INNOPOLIS UNIVERSITY

Machine Learning

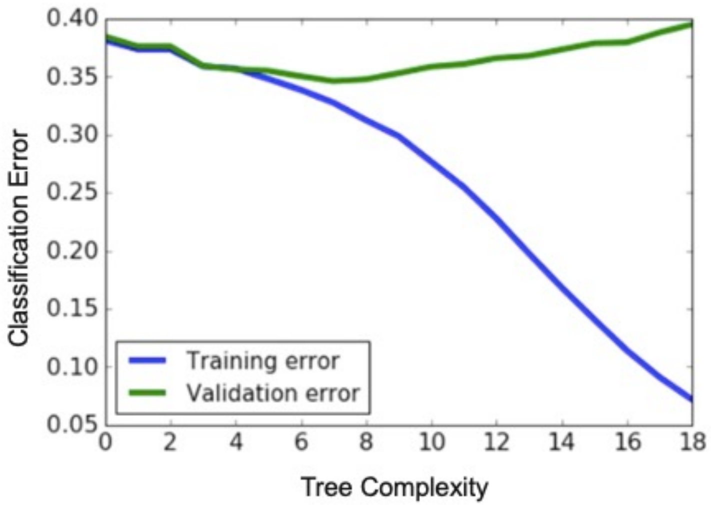
Prof. Adil Khan

Objectives

1. What is clustering? How is it different from classification? Why is it called unsupervised learning?
2. What is k-means? How does it work? What is its objective function? How is it motivated?
3. What is k-means++? Why is it motivated?
4. Limitations of k-means

Recap (1)

DT Overfitting



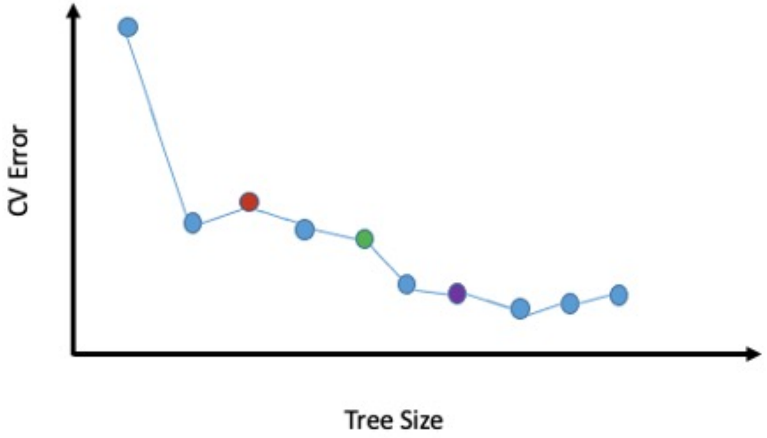
How to avoid Overfitting in DTs?

1. Early Stopping: Stop learning before the tree becomes too complex
2. Pruning: Simplify tree after learning algorithm terminates

Summary: Early Stopping

1. Limit tree depth
   - Stop splitting after a certain depth
2. Prediction performance
   - Do not consider a split unless it provides a significant boost in prediction performance
3. Minimum node size
   - Do not split a node unless it contains a significant number of data points

Why prune?



Recap (2)

Prune_split (T, M)

```
1. Compute total cost $C(T)$
2. Let $T_{small}$ be the tree after pruning $T$ at $M$
3. Compute $C(T_{small})$
4. If $C(T_{small}) < C(T)$, prune $T$ to $T_{small}$
```

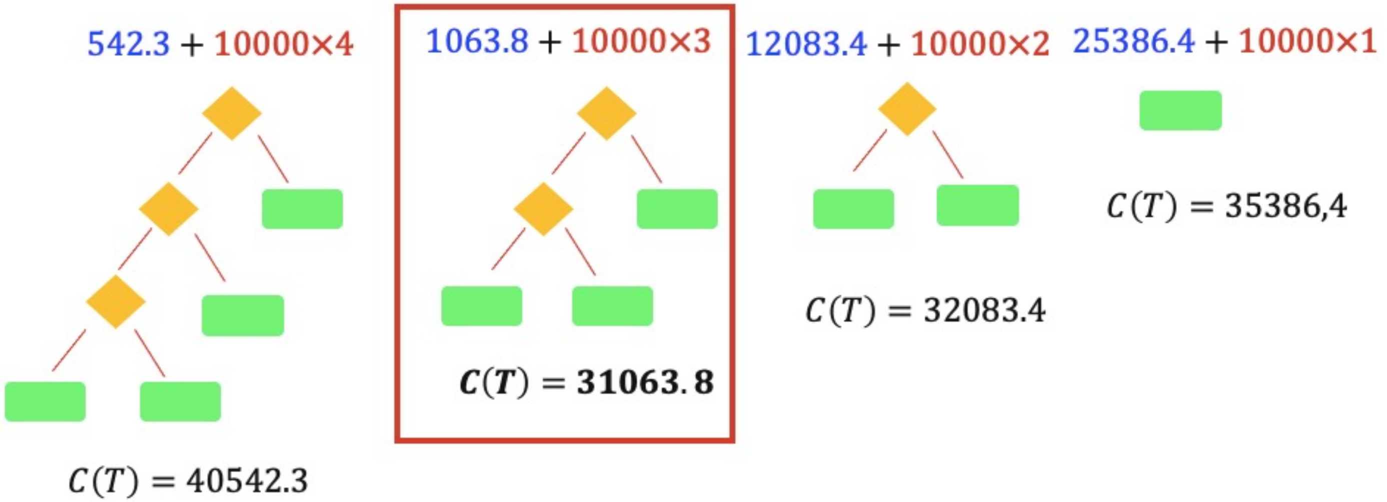
$$C(T) = Error(T) + \lambda L(T)$$

$Error(T)$ is prediction error (large means bad fit to the data)

$L(T)$ is Number of Leaves (large means likely to overfit)

Example

$C(T) = Error(T) + \lambda L(T)$ λ=10000



$542.3 + 10000×4$ $C(T) = 40542.3$

$1063.8 + 10000×3$ $\boldsymbol{C(T) = 31063.8}$

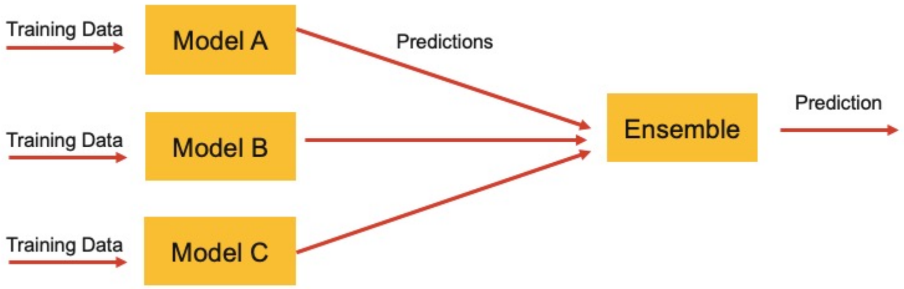
$12083.4 + 10000×2$ $C(T) = 32083.4$

$25386.4 + 10000×1$ $C(T) = 35386{,}4$

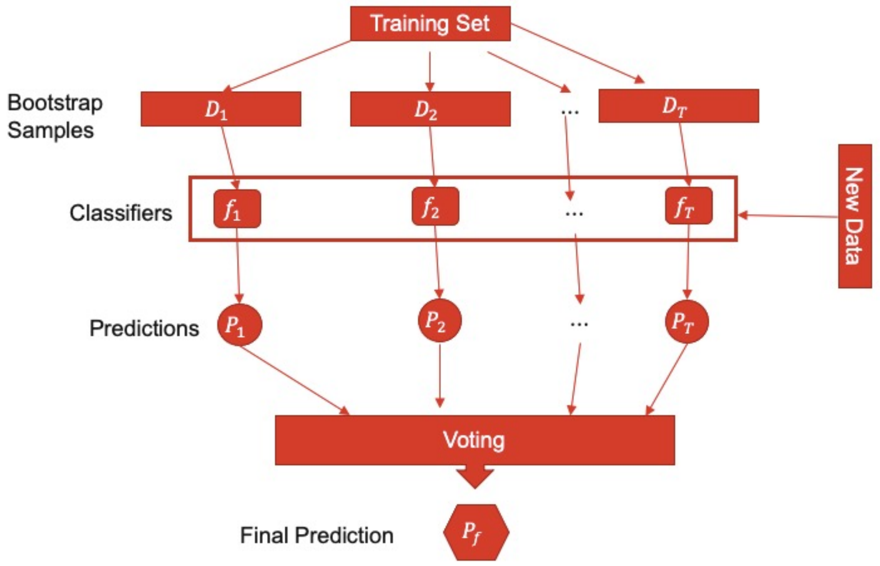
Recap (3)

Ensemble Learning

- Meta-learning algorithms that combine several ML models into one predictive model



Bagging: Reducing Variance using An Ensemble of Classifiers



Boosting: Converting Weak Learners to Strong Learners through Ensemble Learning

AdaBoost

1. Start with the same weights for all points: $\alpha_i = \frac{1}{m}$
2. For each $t = 1, \cdots, T$
   - Learn $f_t(x)$ with data weights α_i
   - Compute coefficient $\hat{w}_t$
   - Recompute weights α_i
   - Normalize α_i

- Final model predicts as:

$$\hat{w}_t = \frac{1}{2} ln\left(\frac{1 - weighted\ error(f_t)}{weighted\ error(f_t)}\right)$$

$$\alpha_i \leftarrow \begin{cases} \alpha_i e^{-\hat{w}_t}, & \text{if } f_t(\mathbf{x}_i) = y_i \\ \alpha_i e^{\hat{w}_t}, & \text{if } f_t(\mathbf{x}_i) \neq y_i \end{cases}$$

$$\alpha_i \leftarrow \frac{\alpha_i}{\sum_{j=1}^{N} \alpha_j}$$

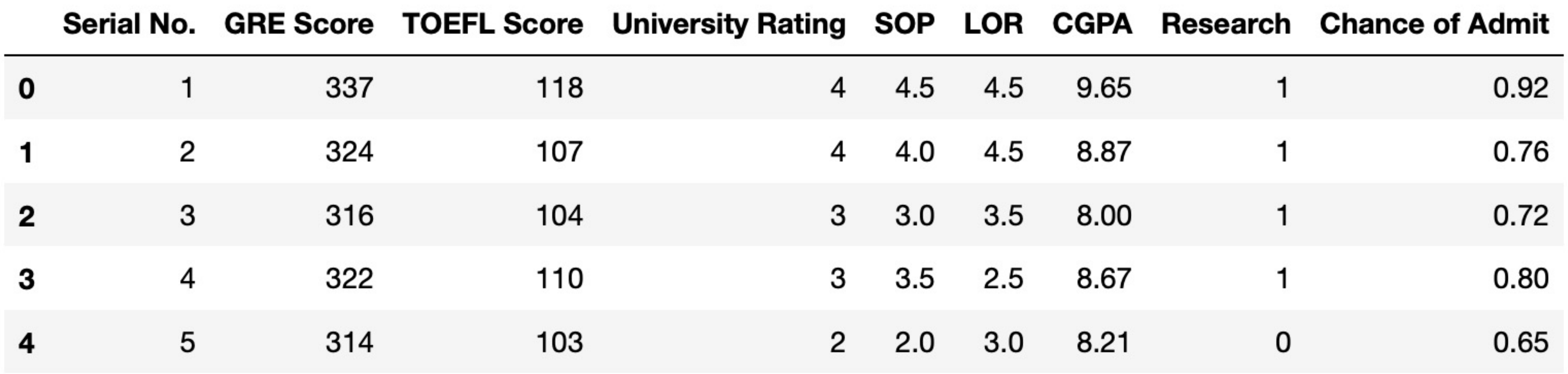
$$\hat{y} = sign\left(\sum_{t=1}^{T} \hat{w}_t f_t(x)\right)$$

Clustering

Recall: Supervised Learning

- Learning with a teacher
- Teacher provides the supervision
- In the context of machine learning, the labels Y provides the supervision for X

X Y



	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

Recall: Supervised Learning (2)

- Easy and well-defined

- Given $Y = f(X)$ and labeled dataset

$$D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}, \qquad \boldsymbol{x} \in \mathbb{R}^d$$

- Estimate the function f as $\hat{f}$, such that when a new input data is given, we can predict its outcome.

Unsupervised Learning

- We have only predictors (a.k.a inputs, or features) but no labels

$$D = \{\boldsymbol{x}_i\}_{i=1}^{N}, \qquad \boldsymbol{x} \in \mathbb{R}^d$$

- Then what is the goal of learning?

Scenario (1)

- Imagine you run an online store and would like to personalize your customers' shopping experience
- You think you can do this by providing each customer with personalized recommendations
- You do not know each user's personal preferences and tastes but you have lots of data on their purchases
- How can you used this data to create recommendations?

Scenario (2)

- Imagine you are a biologist interested in studying the behavior of bees
- You have collected a lot of videos (or other data on them)
- How can you use this data to study bee behavior?

Unsupervised Learning

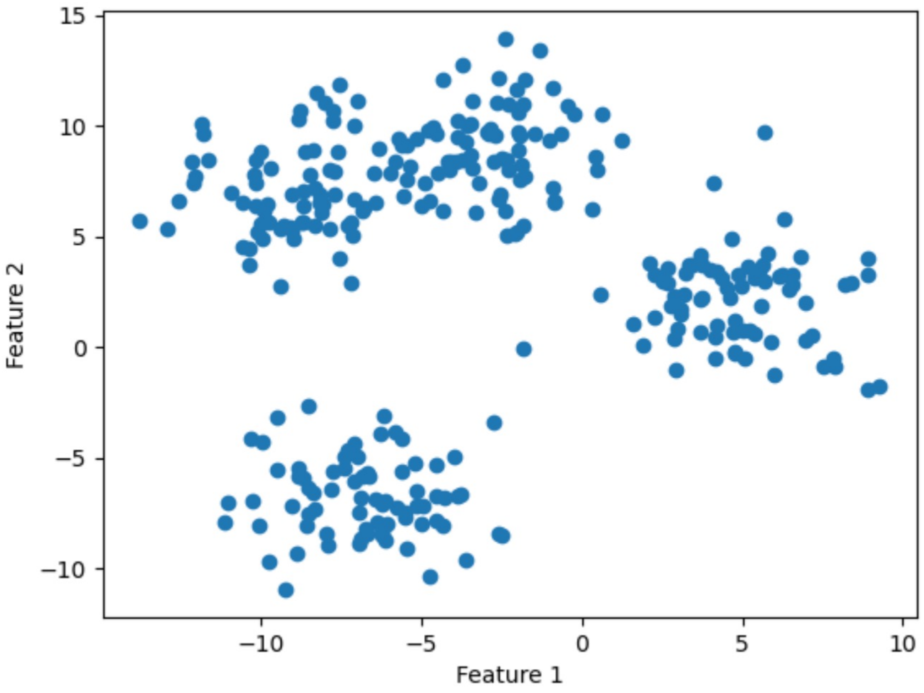
- In the discussed scenarios, we are not interested in prediction, because we do not have an associated response variable

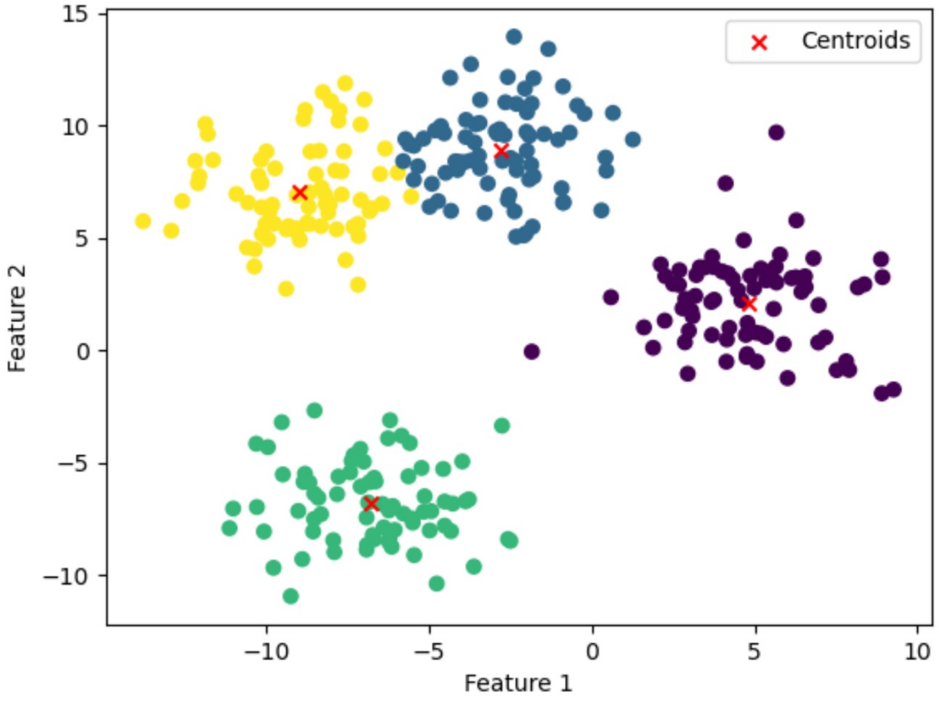
- Instead, the unsupervised **learning goal** is to model the hidden patterns or underlying structure in the given input data in order to learn about the data

- This *underlying structure* is what we usually refer to as *groups* of data

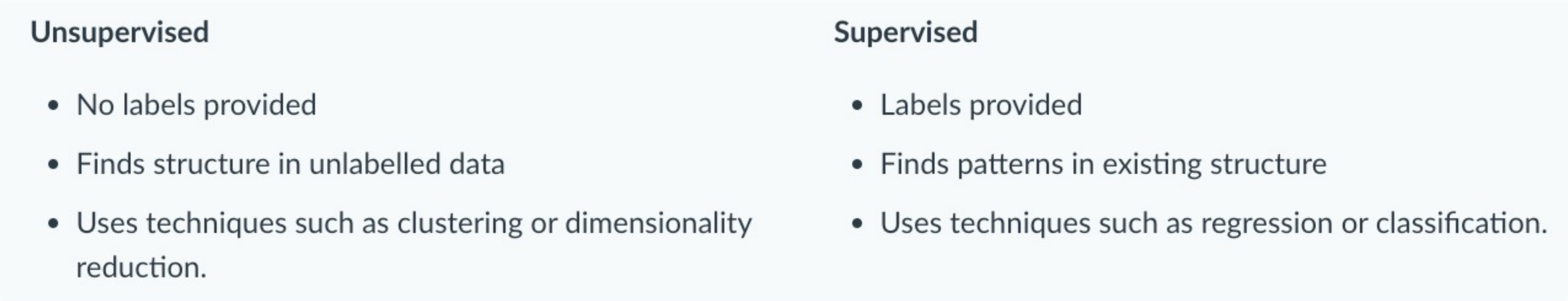
- And these groups are what we refer to as ***Clusters***

Clustering


15
10
5
0
−5
−10
Feature 2
−10
−5
0
5
10
Feature 1


Centroids
15
10
5
0
−5
−10
Feature 2
−10
−5
0
5
10
Feature 1

Key Takeaway

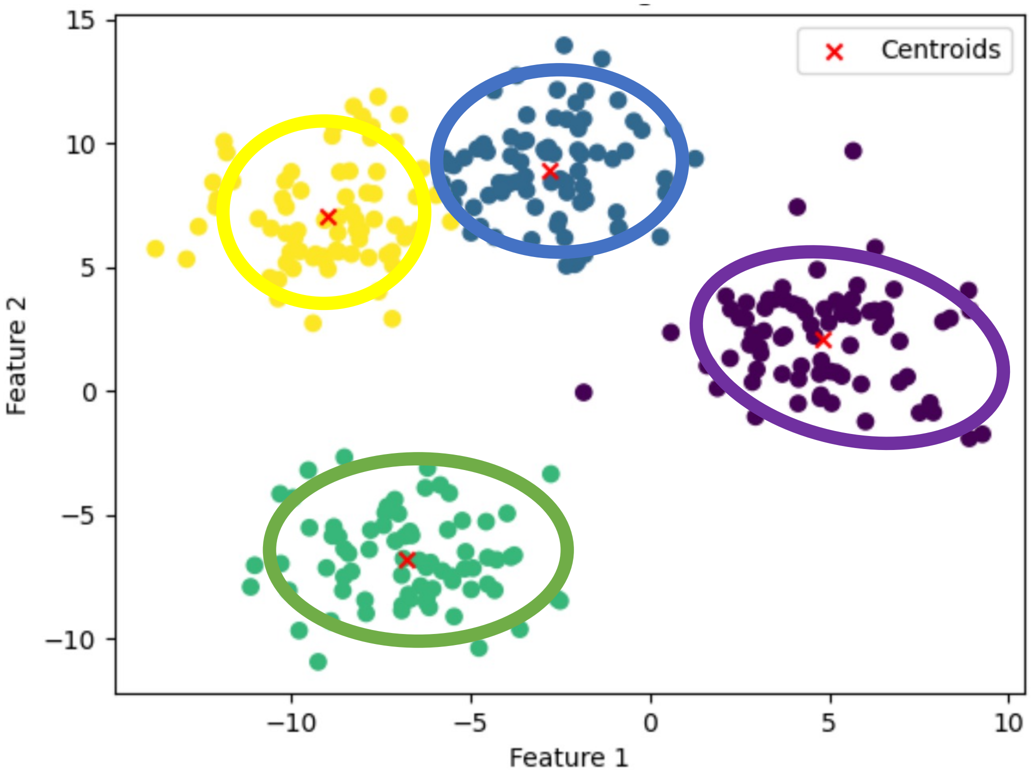


Unsupervised	Supervised
No labels provided	Labels provided
Finds structure in unlabelled data	Finds patterns in existing structure
Uses techniques such as clustering or dimensionality reduction.	Uses techniques such as regression or classification.

Let’s first learn some basic things about clustering

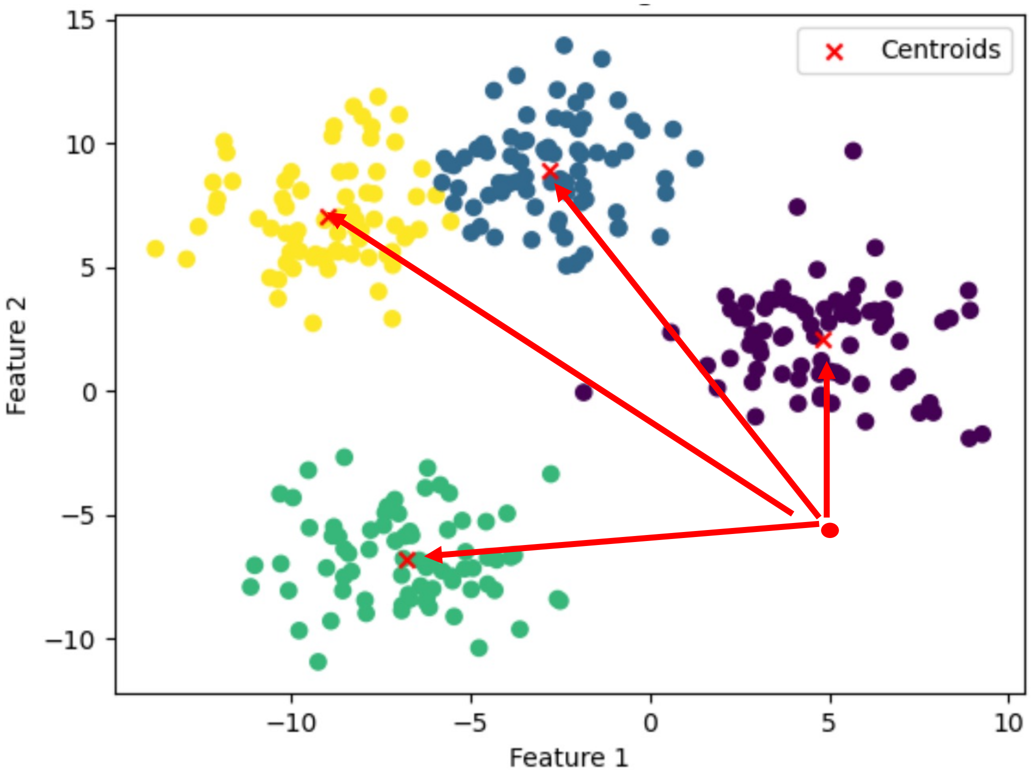
What defines a cluster?

- Clusters are defined by a center and a spread (which you can also call shape)



Assigning a Point to a Cluster

- And we assign a given point to a cluster by computing its similarity (using distance) to the centers of the clusters, and choosing the one that is the most similar



New data point

Similarity

- Central to all of the goals of cluster analysis is the notion of the degree of similarity (or dissimilarity) between the individual objects being clustered.

- A clustering method attempts to group the objects based on the definition of similarity (which is usually measured in the form of a distance from the center of a cluster) supplied to it.

Multiple Choices For Measuring Dissimilarity

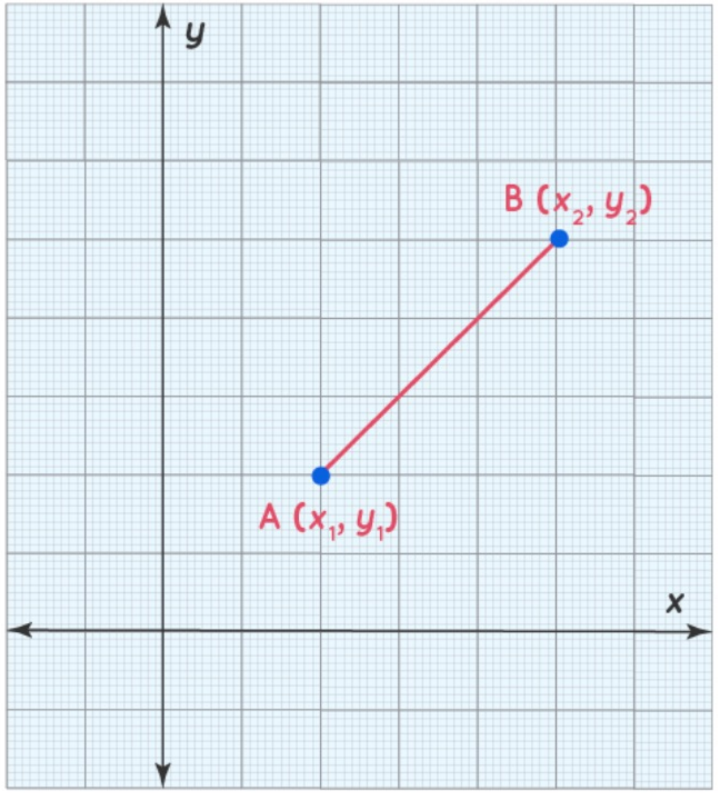
- *Euclidean distance (the most common)*
- Manhattan distance
- Correlation based distances
  - Pearson correlation distance
  - Eisen cosine correlation distance
  - Spearman correlation distance
  - Kendall correlation distance
- Etc.

https://www.analyticsvidhya.com/blog/2020/02/4-types-of-distance-metrics-in-machine-learning/

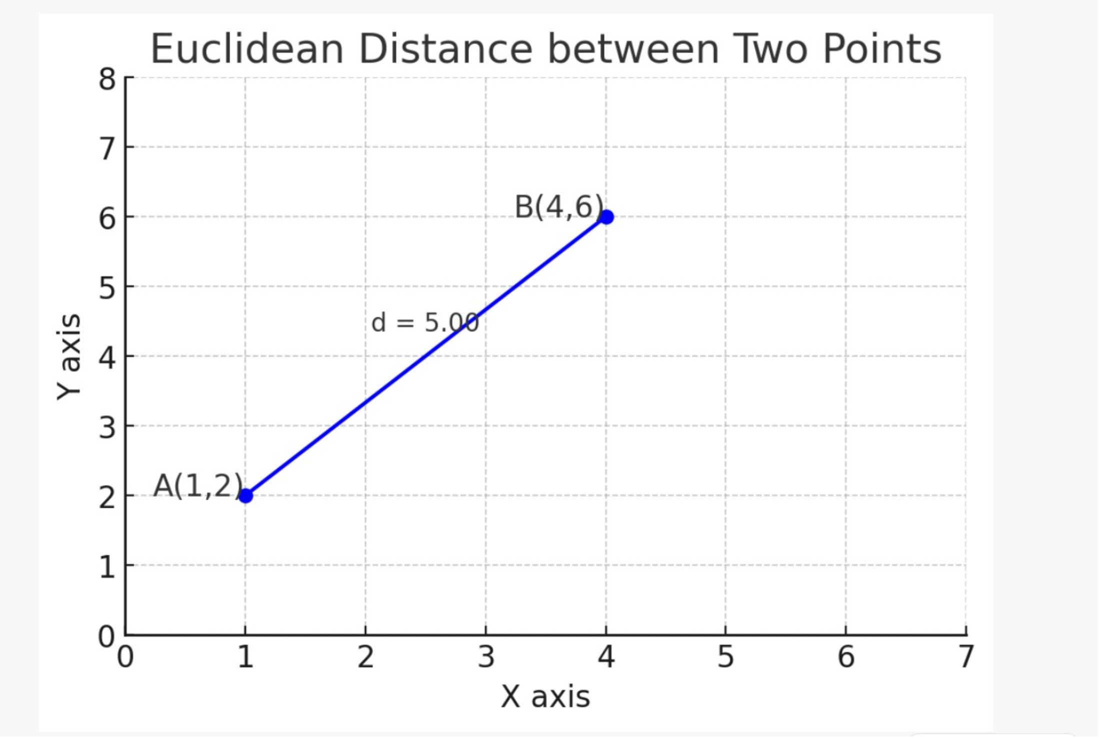
Euclidean Distance

Euclidean Distance Formula

cuemath
THE MATH EXPERT

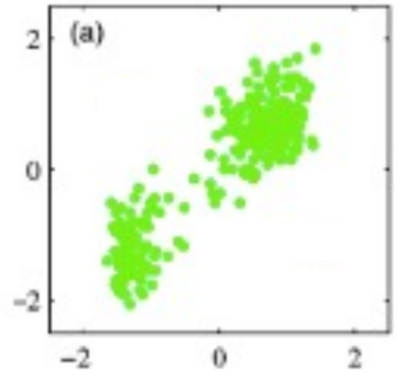


$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



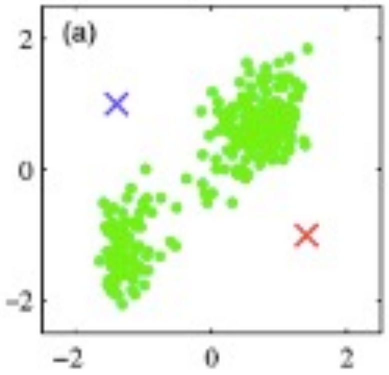
k-means Clustering

k-means Overview



- We are given unlabelled data points

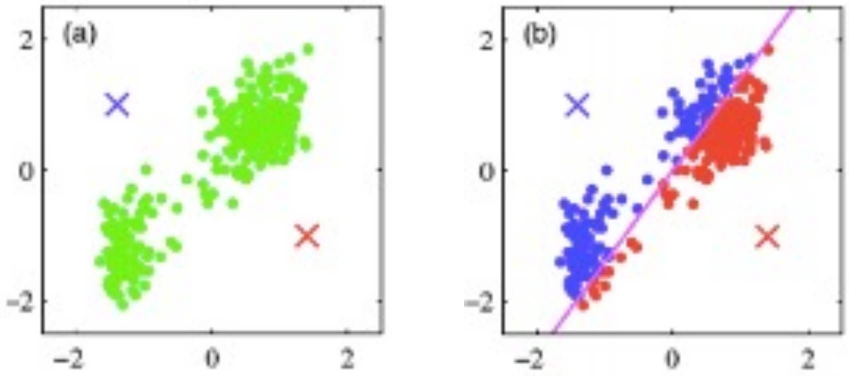
k-means Overview



- We are given unlabelled data points

1. We choose K centroids (K represents the anticipated number of distinct clusters).
2. K-means randomly positions the K centroids within the data.

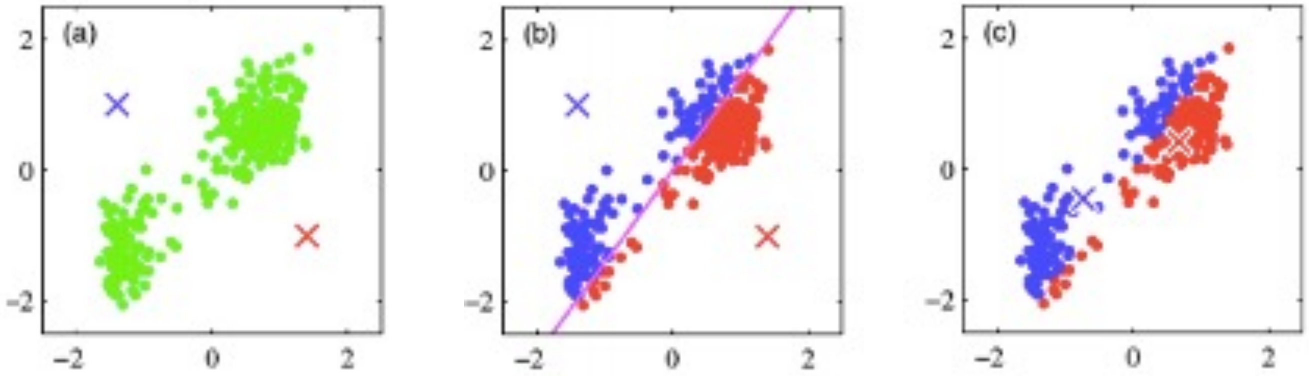
k-means Overview



- We are given unlabelled data points

1. We choose K centroids (K represents the anticipated number of distinct clusters).
2. K-means randomly positions the K centroids within the data.
3. It computes the Euclidean distance between each centroid and all the points in the data.
4. And assigns each training data point to the closest centroid, forming groups.

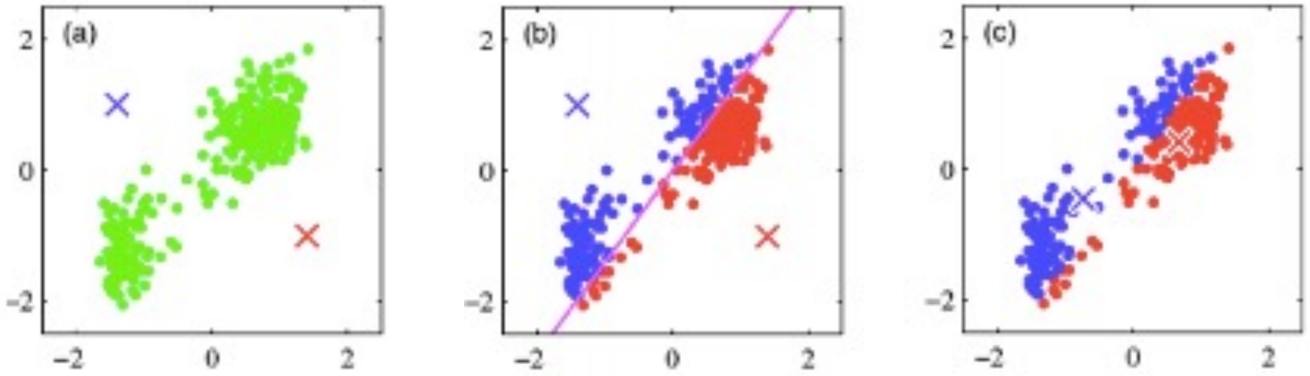
k-means Overview



- We are given unlabelled data points

1. We choose K centroids (K represents the anticipated number of distinct clusters).
2. K-means randomly positions the K centroids within the data.
3. It computes the Euclidean distance between each centroid and all the points in the data.
4. And assigns each training data point to the closest centroid, forming groups.
5. Within each group, it calculates the average data point and moves the corresponding centroid to that position.

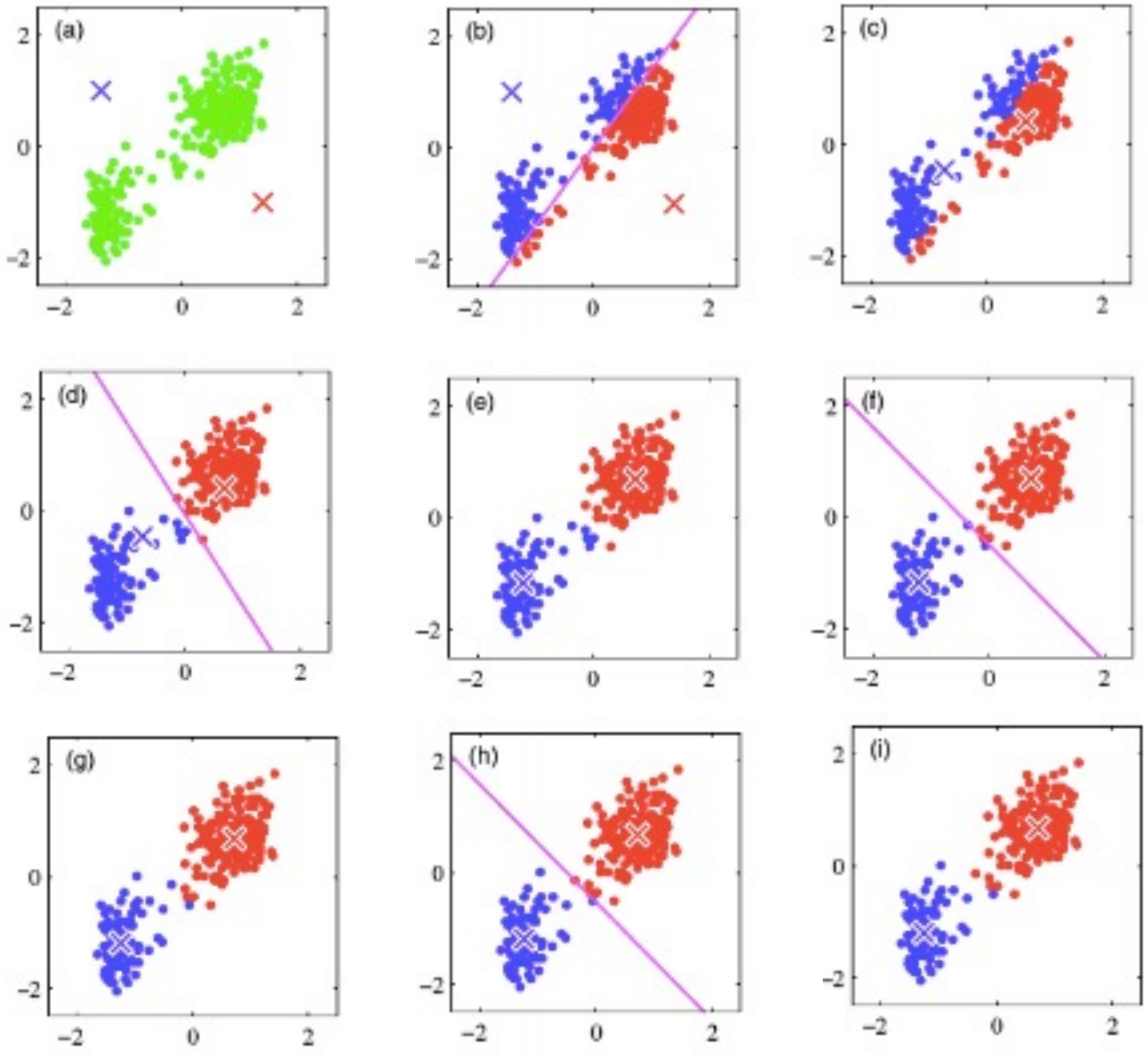
k-means Overview



- We are given unlabelled data points

1. We choose K centroids (K represents the anticipated number of distinct clusters).
2. K-means randomly positions the K centroids within the data.
3. It computes the Euclidean distance between each centroid and all the points in the data.
4. And assigns each training data point to the closest centroid, forming groups.
5. Within each group, it calculates the average data point and moves the corresponding centroid to that position.
6. The algorithm continues this process (steps 3 - 5) until convergence is reached, which occurs when there are no more changes in group membership.

k-means Overview



k-means Clustering Pseudocode

- k-means simply works as follows
  - First Step: Initialization - randomly initialize cluster centers
  - Second Step: The algorithm then iteratively alternates between the following two steps
    - ➢ Assignment Step: assign each data point to the closest center
    - ➢ Refitting Step: move each cluster center to the mean of the data points assigned to it

Let’s Formulate Clustering as an Optimization Problem

First Thing to Understand

1. Clustering takes unsupervised data (only X) as input – Let N be the total number of data points

2. And returns two things:
   - K cluster centers
   - N cluster assigmnets

Deriving Clustering Objective (1)

- Given $D = \{\boldsymbol{x}_i\}_{i=1}^{N}$ Find cluster centers $\{\boldsymbol{c}_k\}_{k=1}^{K}$ and assignments $\{\boldsymbol{a}_i\}_{i=1}^{N}$ to minimize the sum of squared distances of data points to their assigned centers

$\boldsymbol{x}_i \in \mathbb{R}^d$

$\boldsymbol{c}_k \in \mathbb{R}^d$

$\boldsymbol{a}_i \in \mathbb{R}^k$

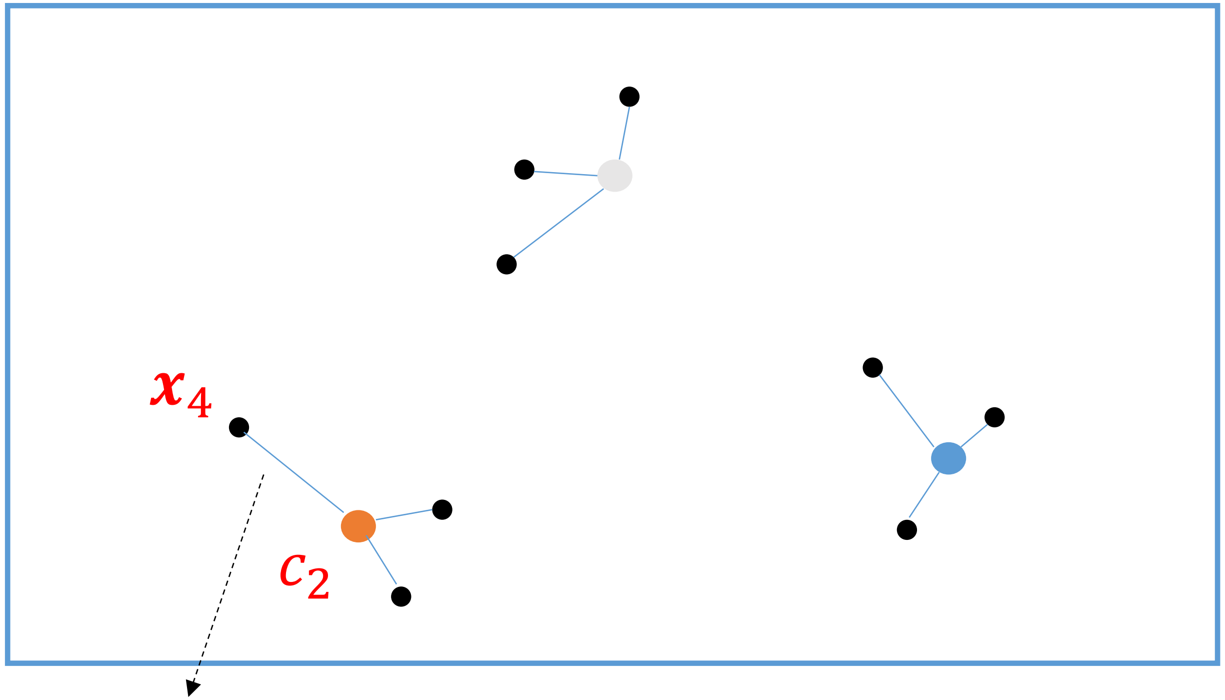
Deriving Clustering Objective (2)

- Given $D = \{\boldsymbol{x}_i\}_{i=1}^{9}$ Find cluster centers $\{\boldsymbol{c}_k\}_{k=1}^{3}$ and assignments $\{\boldsymbol{a}_i\}_{i=1}^{9}$ to minimize the sum of squared distances of data points to their assigned centers

$\boldsymbol{x}_i \in \mathbb{R}^2$

$\boldsymbol{c}_k \in \mathbb{R}^2$

$\boldsymbol{a}_i \in \mathbb{R}^3$



$\boldsymbol{a}_4 = [0, 1,0]^T$

Deriving Clustering Objective (3)

- Optimization problem:

$$\arg\min_{c_k, a_i} \sum_{i=1}^{N} \sum_{k=1}^{K} a_i^k d(c_k, x_i)$$

- Problem is hard when minimizing jointly
- But becomes easy when we fix one and minimize over the other

How to Solve the Optimization Problem? (2)

- That is:
  - **First** we fix the centers, and find the optimal cluster assignments by assigning each point to the cluster with the nearest neighbor

How to Solve the Optimization Problem? (3)

- Next:
  - For the found (and fixed) assignments, we can find the optimal cluster centers by setting each cluster's centers to the average of its assigned data points

How to Solve the Optimization Problem? (4)

- Thus

$$\arg\min_{c_k, a_i} \sum_{i=1}^{N} \sum_{k=1}^{K} a_i^k d(c_k, x_i)$$

  - We solve this optimization problem by alternating between minimizing our objective with respected to $\{\boldsymbol{c}_k\}$ and $\{\boldsymbol{a}_i\}$

  - This is called alternating minimization

k-means Clustering Pseudocode

- Given:
  - K, data $X = \{X_1, X_2, \cdots, X_N\}$
- Goal:
  - Cluster X into K clusters

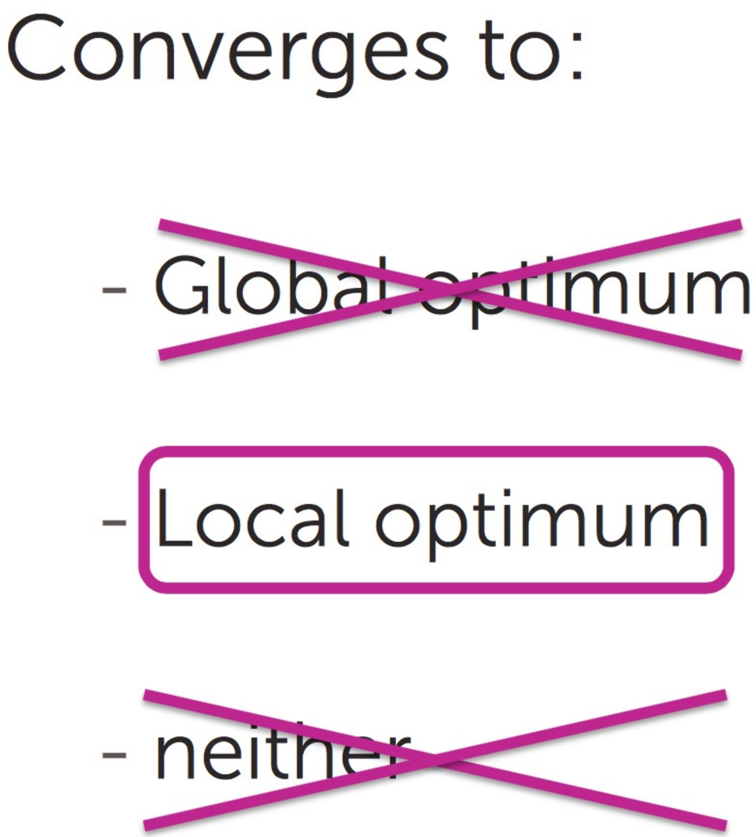
- $D(X_i, c_j)$: the between instance X_i and cluster centre c_j;

- Euclidian distance:

$$D\left(X_i, c_j\right)^2 = \sum_{k=1}^{p} \left(X_{ik} - c_{jk}\right)^2$$

```
❑ Input: $K$, data points $X_1, X_2, \cdots, X_N$
❑ Place centroids $c_1, c_2, \cdots, c_K$ at random locations
❑ Repeat until convergence:
    ▪ For each point $X_i$:
        • Find nearest centroid $c_j$:
            $argmin_j D(X_i, c_j)$
        • Assign the point $X_i$ to cluster $j$
    ▪ For each cluster $j = 1,2, \cdots, K$:
        • New centroid $c_j$ is set to the mean of all points $X_i$ assigned to cluster $j$ in previous step
❑ Stop when none of the cluster assignments change
```

Convergence of k-means

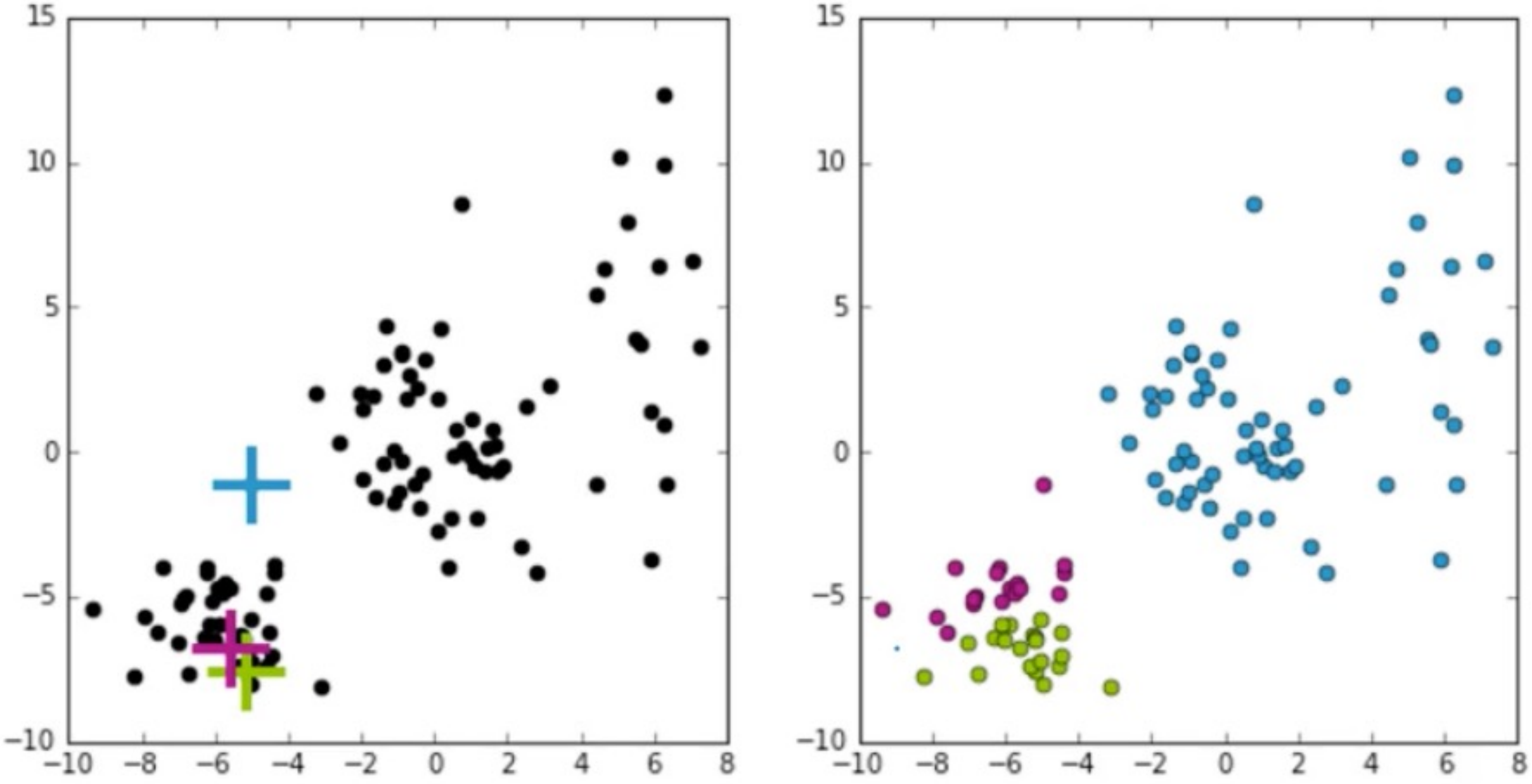


Converges to:

- ~~Global optimum~~
- Local optimum
- ~~neither~~

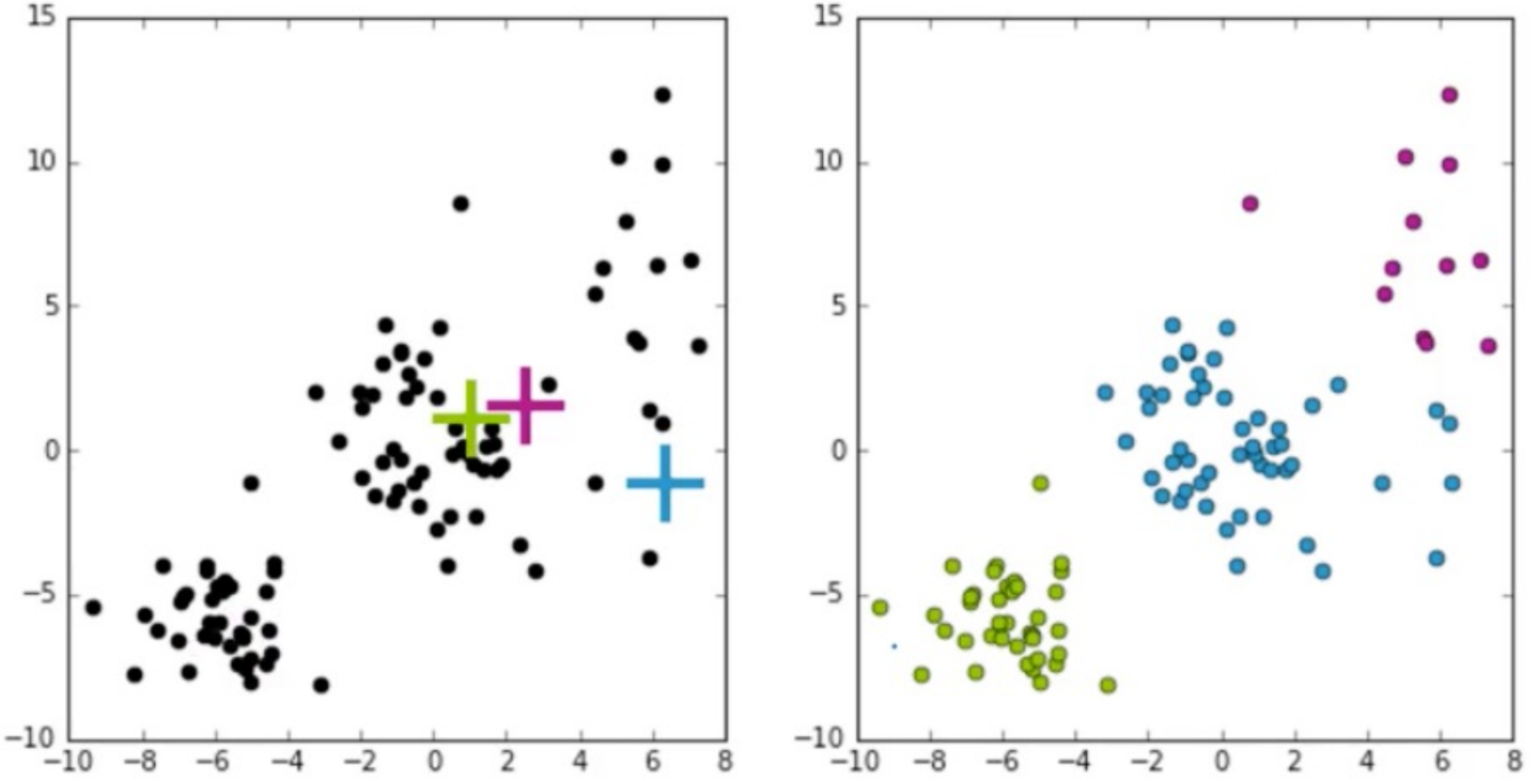
The first one does not mean that it can never converge to global optimum, it is just that there is no guarantee that it will.

Initialization Matters



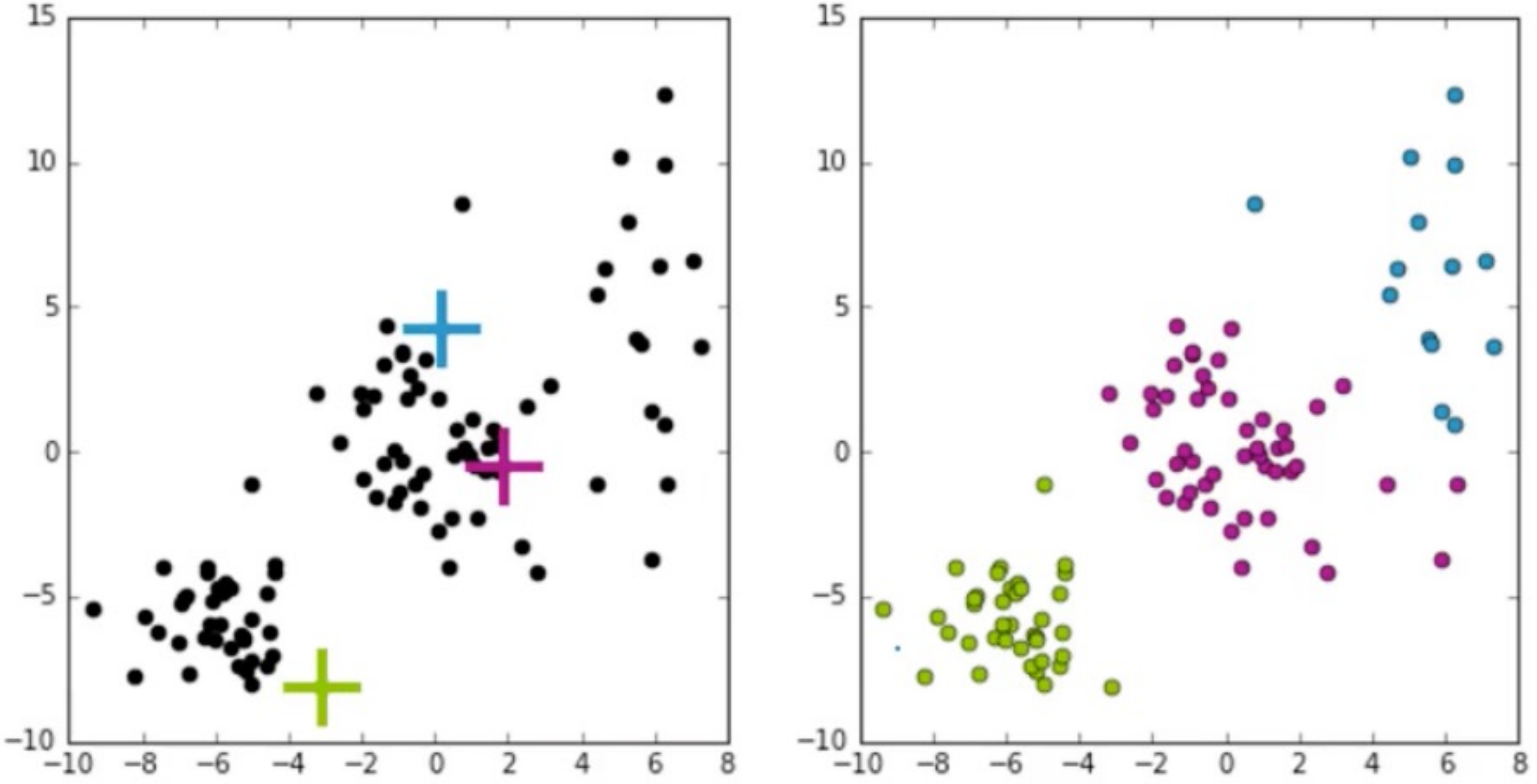
Centroids are chosen as shown in the first figure

Initialization Matters (2)



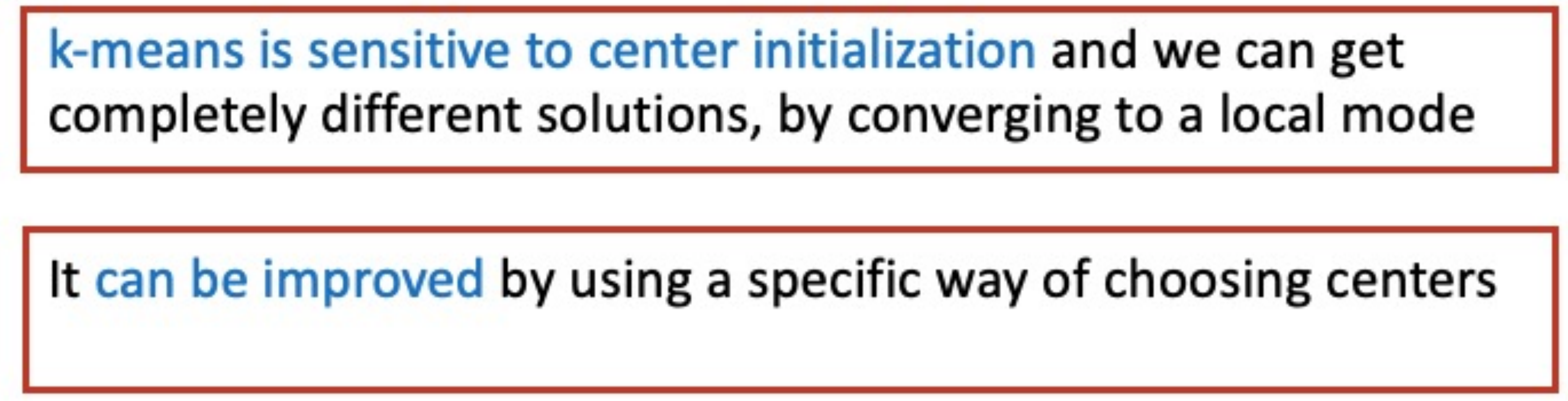
Different Initialization

Initialization Matters (3)



An even different initialization

Bottom-line



k-means is sensitive to center initialization and we can get completely different solutions, by converging to a local mode

It can be improved by using a specific way of choosing centers

k-means++

k-means++

- The intuition behind this approach is that spreading out the *k* initial cluster centers is a good thing

- The first cluster center is chosen uniformly at random from the data points that are being clustered

- After which each subsequent cluster center is chosen from the remaining data points with probability proportional to its squared distance from the existing cluster center.

How does k-means++ work?

- Let $d(x)$ denote the shortest distance from a data point to the closest center we have already chosen

- Then k-means ++ works as follows

1. Take the first center $\boldsymbol{m}_1$, chosen uniformly at random from D
2. Take the next center, choosing $\boldsymbol{x}_i$ with probability $\frac{d(\boldsymbol{x}_i)^2}{\sum_{\boldsymbol{x}_j \in X} d(\boldsymbol{x}_j)^2}$
3. Repeat step 2 until we have chosen K cluster centers

4. Proceed with the standard k-means algorithm

Example

- Flet’s say we have four data points A, B, C and D
- We have already chosen the first cetroid as point A, and need to choose the second centroid from the remaining point
- First, we compute the squared distance from each of these points to point A (since it is the nearest centroid) – Let’s say this turns out to be:

Point	Distance
B	1
C	4
D	9

Example

Point	Distance
B	1
C	4
D	9

- Now, to choose the next centroid, we assign probabilities to each of these points proportional to their distances

- The sum of the squared distance is 1+4+9 = 14, thus the probabilities will turn out to be

Point	Probability
B	1/14
C	4/14
D	9/14

Example

Point	Probability
B	1/14
C	4/14
D	9/14

- We then choose the next centroid based on these probabilities
- Point D has the highest probability in this case
- This process ensures that data points that are farther away from the existing centrioid are most likely to be chosen

k-means++ Summary

- Smart initialization is computationally costly relative to random initialization

- However, due to smart initialization, the subsequent k-means often converges more rapidly

- Tends to improve the quality of the local optimum

How to choose k for k-means?

Hyperparameters

1. Hyperparameters are the choice that we make about the model

2. Examples:
   - Polynomial Regression: Degree of the polynomial
   - k-means: Value of k (number of clusters)

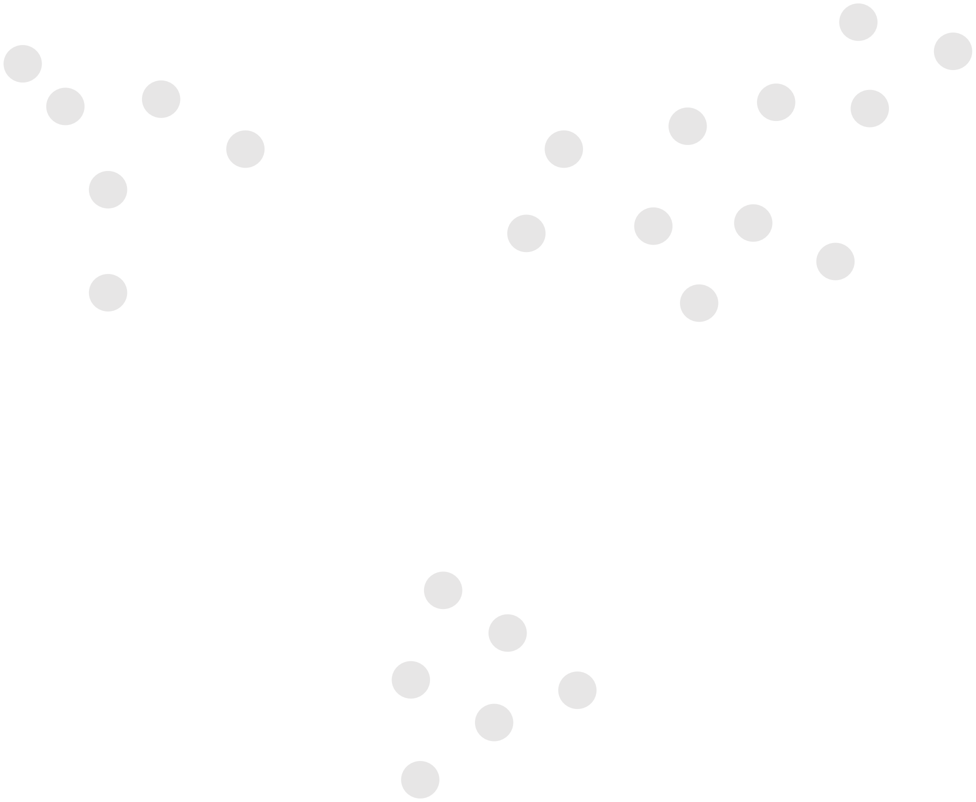
3. It is important to choose their optimum values

k-means Objective

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \boxed{\sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2}$$

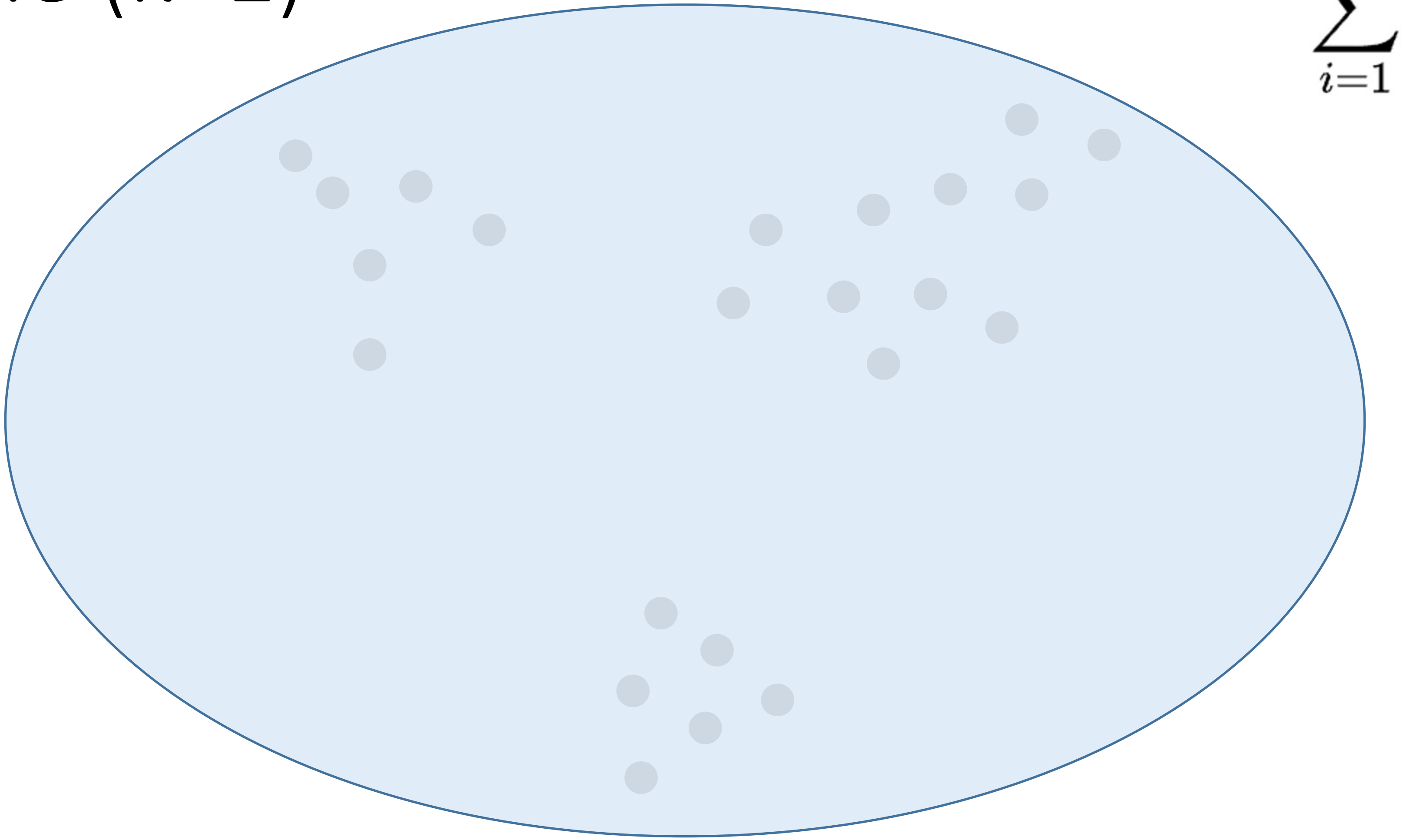
Example

$$\sum_{i=1}^{N} \sum_{k=1}^{K} a_i^k d(c_k, x_i)$$



Example (k=1)

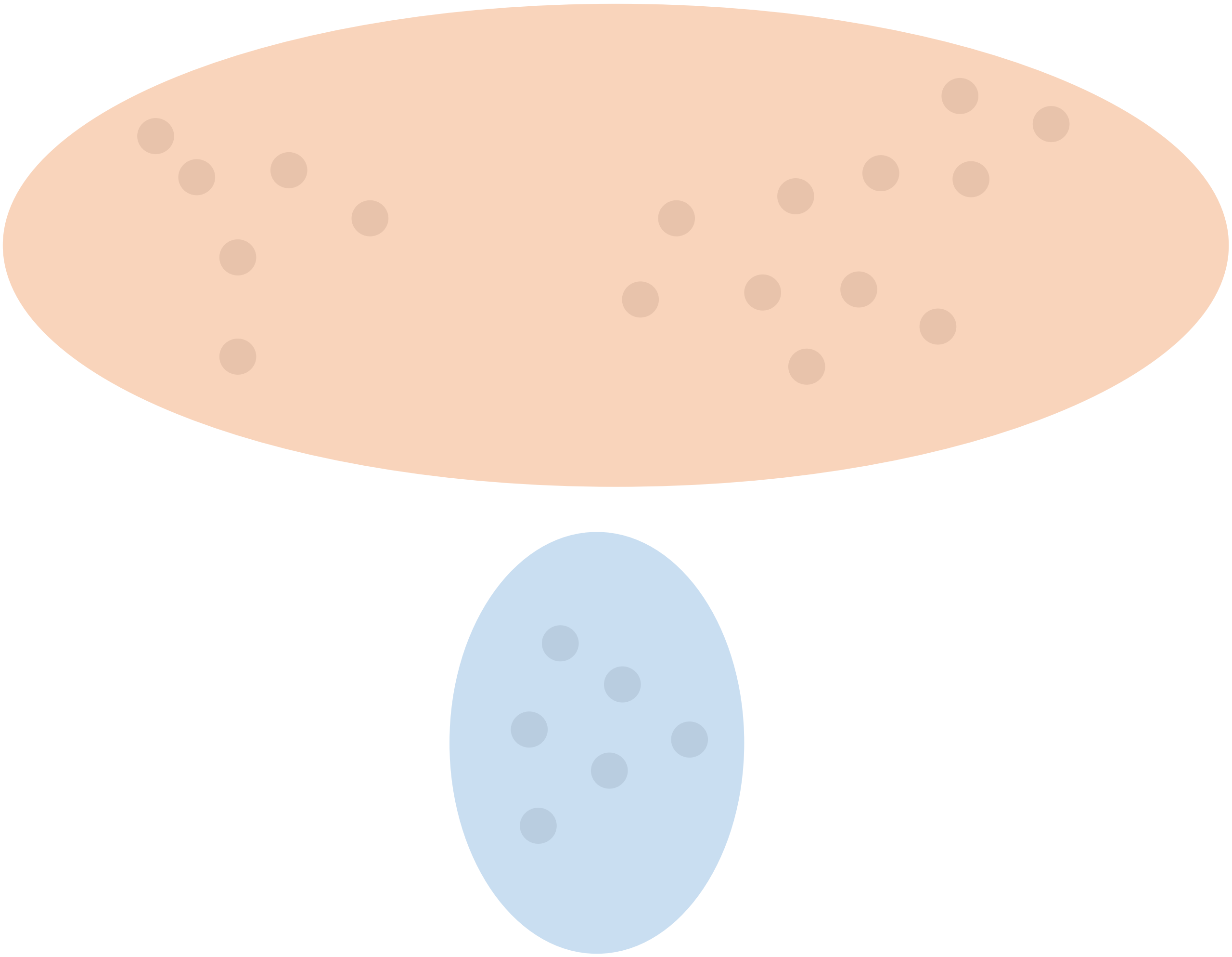
$$\sum_{i=1}^{N}\sum_{k=1}^{K} a_i^k d(c_k, x_i)$$



Center will be somewhere in the middle of this sphere, and the sum of distances will be some large positive value

Example (k=2)

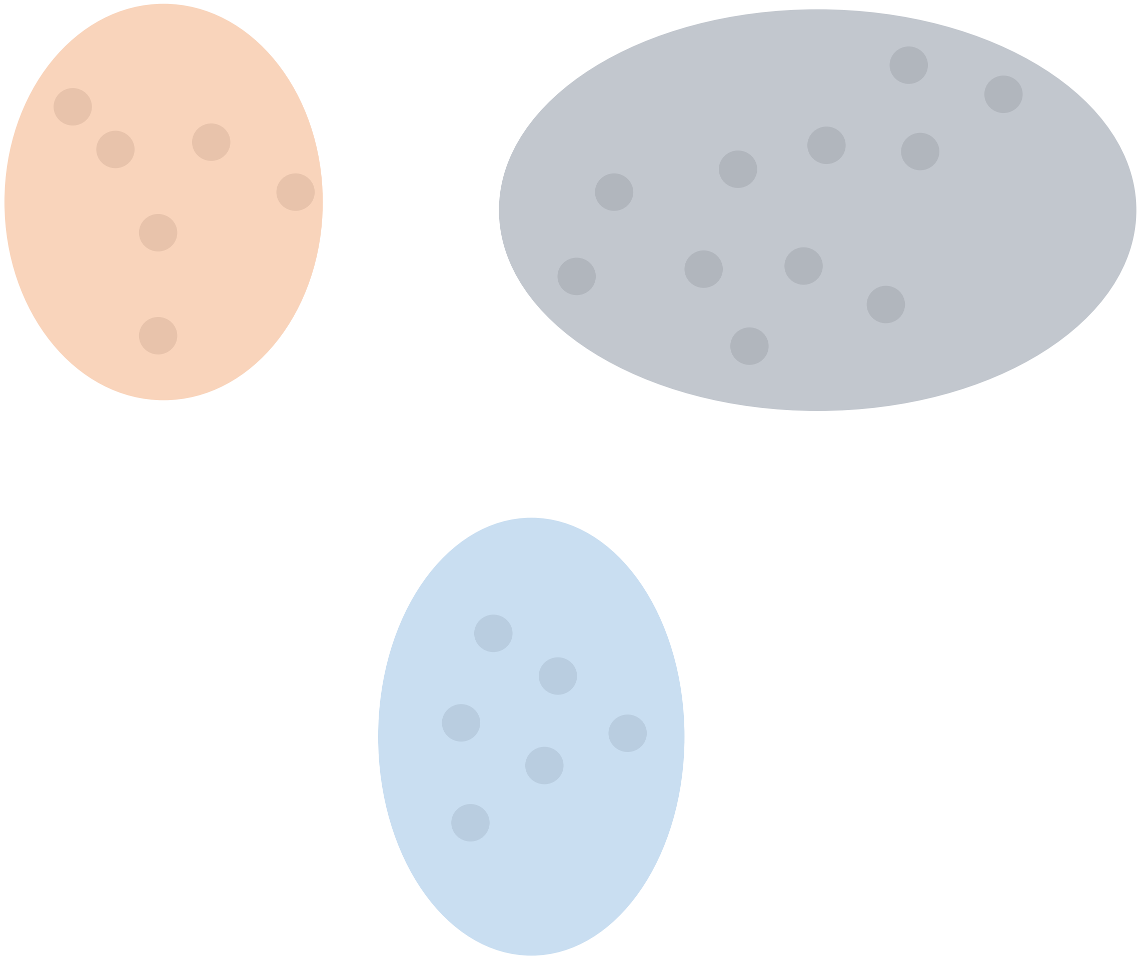
$$\sum_{i=1}^{N} \sum_{k=1}^{K} a_i^k d(c_k, x_i)$$



Each point is now more close its center as compared to the previous case, so the sum of distances would decrease

Example (k=3)

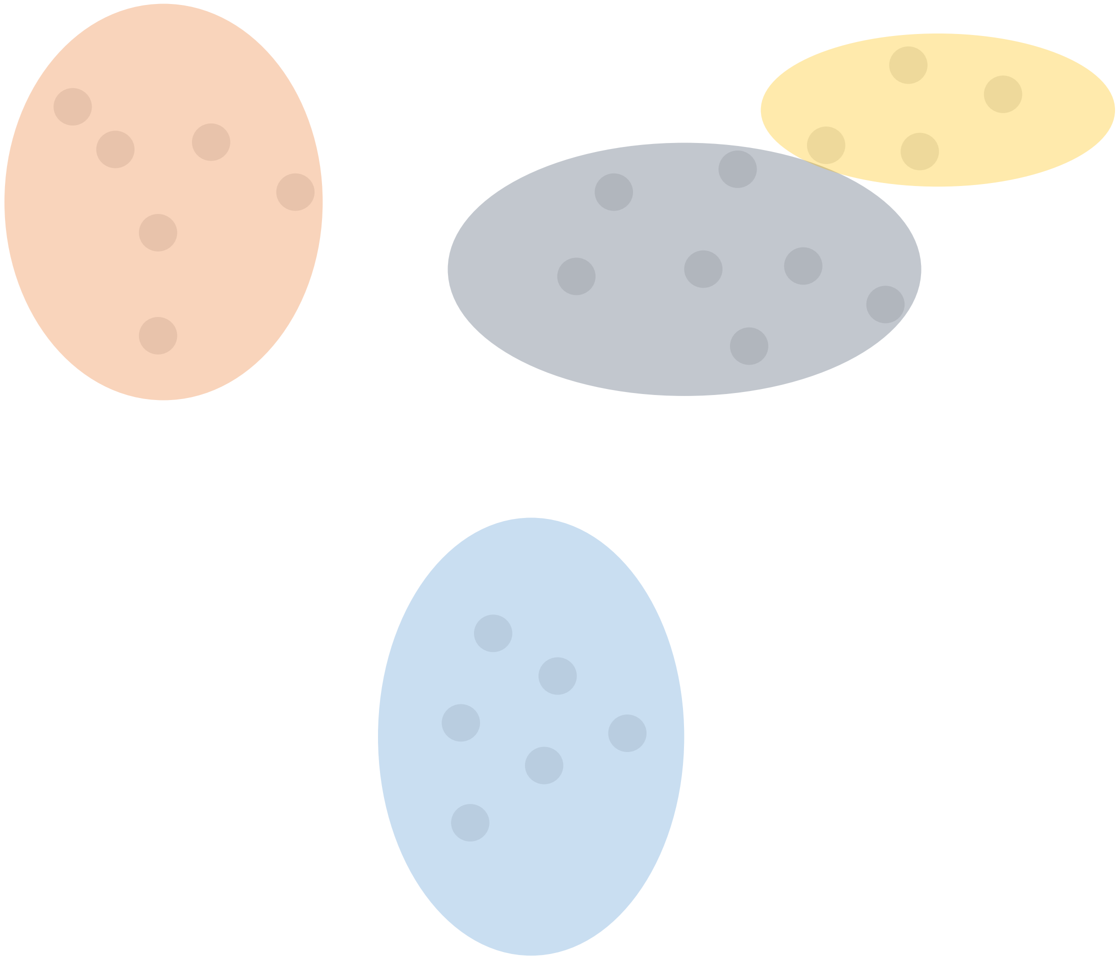
$$\sum_{i=1}^{N}\sum_{k=1}^{K} a_i^k d(c_k, x_i)$$



The sum would decrease even futher

Example (k=4)

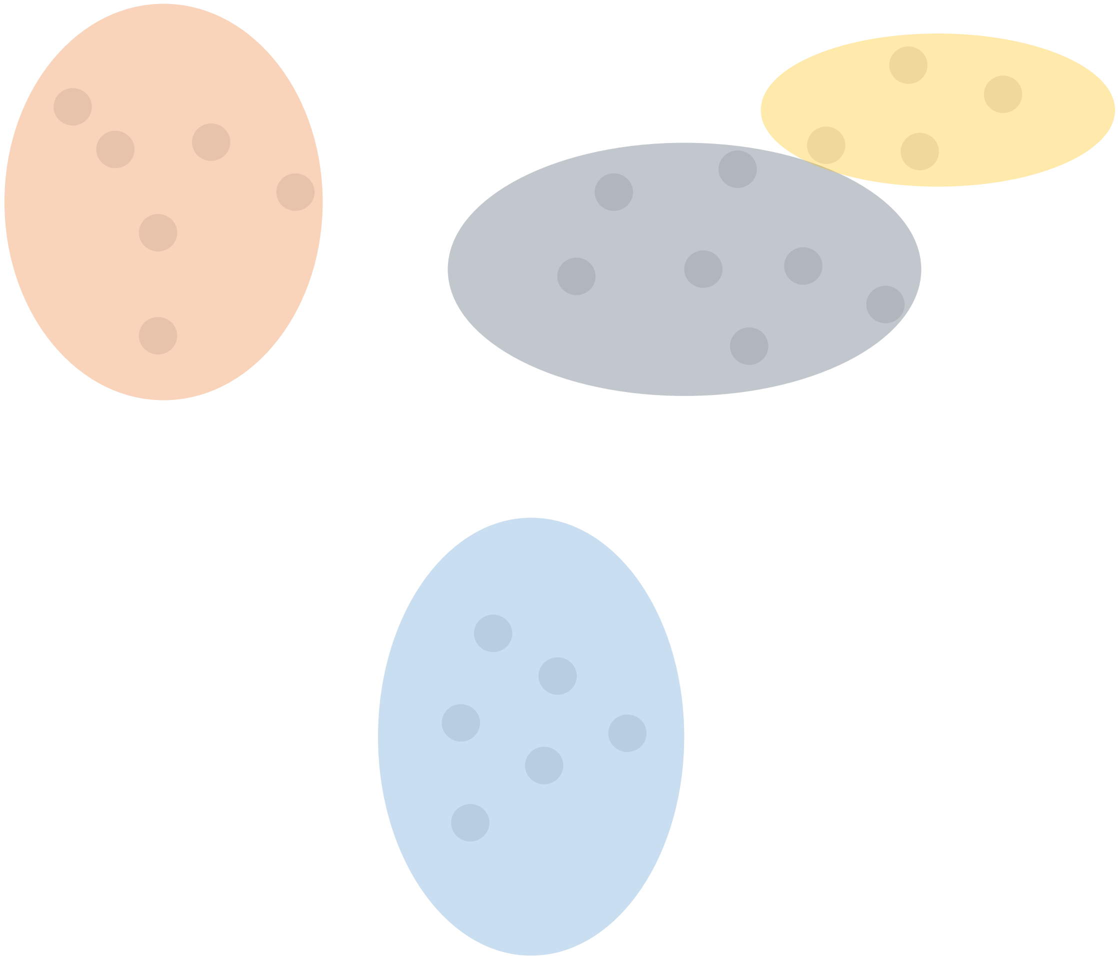
$$\sum_{i=1}^{N}\sum_{k=1}^{K} a_i^k d(c_k, x_i)$$



Even further descreas!!!

Example

$$\sum_{i=1}^{N}\sum_{k=1}^{K} a_i^k d(c_k, x_i)$$



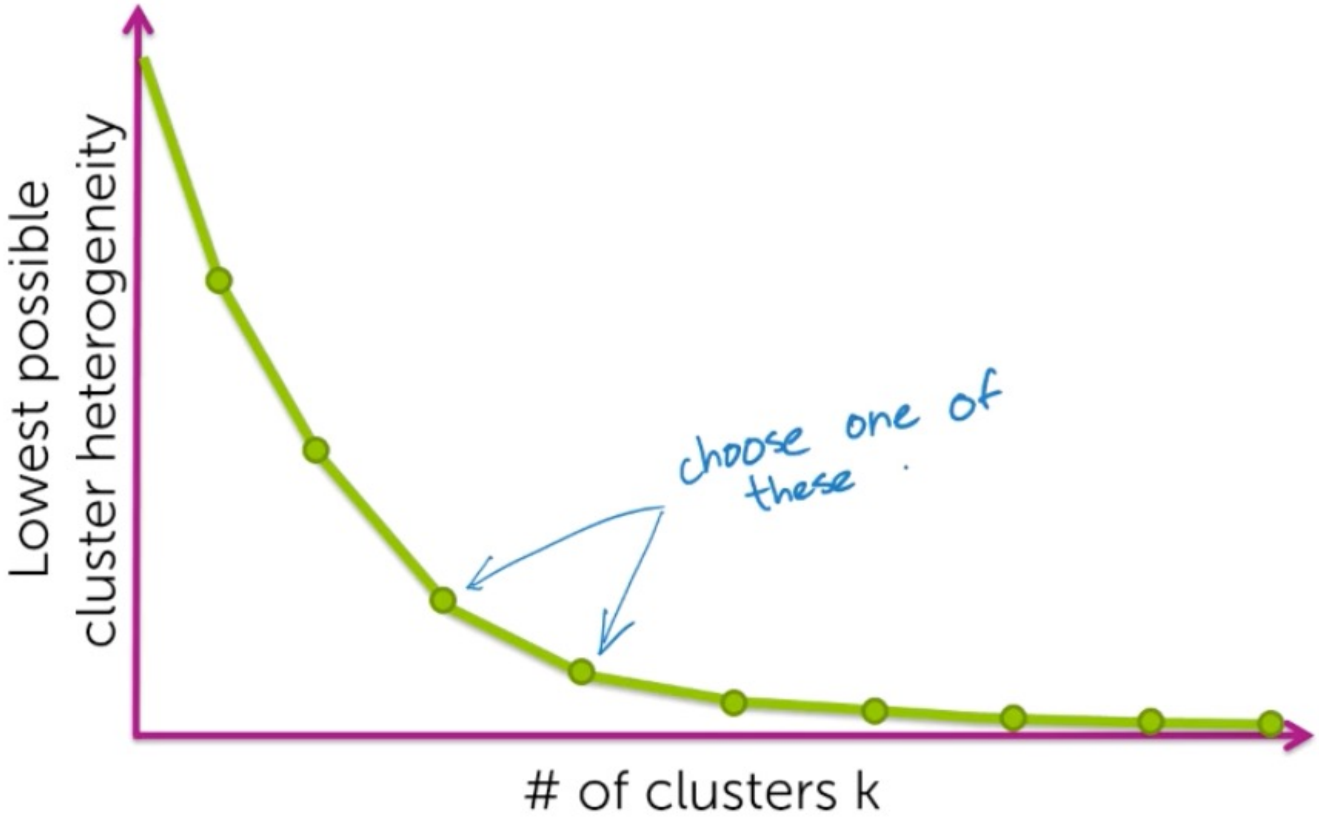
Now, take a moment and think about the case where set k = N

The sum will become 0!!

Overfitting in Clustering

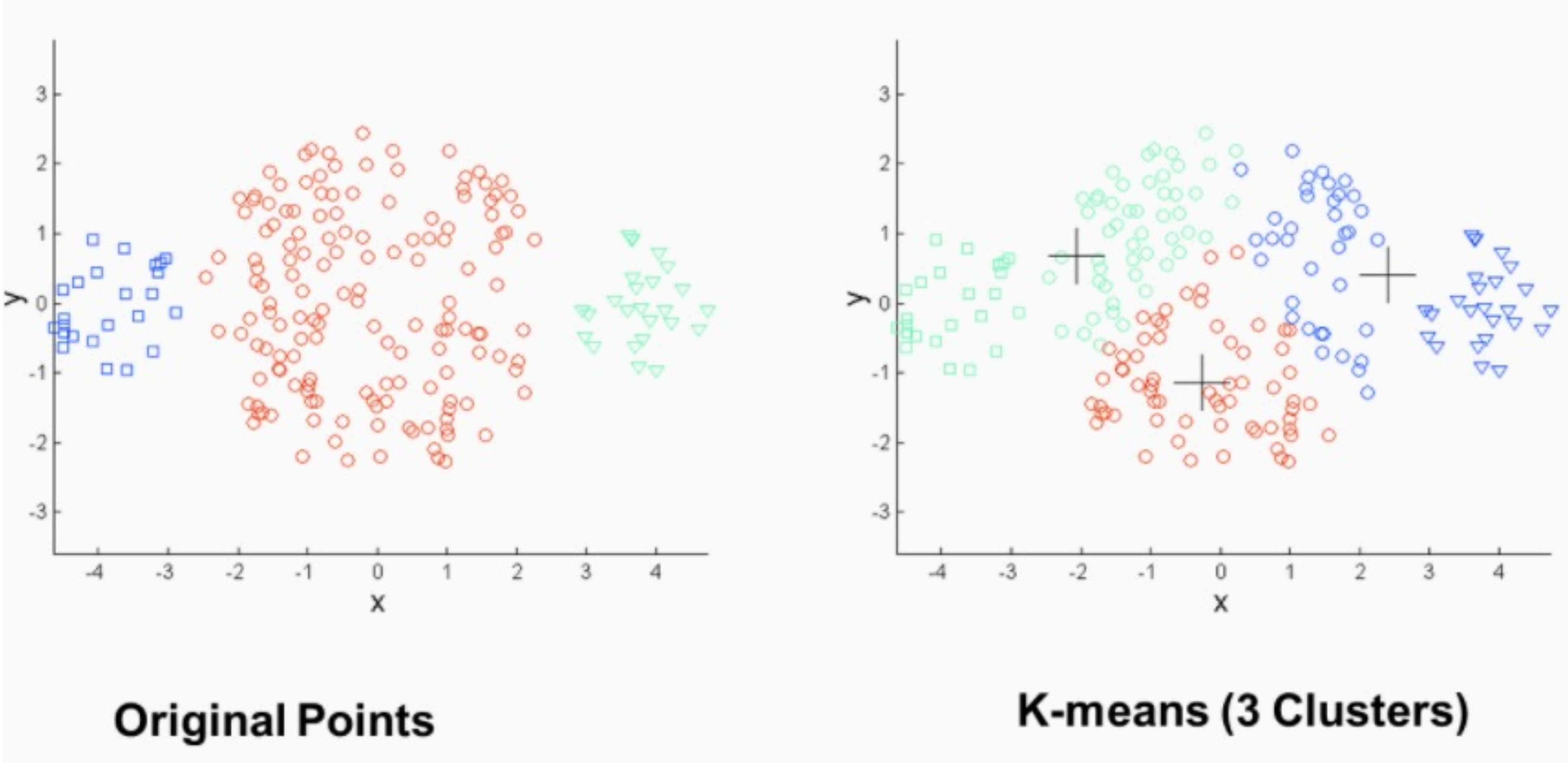
- Remember, we are in unsupervised learning.
  - In this case, overfitting would mean that we did not learn the true structures in the data, instead we ended up memorizing the position of the sample in the space.

Choosing k – The Elbow Method

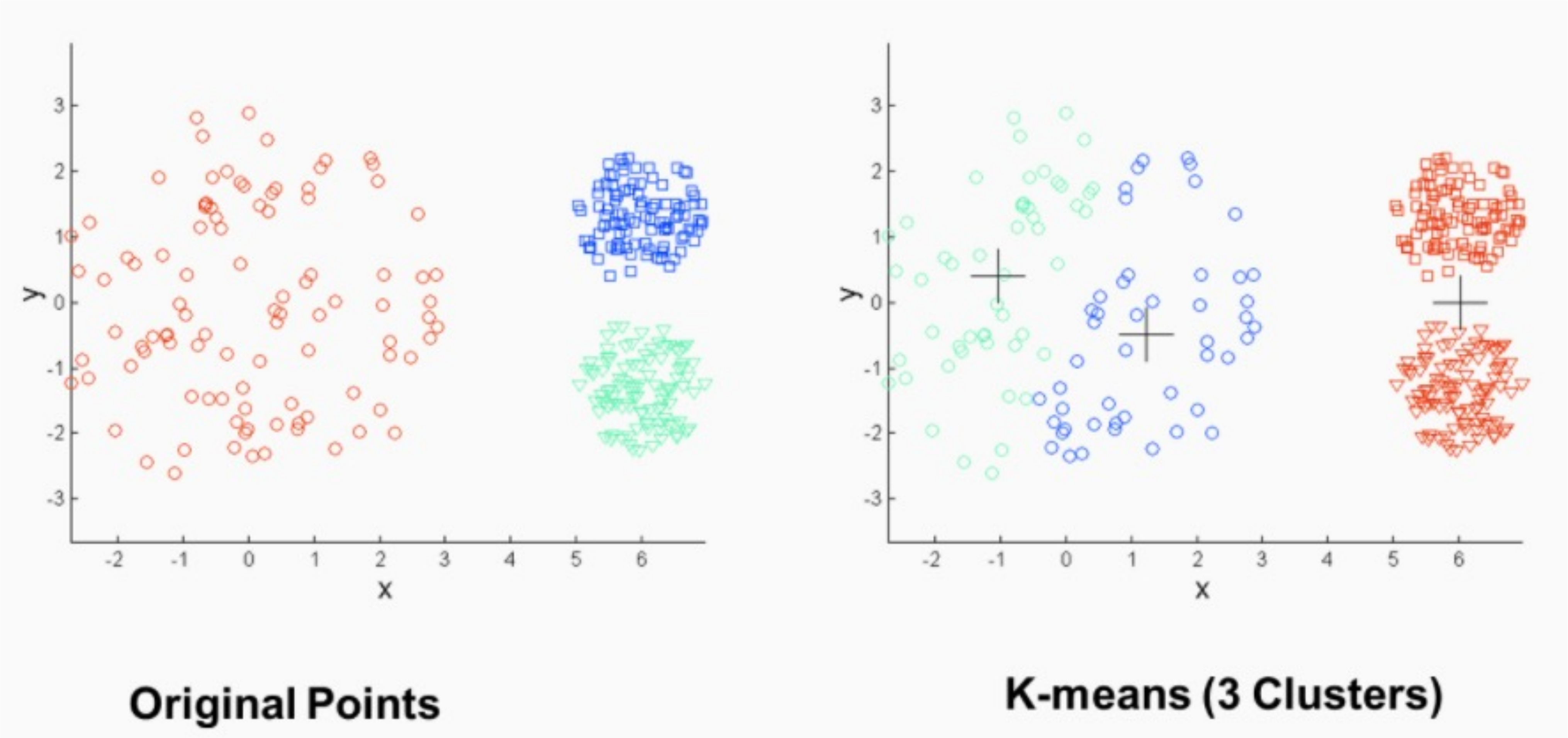


Elbow of the curve

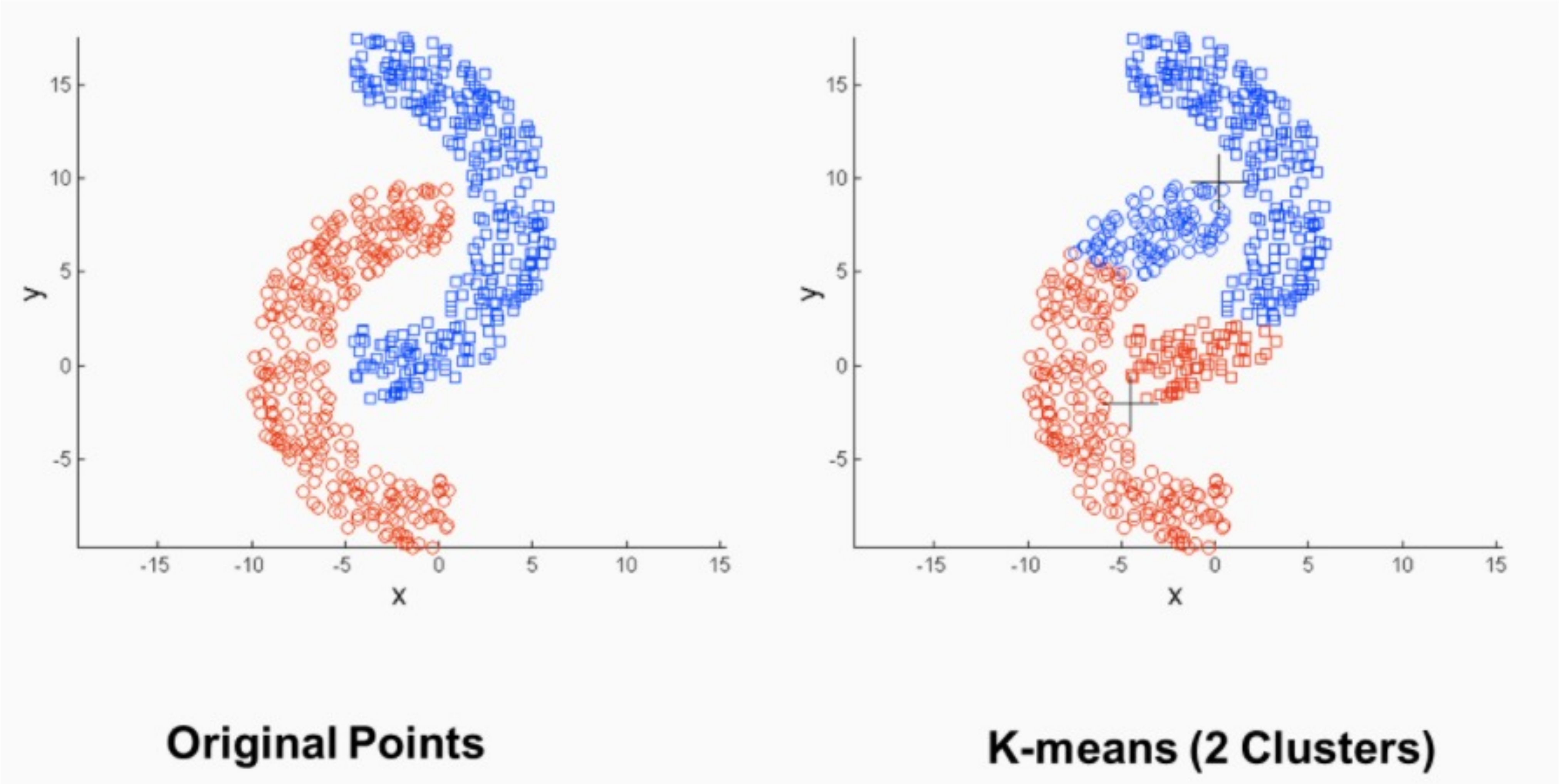
Issues with k-means (1)



Issues with k-means (2)



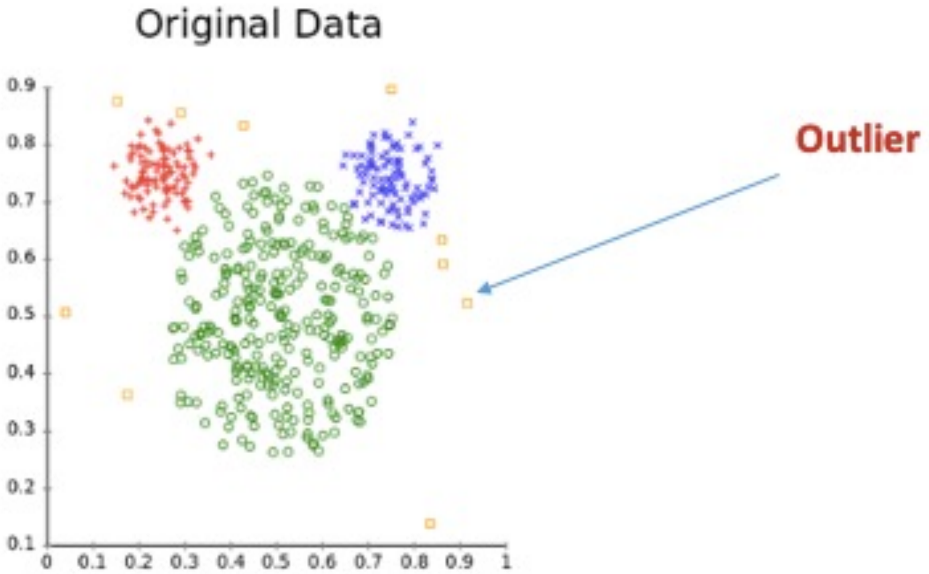
Issues with k-means (3)



Original Points

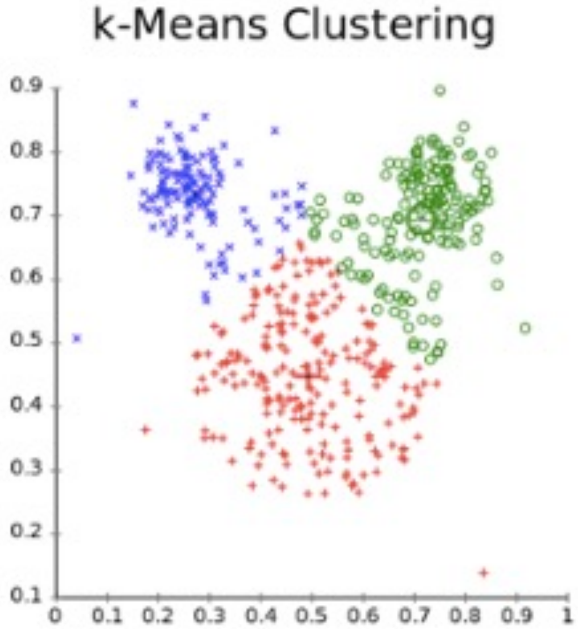
K-means (2 Clusters)

Issues with k-means (4)



Original Data

Outlier



k-Means Clustering

Issues with k-means (Summary)

1. Not knowing the optimum value of K

2. Also, k-means clustering does not work well when

   - We want to discover clusters of varying sizes, densities and shapes

   - We do not want to include noisy points (outliers) into any clusters

Are there any Alternatives?

- Many

- But the two that you will be introduced later (level 6) include
    - Hierarchical Clustering
    - Density Based Clustering (DBSCAN)

Summary

1. Clustering: discovering structures/groups in the data
    - An unsupervised learning problem
2. Principle on which clustering works
3. K-means clustering
    - Its objective function
    - How is it motivated
    - How k-means solves it optimization problem
    - Convergence of k-means
    - K-means++
4. Limitations of k-means