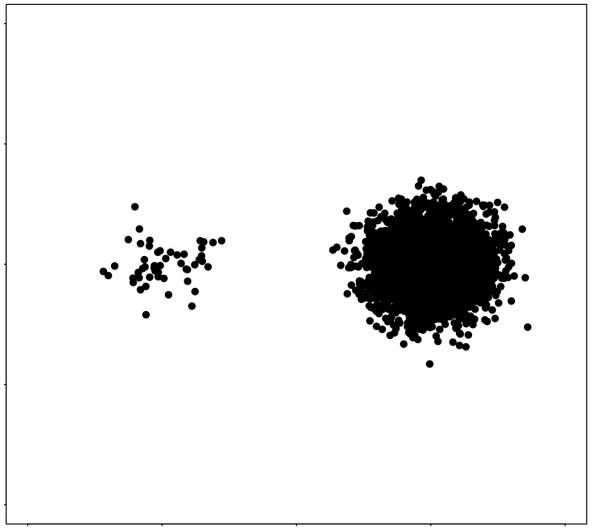


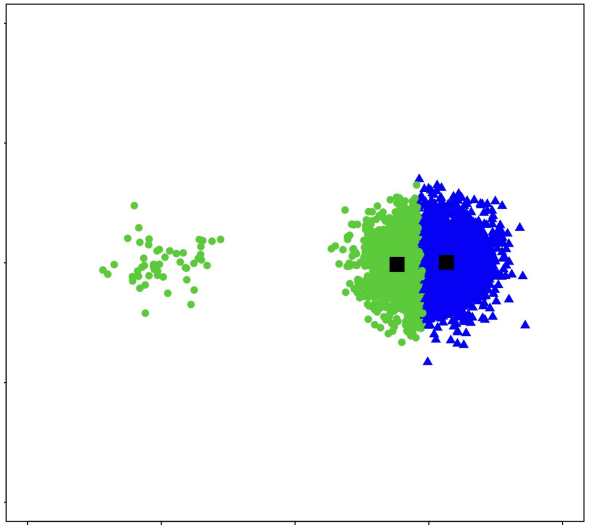


When to use K-Means?

K-means works well when:

- Data “circular”
- Clusters have roughly the same size





When to use K-Means?

K-means works well when:

- Data “circular”
- Clusters have roughly the same size
- Clusters are well separated