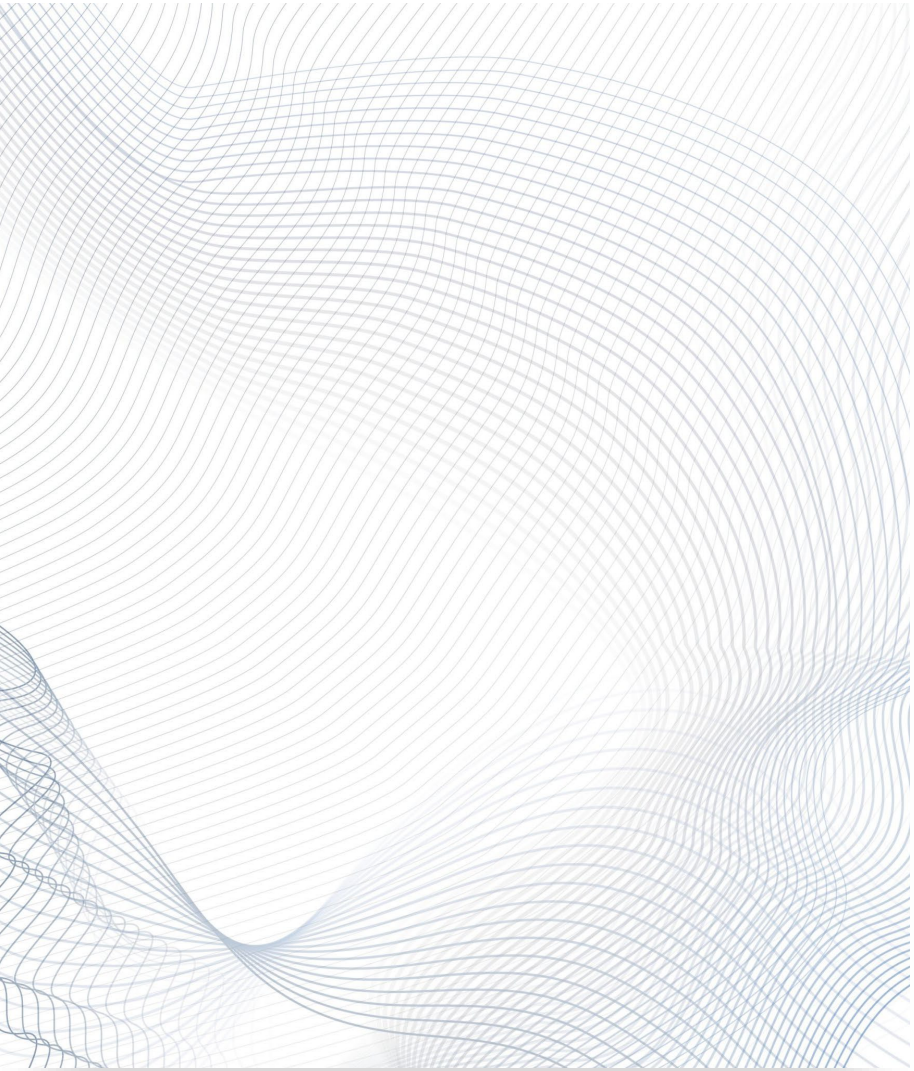


Clustering Methods in Machine Learning

Outline

- What is cluster analysis?
- Background
  - Unsupervised
  - Distance functions
- Description and Explanation of algorithms
  - Partitional algorithms: K-means
    - An example for K means clustering

String dataset example

	session	start	end	customer	ordernum	value	products
1	123@345	2015-11-02 18:34	2015-11-02 21:45	c1	13	45£	milk1, eggs1, butter1, diapers1, toilet_paper1, cheese1, hamburgers1
2	823@472	2015-11-08 19:02	2015-11-08 22:57	c1	14	50£	milk1, eggs1, diapers1, beer1, carrots1, potatoes1, ckicken1, dishwasher1
3	492@487	2015-11-16 21:13	2015-11-17 08:21	c1	15	44£	milk1, eggs1, toilet_paper2, pork1, rice1, jam1, crackers1
4	109@232	2015-11-22 08:40	2015-11-22 19:33	c1	16	48£	soyamilk1, eggs1, diapers1, beer2, bread1, beef1, water1
5	299@128	2015-11-29 19:14	2015-11-29 21:48	c1	17	75£	milk1, eggs1, butter1, toilet_paper1, beer3, salmon1, lamb1
6	564@342	2015-12-05 18:51	2015-12-05 22:15	c1	18	43£	milk1, eggs1, diapers1, beer3, ham1, jam2, crackers1, pizza1
7	345@761	2015-11-05 16:25	2015-11-05 20:02	c2	1	43£	juice1, pasta1, pet_food1, detergent1, bin_bags1, bananas1
8	519@344	2015-11-25 17:03	2015-11-25 19:23	c2	2	65£	wine1, pet_food2, apples1, pizza2, soup1, bread2
9	984@571	2015-11-10 14:00	2015-11-10 14:02	c3	1	152£	wine1, wine2, wine3, wine4, wine5
10	711@213	2015-11-04 12:31	2015-11-04 13:11	c4	6	77£	milk2, diapers2, wine2, beer4, brocolli1, salad1, cookies1, fish_fingers1, wipes1
11	876@543	2015-11-09 10:43	2015-11-09 12:01	c4	7	80£	milk2,beer2, rice2, chicken2, tomatoes1, juice1, kitchen_roll1, lego_toy1
12	357@975	2015-11-14 11:23	2015-11-14 13:45	c4	8	75£	milk2, diapers2, beer3, carrots2, spinach1, hamburgers2, chips1, toilet_paper2
13	234@234	2015-11-20 08:50	2015-11-20 10:20	c4	9	72£	wine2, pasta2, meatballs1, tomatoes1, ham1, jam2, juice2, apples2, tissues1
14	654@432	2015-11-26 10:34	2015-11-26 18:21	c4	10	67£	milk2, eggs2, beer3, pork2, onions1, mushrooms1, potatoes2, bananas1
15	715@457	2015-12-03 11:03	2015-12-03 12:53	c4	11	91£	milk2, diapers3, beer4, wipes1, salmon2, chicken2, tomatoes1, carrots1, toilet_paper2
16	135@790	2015-11-02 21:30	2015-11-02 23:15	c5	6	30£	diapers, beer, pet_food
17	032@822	2015-12-03 22:09	2015-12-04 00:23	c5	7	20£	diapers, pet_food
18	388@554	2015-11-12 15:54	2015-11-13 21:23	c6	13		
19	432@987	2015-12-02 18:19	2015-12-04 22:59	c6	14		
20	910@734						

Clustering

Do c2 and c5 belong to the same group? (pet owners)

What is Cluster analysis?

- “Finding groups of objects such that the objects in a group will be similar (or related) to one another and different form (or unrelated to) the objects in other groups.” [Tan, 2014]

- No a priori class labels
- No knowledge of what is “correct”
- Attempt to understand how different data points are related to each other
- Greater similarity in a group, greater difference between groups -> better clustering

Background

- Supervised have access to class-labels
- Unsupervised doesn't
  - train a model similar to a supervised model
  - need ground truth
- Clustering is the partitioning of data into subgroups
  - According to e.g. a distance function
- Two primary settings:
  - Predictive clustering – to build a model that can predict the cluster membership of new, unseen data points.
  - Descriptive clustering – to understand the structure or patterns within the given dataset. It provides a thorough description of the clusters and the relationships between them, often for the purpose of exploration or interpretation.

Clustering vs. classification

- Clustering
  - No prior knowledge
  - Do not know the number and the meaning of clusters
  - Aims at discovering them
  - Sometimes referred to as unsupervised classification

- Classification
  - New unlabeled objects are assigned a class label using a model developed from objects
  - With known class labels.
  - Supervised learning

Cont.

Supervised learning

- An approach in which the model is trained using labeled data, i.e., input data paired with the correct output.
- Characteristics:
  - Guided Learning: Models learn from past data to predict outcomes for unseen data.
  - Requires Labels: Data must be labeled, often manually, to provide a "ground truth" for training.

Unsupervised learning

- An approach where the model is trained using data without any explicit labels, aiming to discover hidden patterns.
- Characteristics:
  - Discovery-Based: Often used to find patterns, clusters, or associations in data.
  - No Ground Truth: Doesn't rely on predefined labels; the model explores the data structure on its own.

Background

Hard cluster approaches

- Each data point belongs to one and only one cluster.
  - Exclusive Membership: Data points are definitively assigned to a single cluster.
  - Clear Boundaries: There's a distinct separation between clusters, with no overlap.
  - Examples: K-means, Hierarchical clustering

Soft cluster approaches

- A data point can belong to multiple clusters with varying degrees of membership.
  - Probabilistic Membership: Each data point has a probability or degree of belonging to each cluster.
  - Allows Overlap: Data points can have partial membership across several clusters.
  - Examples: Fuzzy C-means clustering, Gaussian Mixture Models (GMM)

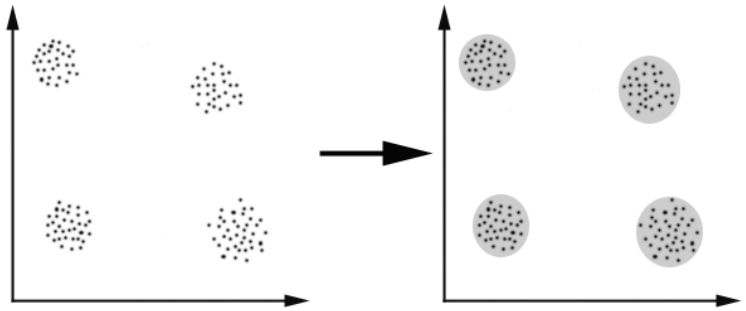
An example

Streaming Service Recommendation System

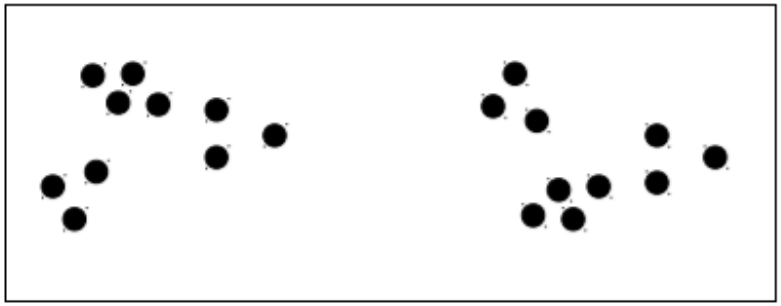
- Scenario: Consider a user who watches movies or listens to music on a streaming platform. The platform wants to recommend content based on the user's interests.
- Data Points: Here, each movie or song can be seen as a data point.
- Soft Clustering: Instead of categorizing a movie or song into a single genre or category (as in hard clustering), soft clustering allows for the possibility that a movie or song can belong to multiple genres with varying degrees of membership. For instance, a movie might be 70% action, 20% romance, and 10% comedy.
- User Profiles: Based on viewing or listening history, a user might have a preference profile like 60% action, 30% romance, and 10% comedy.
- Recommendations: Using the soft clustering approach, the recommendation engine can suggest movies or songs that align closely with the user's profile.

Clustering

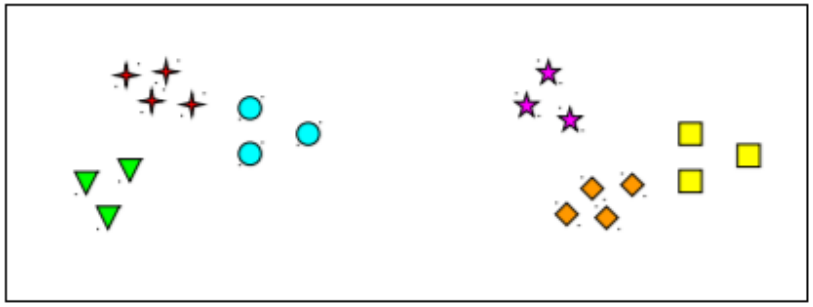
- Clustering can be considered the most important *unsupervised learning* problem
- The goal of clustering is to uncover hidden patterns or structures in the data by organizing similar data items into meaningful groups or clusters.



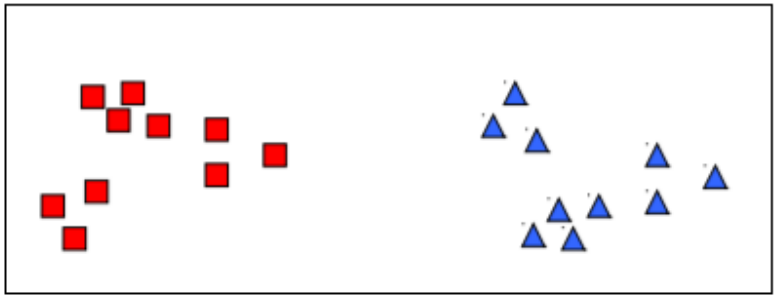
Clustering



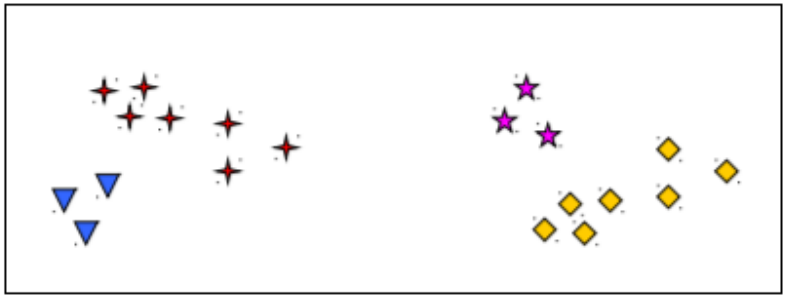
How many clusters?



Six Clusters



Two Clusters



Four Clusters

Applications of cluster analysis

- ***Marketing***: helps businesses segment their customers based on purchasing behavior, demographics, and other properties. This can inform targeted marketing campaigns, product recommendations, and customer retention strategies.
- ***Biology***: classification of plants and animals given their features.
- ***Libraries***: Clustering can help in the categorization of books based on topics, authors, publication dates, and other criteria, making the organization and retrieval processes more efficient.
- ***Insurance***: By clustering policyholders, insurance companies can identify risk groups, set premium amounts, and detect unusual patterns that might indicate fraud.
- ***City-planning***: Grouping houses based on type, value, and location can help city planners make decisions about infrastructure development, zoning, and urban renewal projects.
- ***Earthquake studies***: clustering observed earthquake epicentres to identify dangerous zones;
- ***WWW***: document classification; analyzing web logs can help in understanding user behavior and improving website design and content delivery.

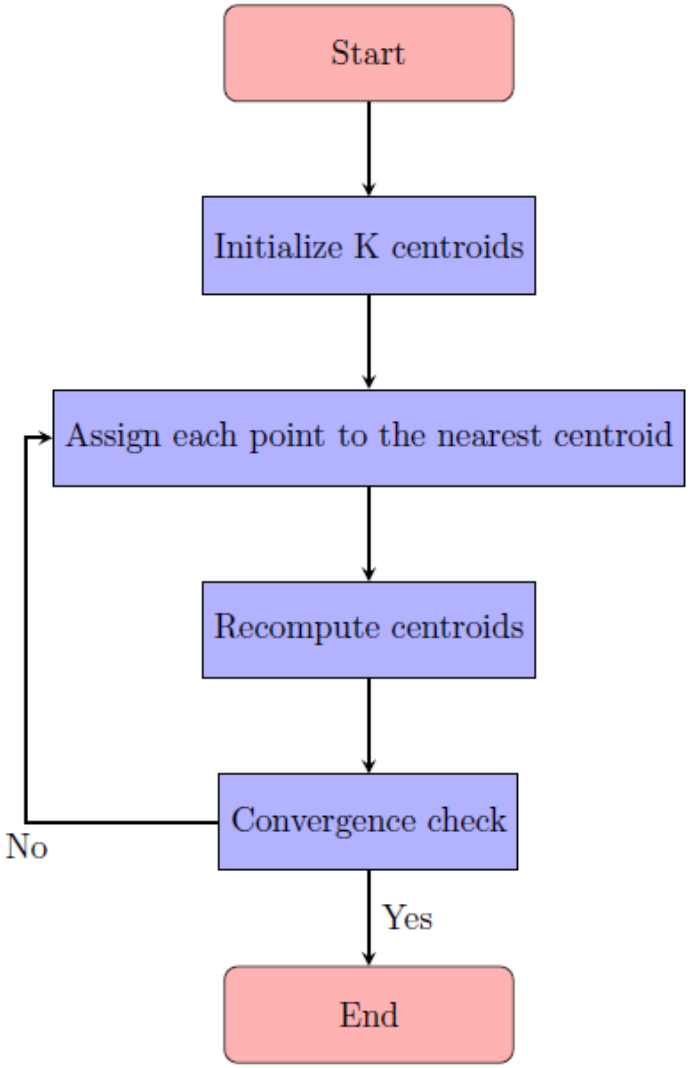
Example algorithms

- Most algorithms assume there are *k* number of clusters
- Often uses a distance function to compare similarity of instances
  - Defines a distance between two points in a set
- Distance function is dependent on the data and problem
  - Euclidian distance is very common (clustering problems)
  - Jaccard distance (binary or categorical data e.g. text data in document clustering)
  - Manhattan distance (where the grid-based distance is meaningful)
- The choice of algorithm depends on the problem
  - Each algorithm is suited for different areas, or different problems
  - Each algorithm is suited for different types of data
- Important that the choice of algorithm is carefully considered
  - Algorithm affects the clustering solution

K-means Overview

- An unsupervised clustering algorithm that classifies data into clusters without prior knowledge of the classes.
- "*K*" stands for number of clusters, it is typically a user input to the algorithm; some methods can be used to automatically estimate *K. (Elbow Method and the Silhouette Method)*
- *K*-means algorithm is iterative in nature
- KMeans can get stuck in a local minimum. This is why the algorithm is sometimes run multiple times with different initializations to obtain a better clustering.
- Works for numerical data but other variations of Kmeans work also for categorical data and mixed data types.
- Easy to implement

How the K-Means Clustering algorithm works?



The Algorithm

Algorithm 8.1 Basic K-means algorithm.

1: Select K points as initial centroids.
2: **repeat**
3: Form K clusters by assigning each point to its closest centroid.
4: Recompute the centroid of each cluster.
5: **until** Centroids do not change.

Tan et al. 2014, Introduction to Data Mining

Sample data example

K-means Clustering with 2D Data Points

Sample Data:

$$(1,2), (2,1), (2,3), (8,9), (9,8), (9,9)$$

We choose $k = 2$ (i.e., two clusters).

Step 1: Choose k

$$k = 2$$

Step 2: Initialize Centroids

$$\text{Centroids} = (2,1) \text{ and } (9,9)$$

Step 3: Assign Points to Nearest Centroid

$$\text{Points closer to } (2,1) : (1,2), (2,1), (2,3)$$
$$\text{Points closer to } (9,9) : (8,9), (9,8), (9,9)$$

Sample data example

Step 4: Recalculate Centroids

$$\text{Mean of the first cluster} = \left(\frac{1+2+2}{3}, \frac{2+1+3}{3}\right) = (1.67, 2)$$

$$\text{Mean of the second cluster} = \left(\frac{8+9+9}{3}, \frac{9+8+9}{3}\right) = (8.67, 8.67)$$

Step 5: Repeat Steps 3 & 4

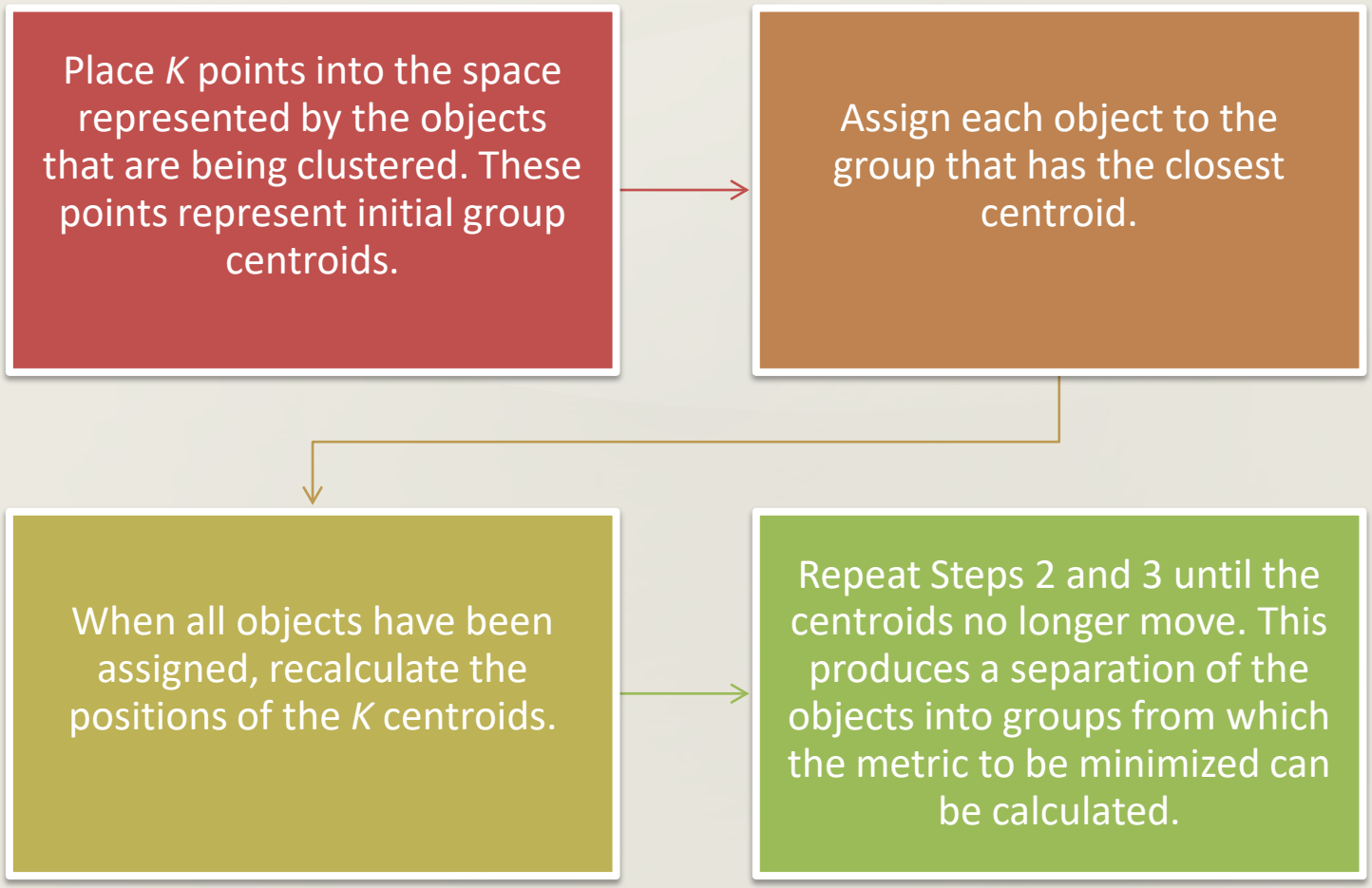
$$\text{Points closer to } (1.67, 2) : (1, 2), (2, 1), (2, 3)$$
$$\text{Points closer to } (8.67, 8.67) : (8, 9), (9, 8), (9, 9)$$

The assignment hasn't changed from the previous step, so the algorithm converges and stops.

Result:

$$\text{Cluster 1: } (1, 2), (2, 1), (2, 3)$$
$$\text{Cluster 2: } (8, 9), (9, 8), (9, 9)$$

K-means clustering algorithm



K-means clustering

Squared Error Function for K-means:

The objective of the K-means clustering algorithm is to partition a set of data points into K distinct clusters, where each data point belongs to the cluster with the nearest mean. The mean of each cluster is commonly referred to as the "centroid."

The squared error for a given cluster is the sum of the squared distances between each data point in that cluster and the centroid of the cluster. The overall squared error function (often called the "objective function" or "cost function" for K-means) is the sum of the squared errors for all K clusters.

Mathematically, the squared error function J for K-means is given by:

$$J = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$

Where:

- K is the number of clusters.
- C_i represents the set of data points in the i^{th} cluster.
- x is a data point in cluster C_i.
- μ_i is the centroid of cluster C_i.
- $||x - \mu_i||^2$ is the squared Euclidean distance between the data point x and the centroid μ_i.

K-means clustering

Minimizing Loss:

The goal of the K-means algorithm is to find cluster assignments and centroids that minimize the squared error function J. Here's how the K-means algorithm works to achieve this:

1. **Initialization**: Randomly select K data points as the initial centroids.
2. **Assignment Step**: Assign each data point to the nearest centroid. This creates K clusters. Formally, for each data point x, assign it to the cluster C_i such that:
$$C_i = \arg\min_{\mu_i} ||x - \mu_i||^2$$
3. **Update Step**: Recompute the centroid of each cluster as the mean of all data points in that cluster:
$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

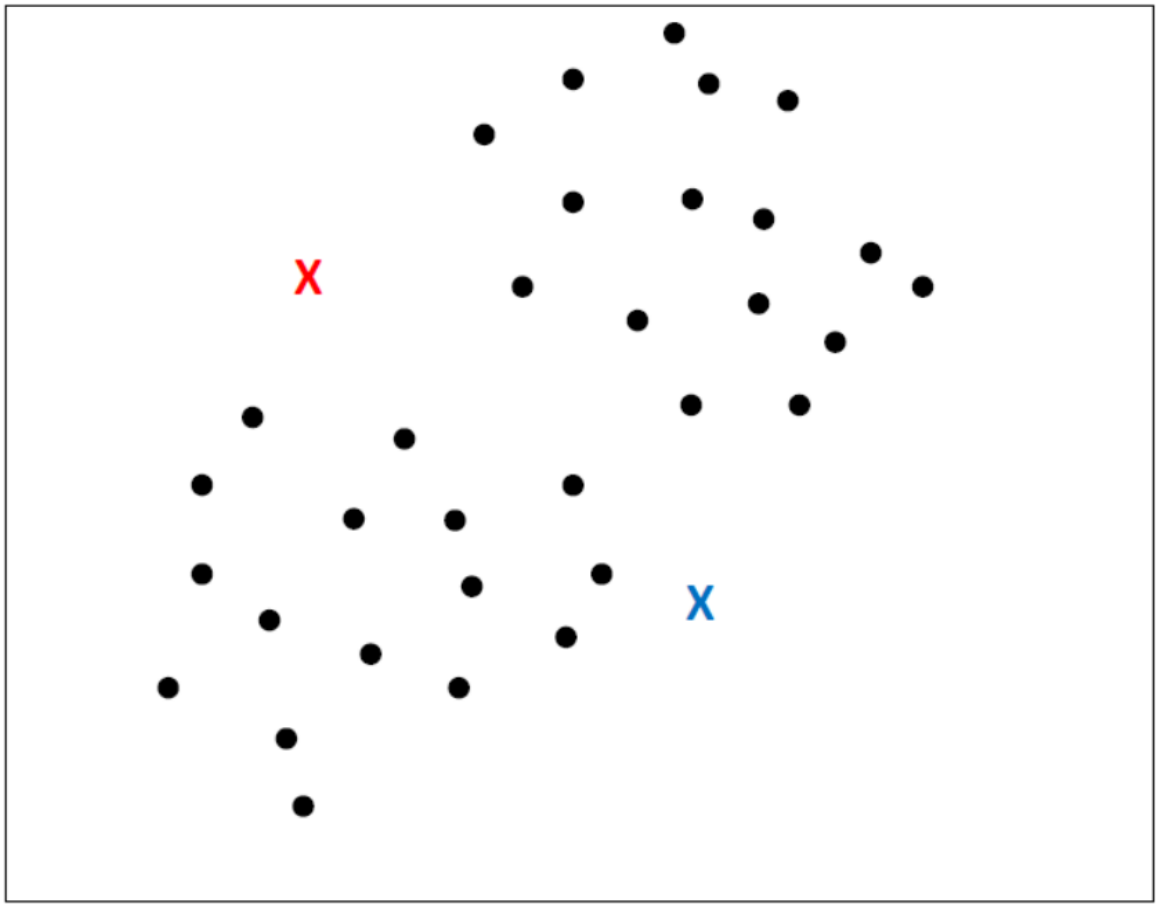
K-means clustering

4. **Convergence**: Repeat the assignment and update steps until the centroids do not change significantly or some other stopping criterion is met.

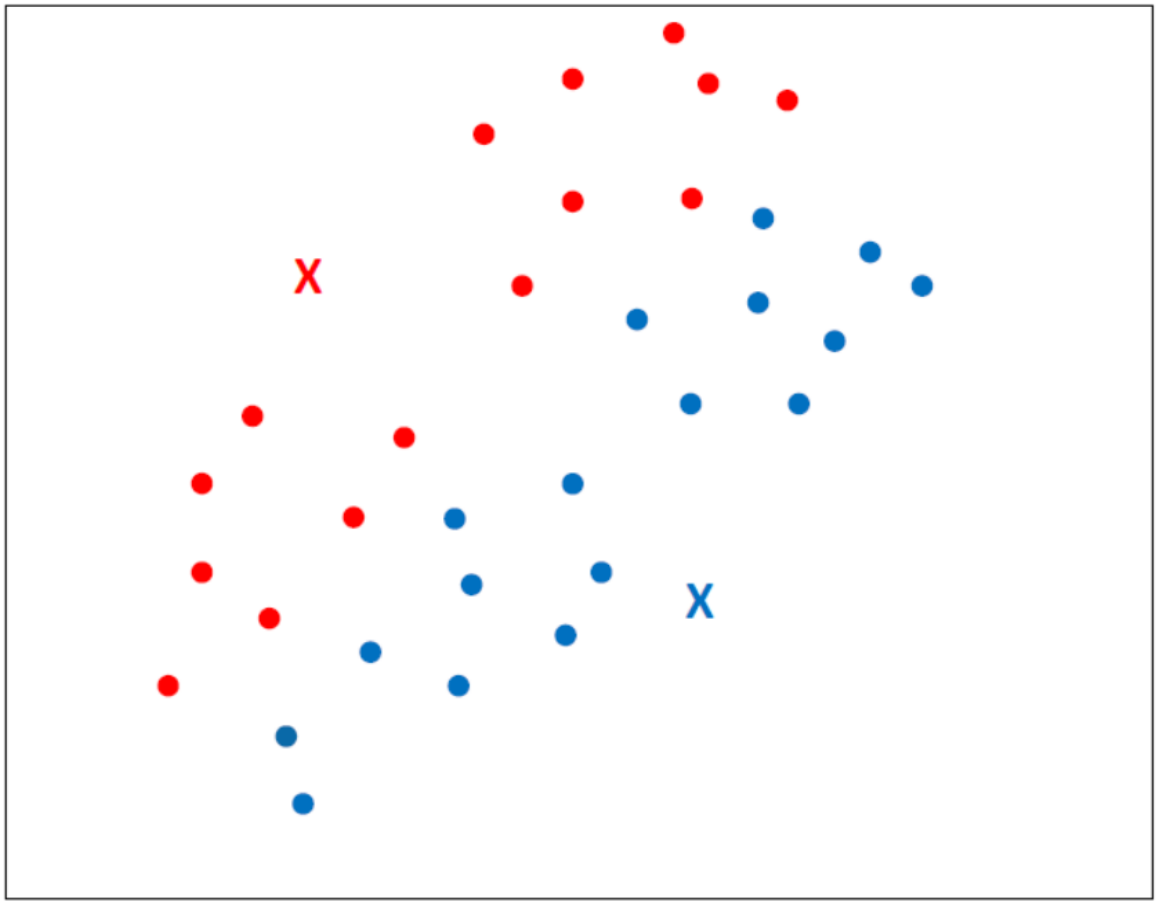
Through this iterative process, the K-means algorithm tries to minimize the squared error function by adjusting the cluster assignments and recomputing the centroids. While the algorithm is guaranteed to converge, it might converge to a local minimum, which means it might not find the best possible clustering with the minimum possible squared error. This is why the algorithm is often run multiple times with different initializations and the best result (with the lowest error) is selected.

In the context of K-means, this sum of squared errors is commonly referred to as SSE (Sum of Squared Errors) or ”inertia.” The goal of the K-means algorithm is to minimize this SSE.

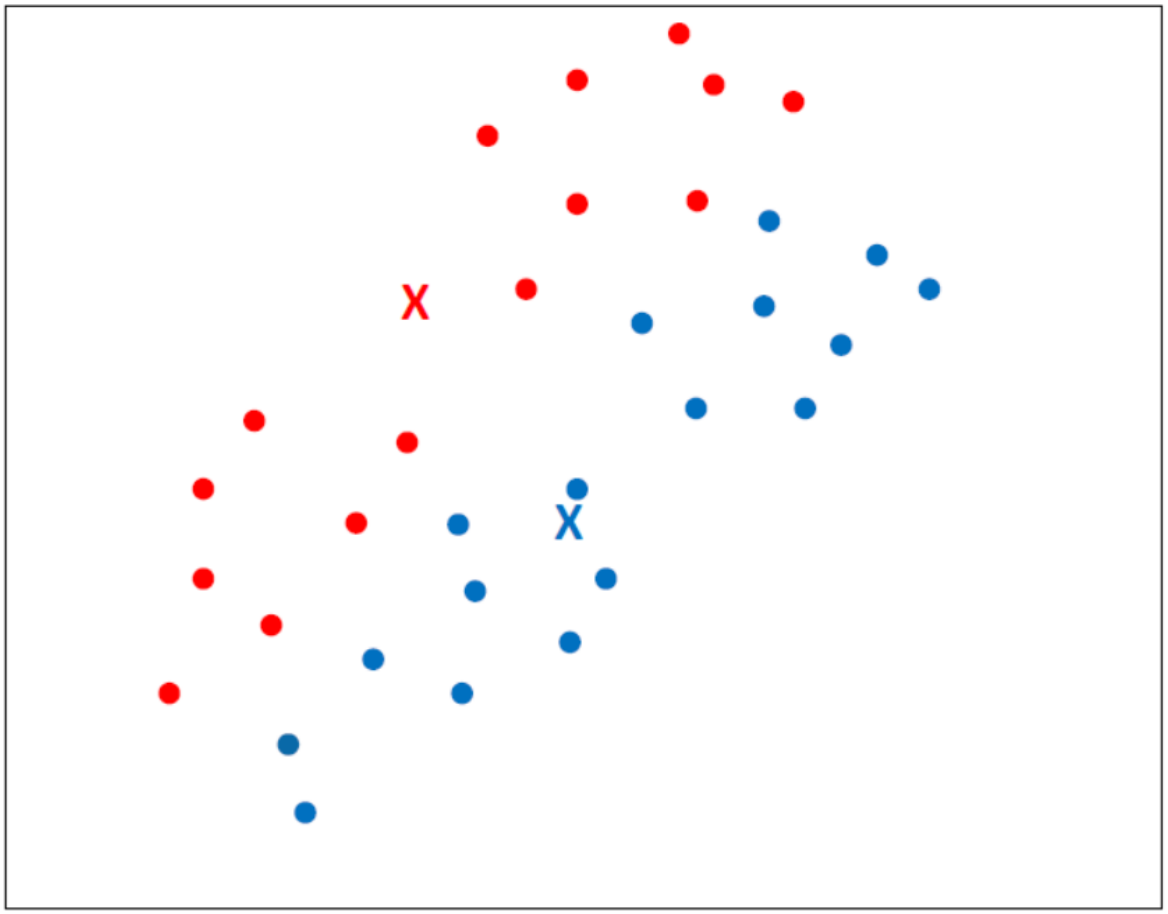
Visual Example: Initialization



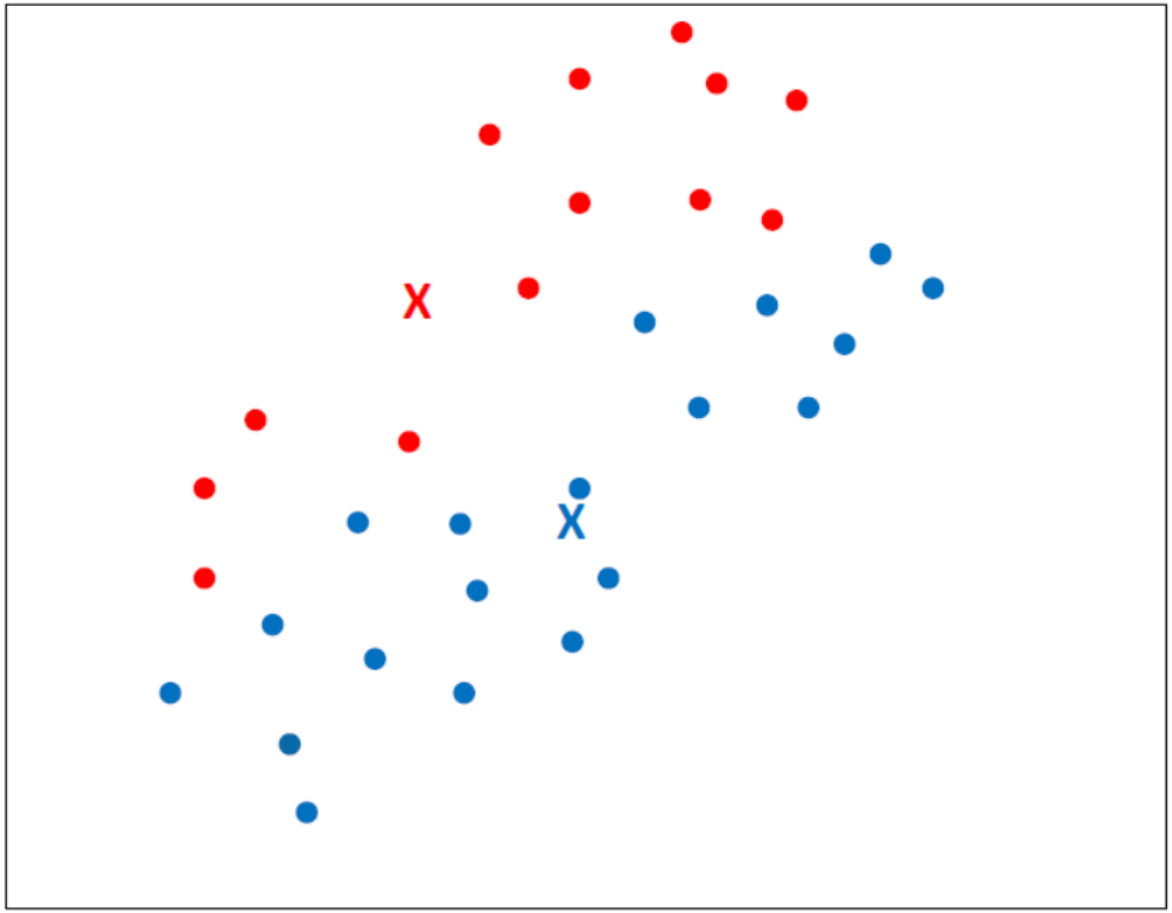
Assignment



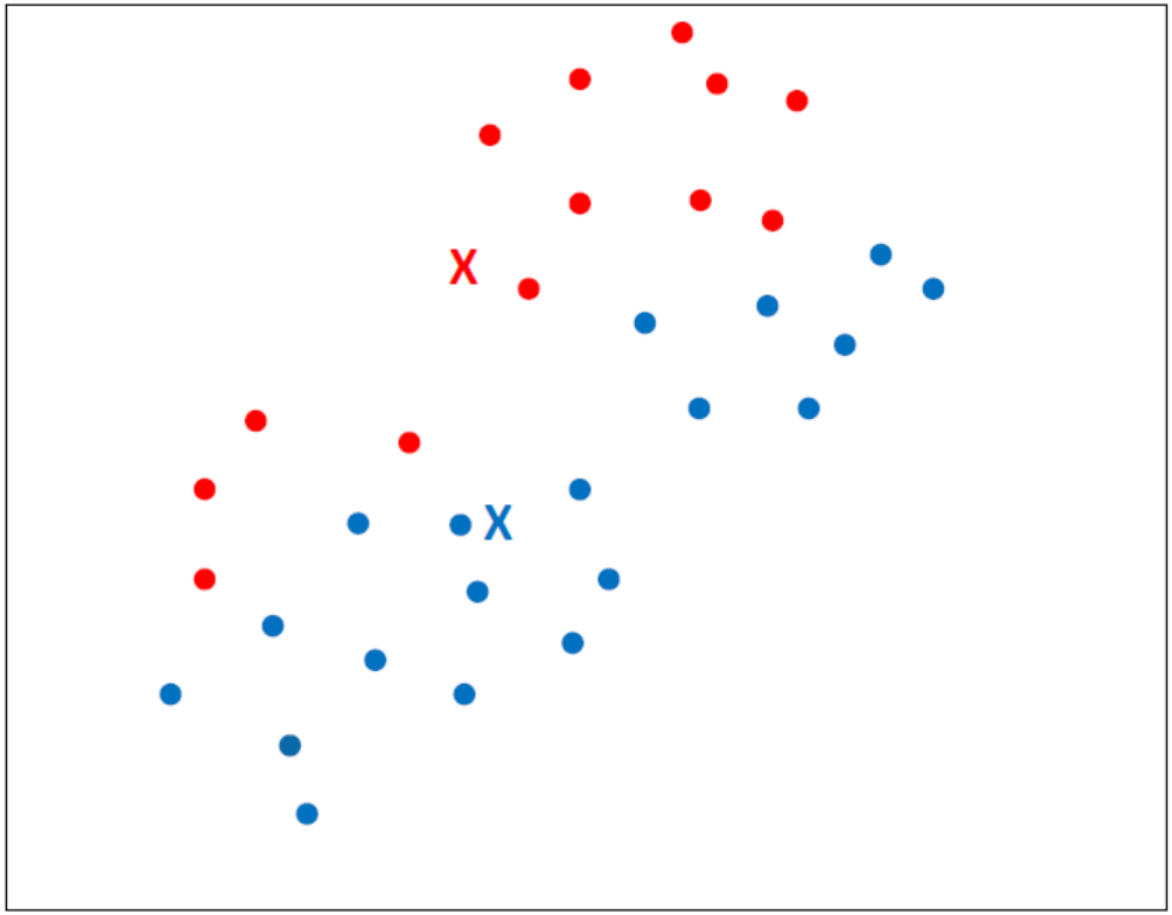
Update



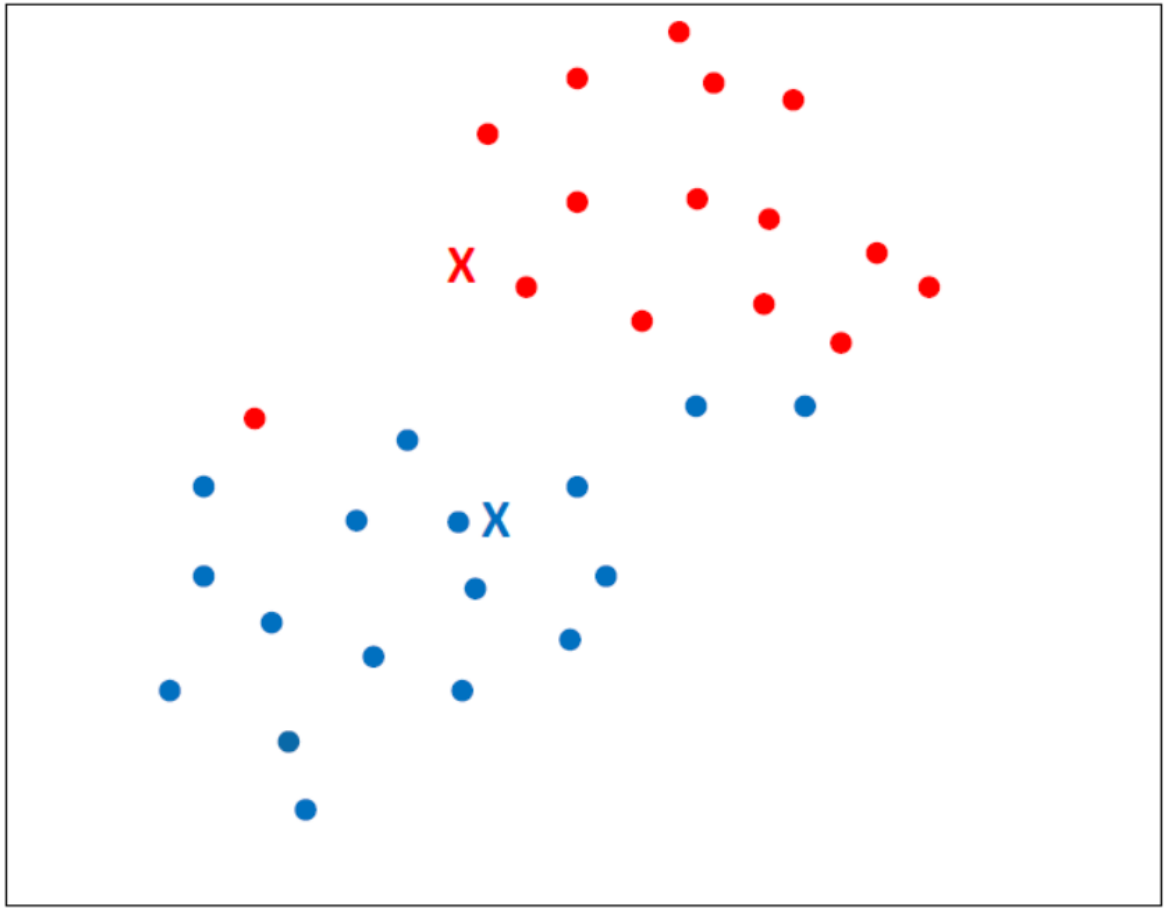
Assignment



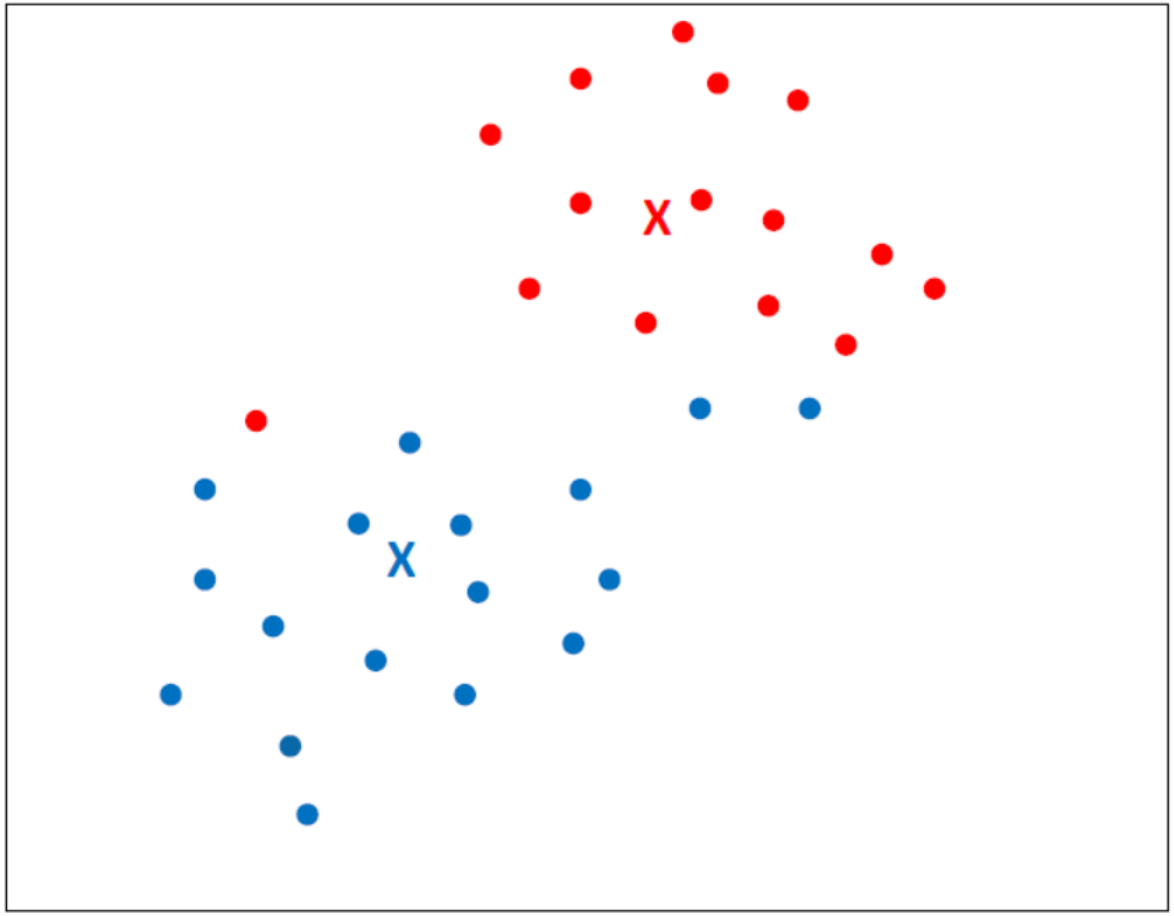
Update



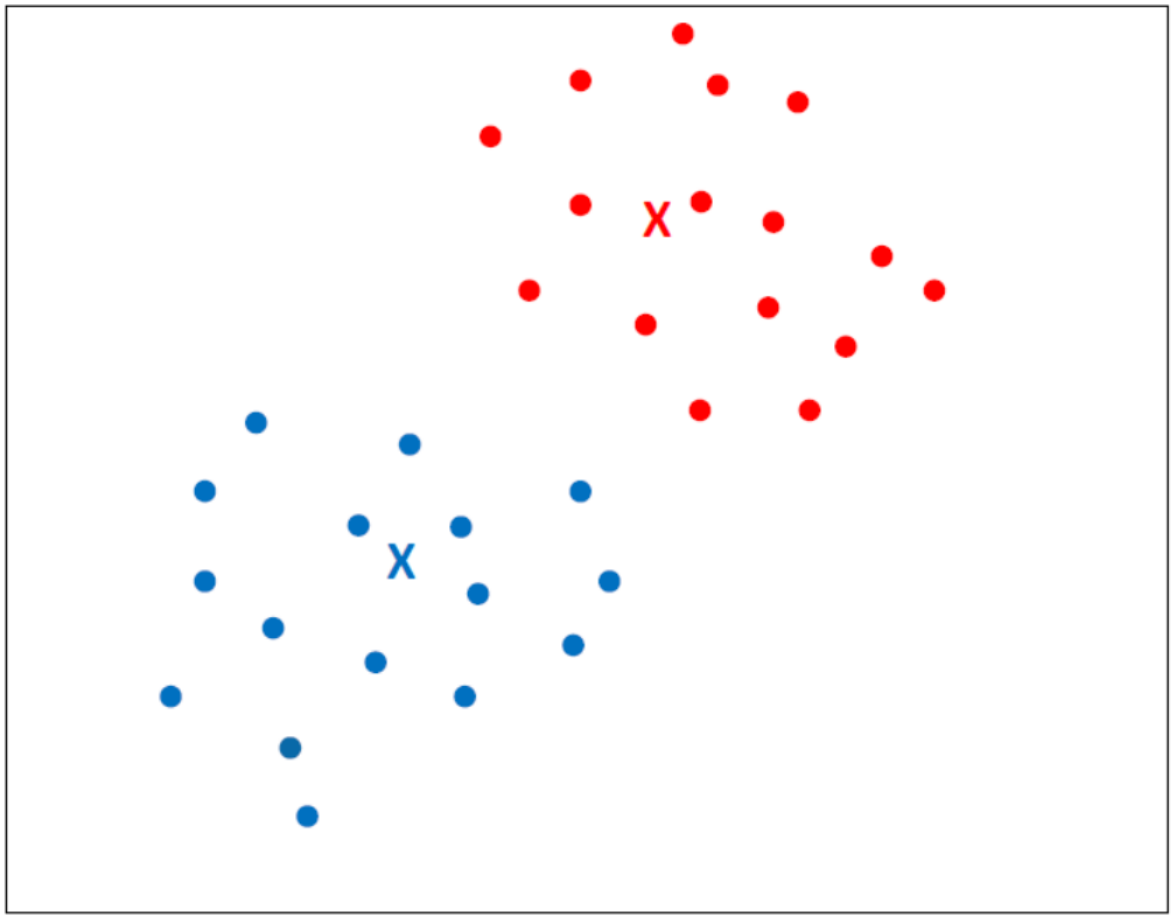
Assignment



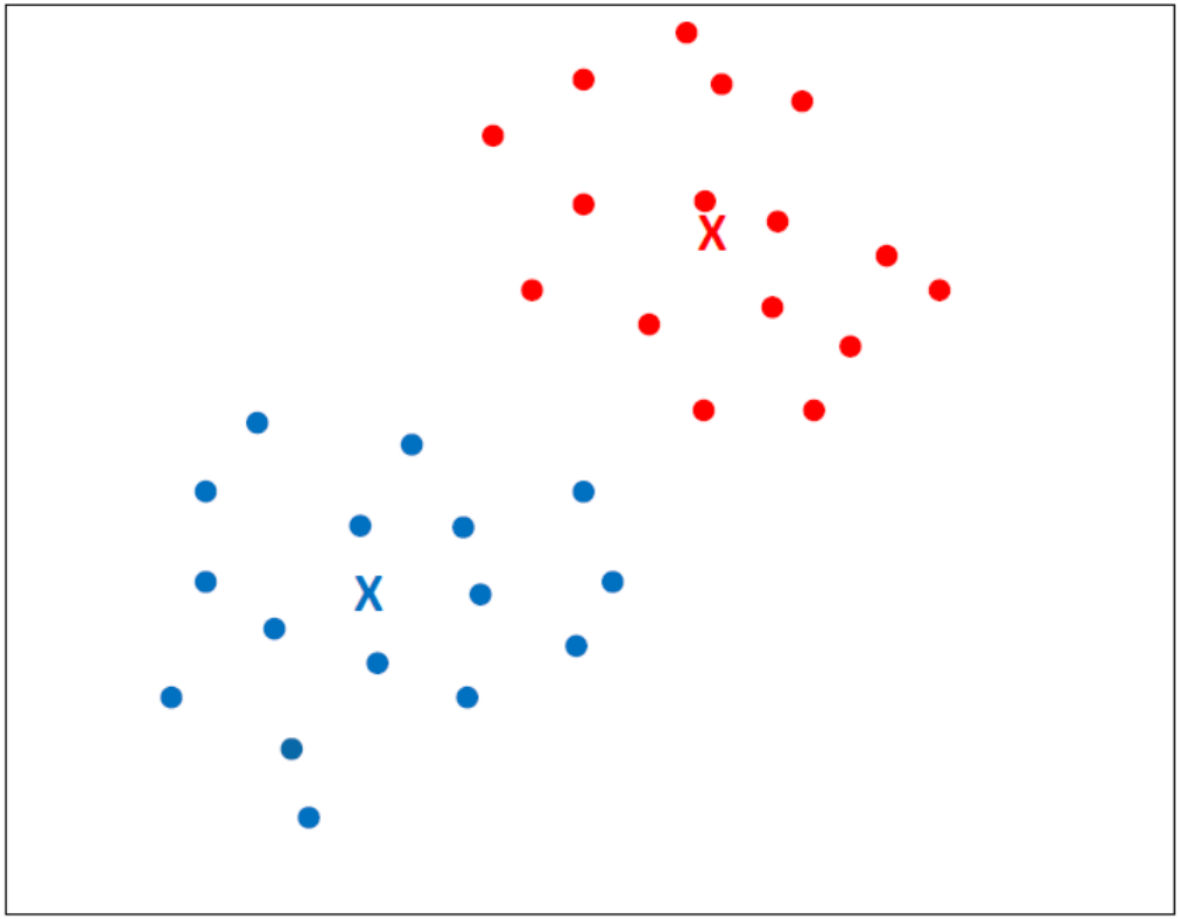
Update



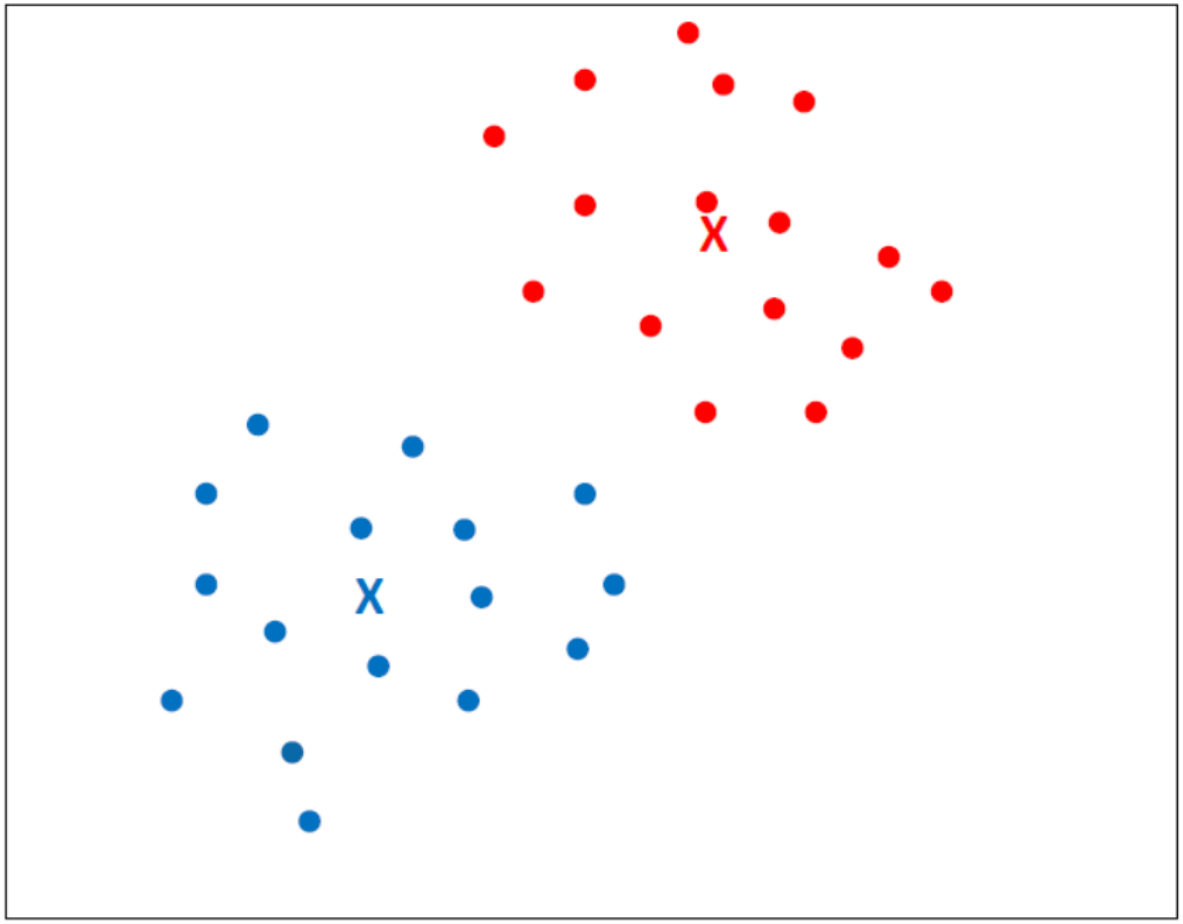
Assignment

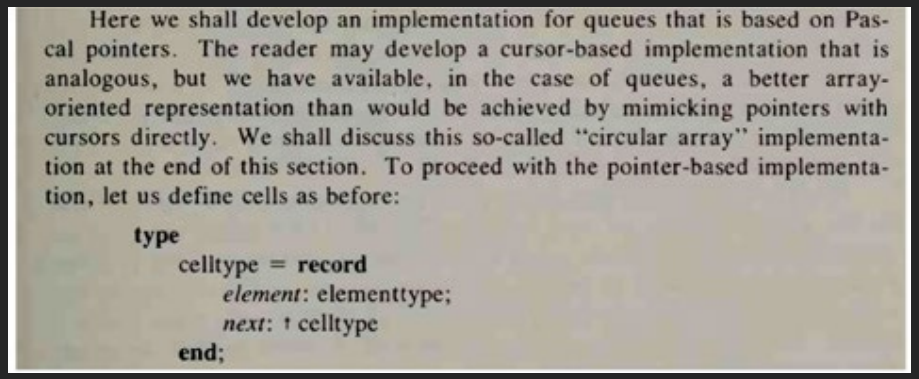


Update



Assignment (no change)





Here we shall develop an implementation for queues that is based on Pascal pointers. The reader may develop a cursor-based implementation that is analogous, but we have available, in the case of queues, a better array-oriented representation than would be achieved by mimicking pointers with cursors directly. We shall discuss this so-called "circular array" implementation at the end of this section. To proceed with the pointer-based implementation, let us define cells as before:

```
type
    celltype = record
        element: elementtype;
        next: ↑ celltype
    end;
```

Here we shall develop an implementation for queues that is based on Pascal pointers. The reader may develop a cursor-based implementation that is analogous, but we have available, in the case of queues, a better array-oriented representation than would be achieved by mimicking pointers with cursors directly. We shall discuss this so-called "circular array" implementation at the end of this section. To proceed with the pointer-based implementation, let us define cells as before:

```
type
    celltype = record
        element: elementtype;
        next: ↑ celltype
    end;
```

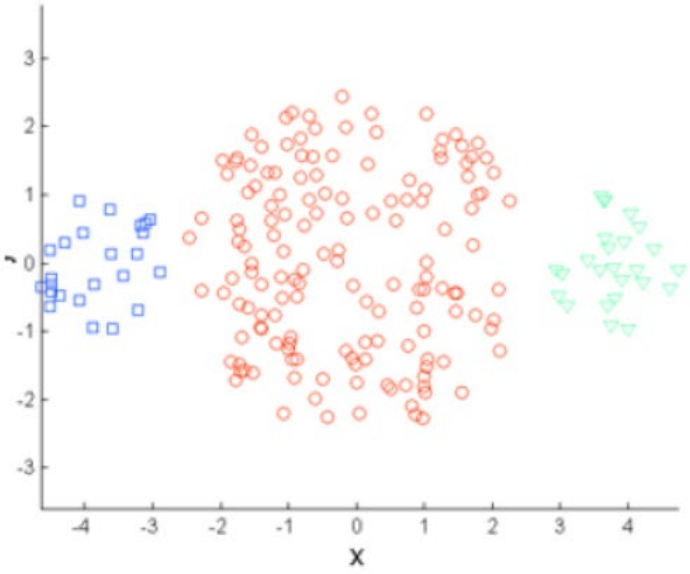
Example:

Image Binarization: which means that we need to cluster pixel values into two clusters, white as a class 1 and black as class 2.

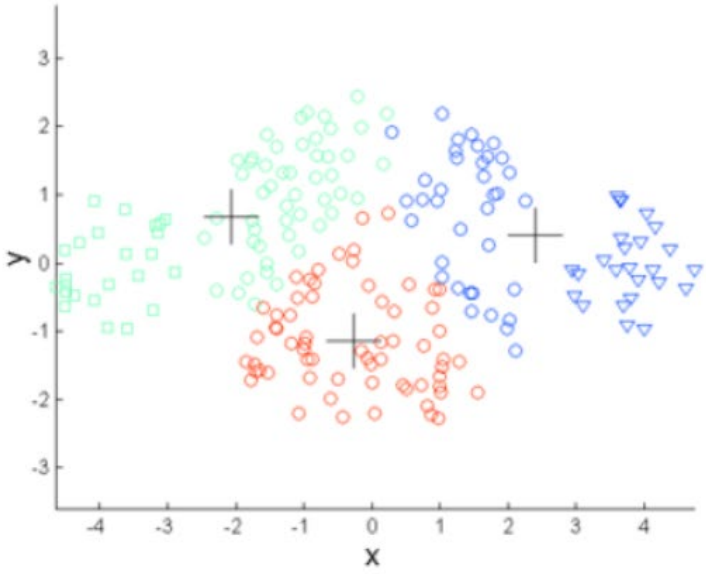
Limitations of K-means

- Works best with data sets that are equally-sized and/or with regular shapes
- K-means has problems when clusters are of different
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers.

K-means with clusters of different sizes



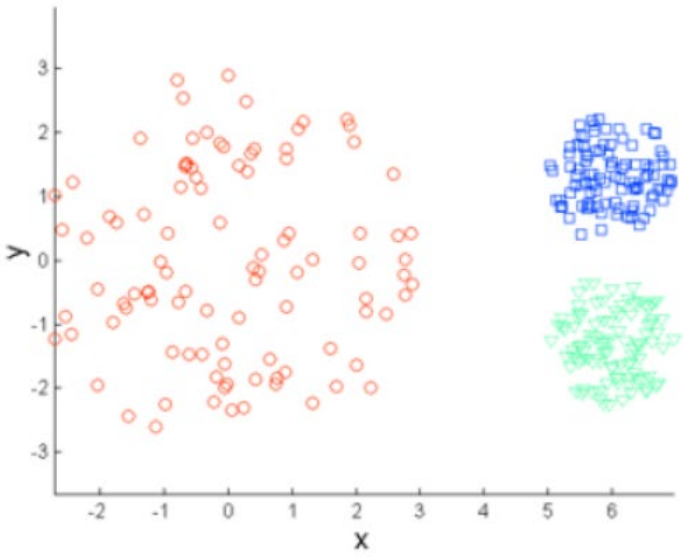
Original Points



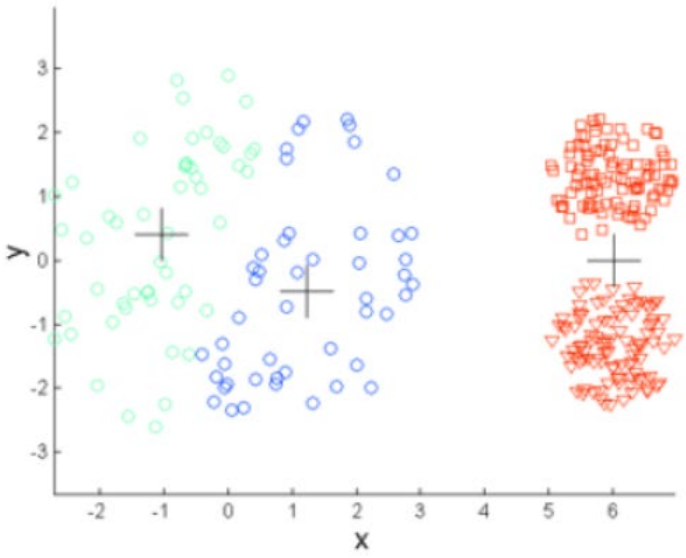
K-means (3 Clusters)

Tan et al. 2014, Introduction to Data Mining

K-means with clusters of different density



Original Points

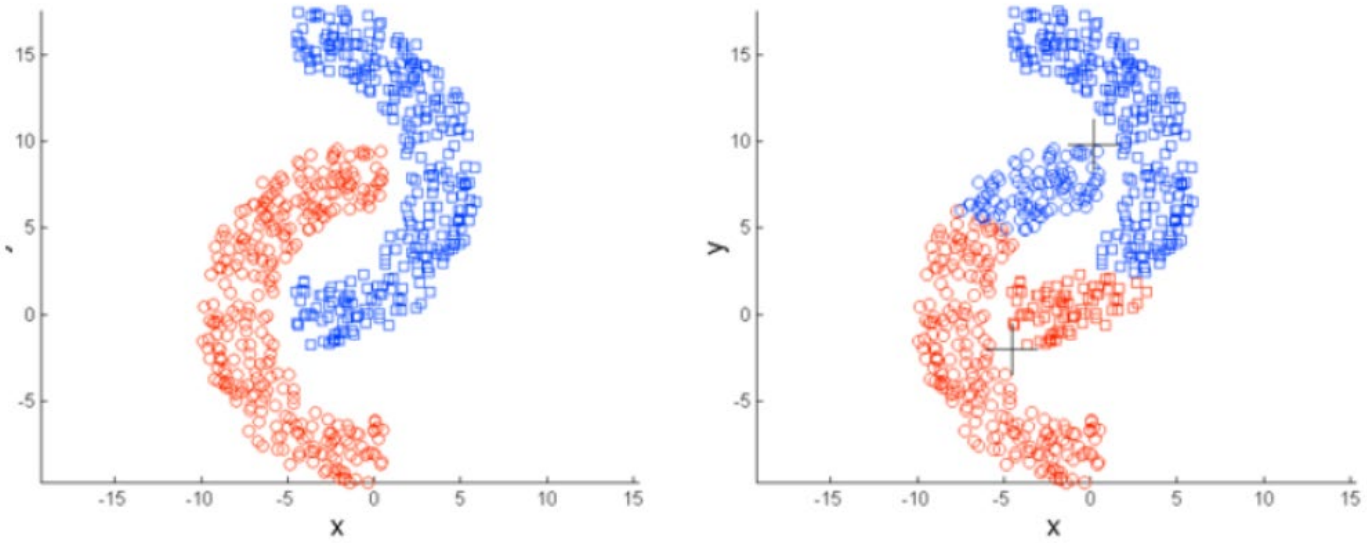


K-means (3 Clusters)

Tan et al. 2014, Introduction to Data Mining

K-means with clusters of non-globular shapes

- Shapes or structures that aren't round or spherical in nature



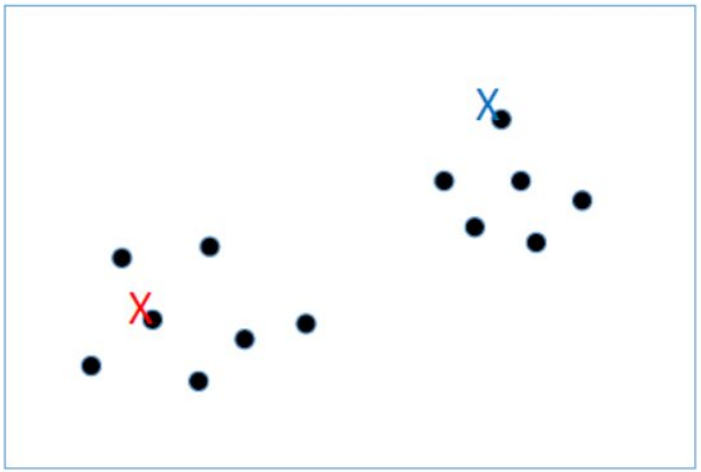
Original Points **K-means (2 Clusters)**

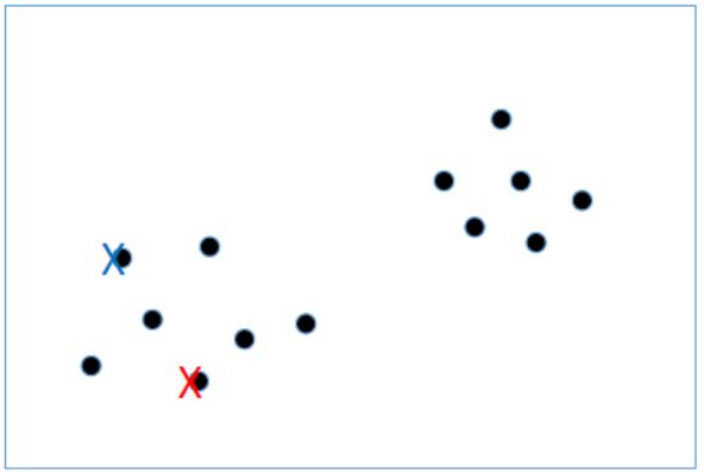
Tan et al. 2014, Introduction to Data Mining

Initializing K-means

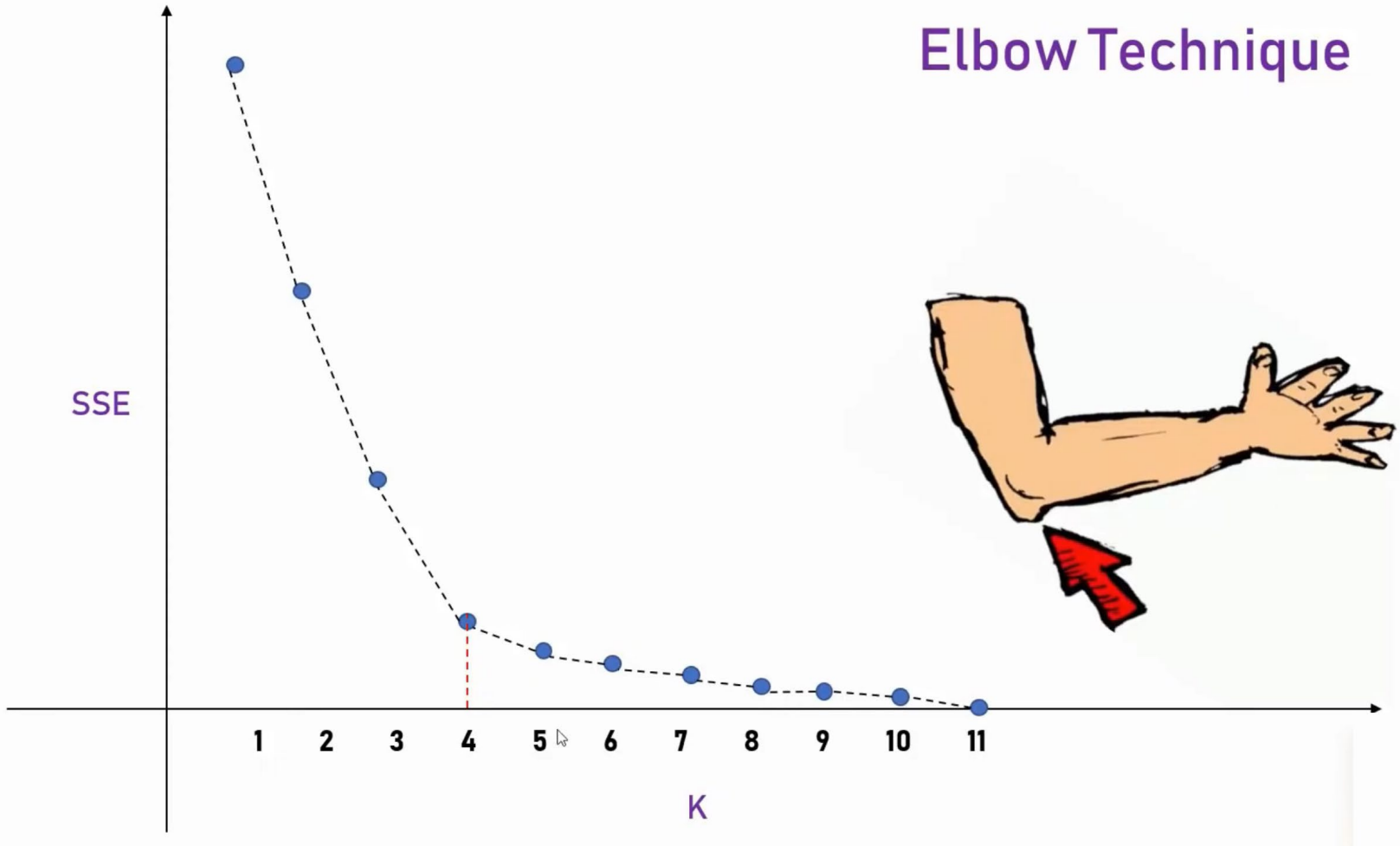
Each initialization can lead to a different solution (clustering)

- K should be less than the number of objects
- Randomly pick K training examples
  - Set these K examples as the cluster centroids

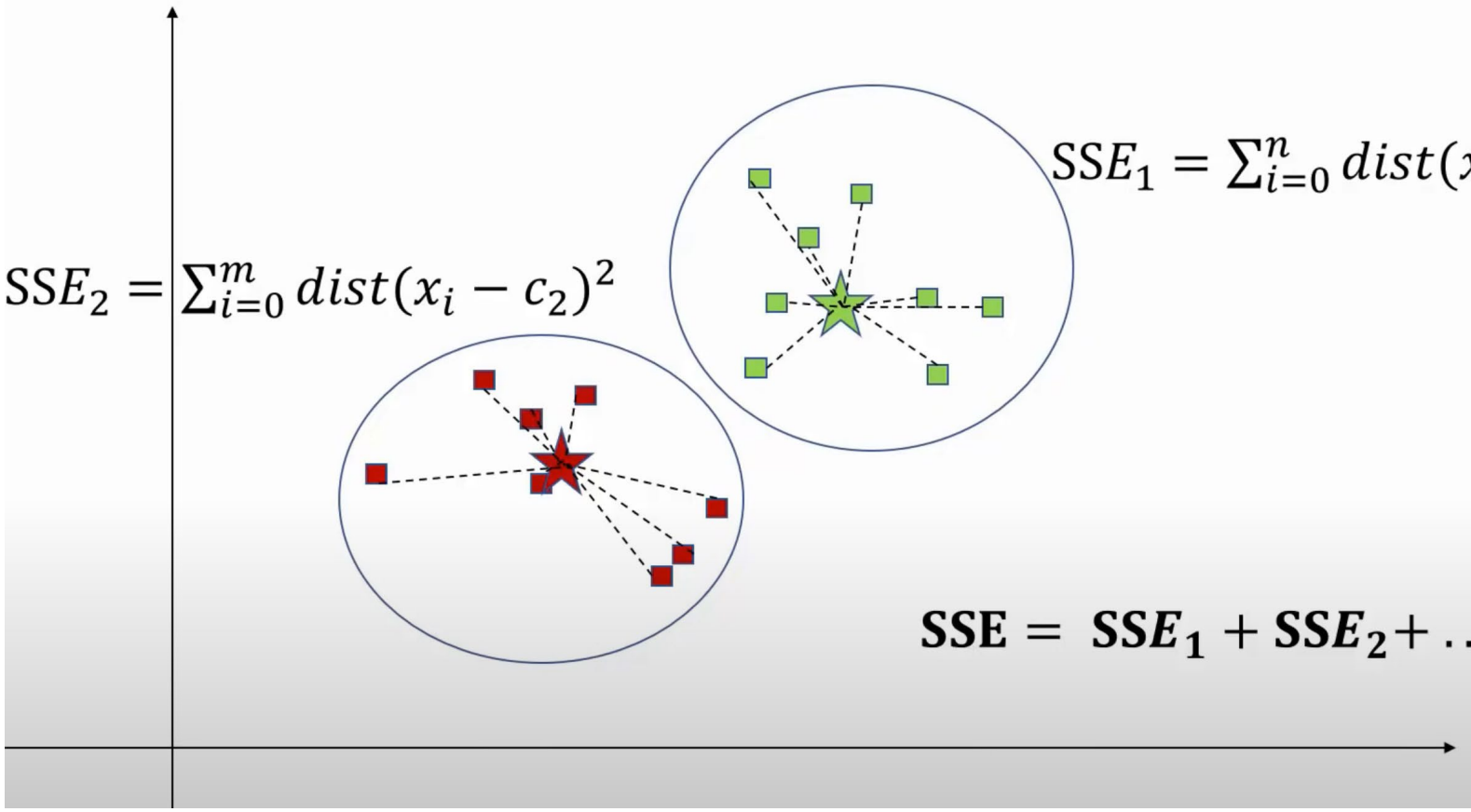




Elbow Technique



SSE = Sum of Squared Errors



$$SSE_1 = \sum_{i=0}^{n} dist(x_i - c_1)^2$$

$$SSE_2 = \sum_{i=0}^{m} dist(x_i - c_2)^2$$

$$\mathbf{SSE} = \mathbf{SSE_1} + \mathbf{SSE_2} + .. + \mathbf{SSE_k}$$