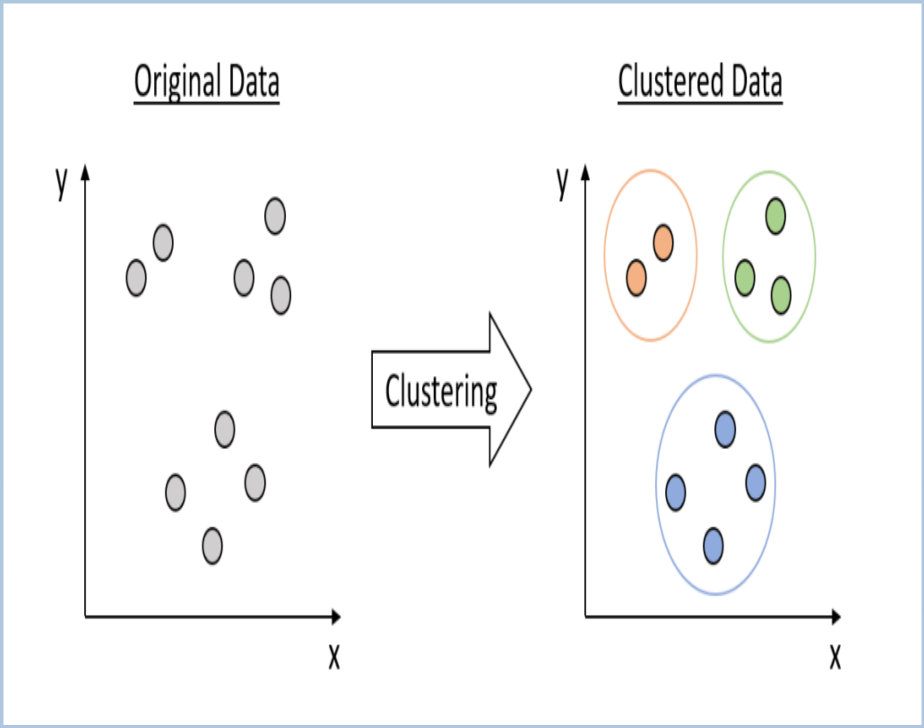
الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

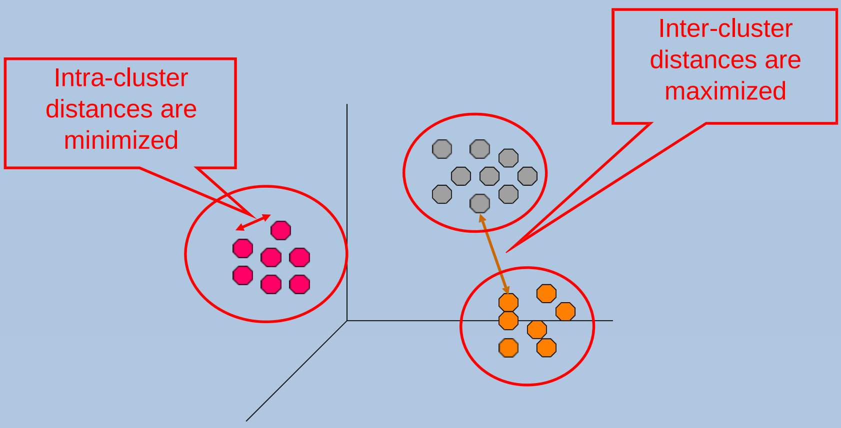
Clustering



Rana Husni

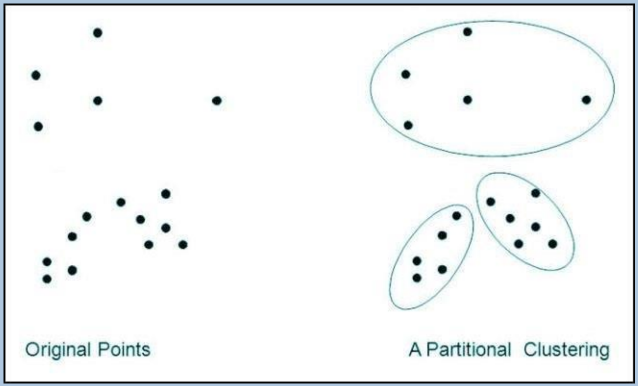
1. What is Cluster Analysis?

- Finding groups of objects such that
  - the objects in a group will be similar to one another
  - and different from the objects in other groups.
- Goal: Get a better understanding of the data

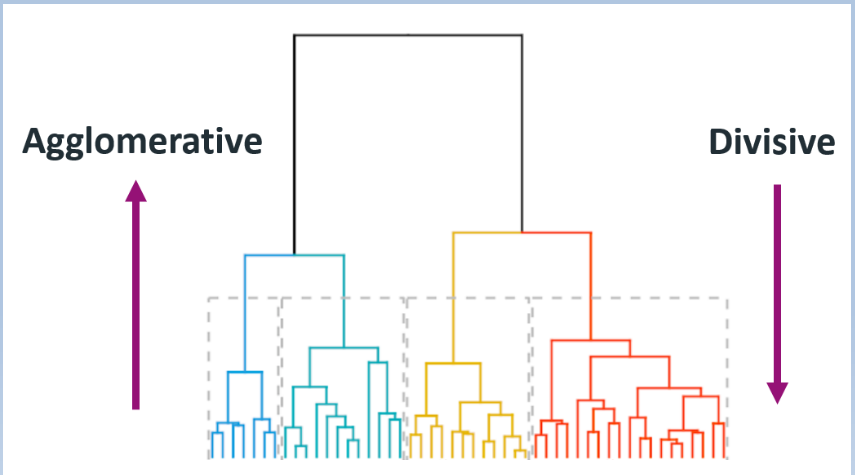


Types of Clusterings

– Partitional Clustering
- A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset



–Hierarchical Clustering
- A set of nested clusters organized as a hierarchical tree



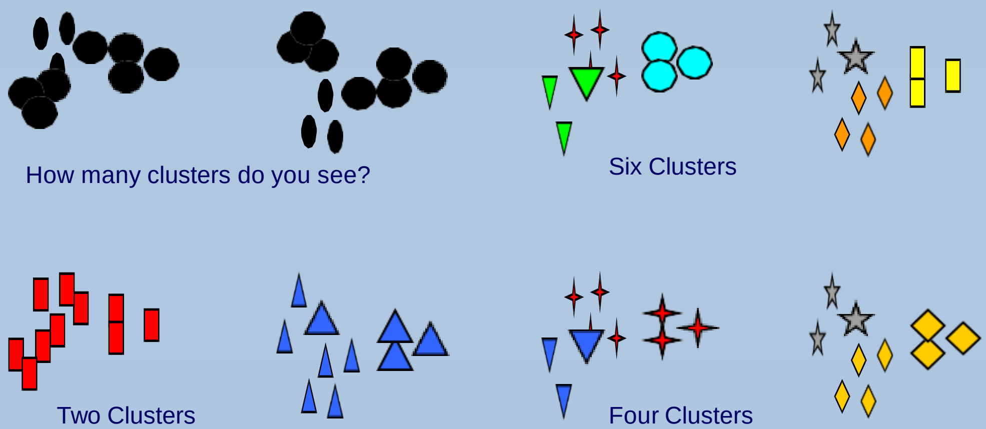
Aspects of Cluster Analysis

– A clustering algorithm
  - Partitional algorithms
  - Density-based algorithms
  - Hierarchical algorithms

A proximity (similarity, or dissimilarity) measure
- Euclidean distance
- Cosine similarity
- Data type-specific similarity measures
- Domain-specific similarity measures

– Clustering quality
  - Intra-clusters distance $\Rightarrow$ minimized
  - Inter-clusters distance $\Rightarrow$ maximized
  - The clustering should be useful with regard to the goal of the analysis

The Notion of a Cluster is Ambiguous



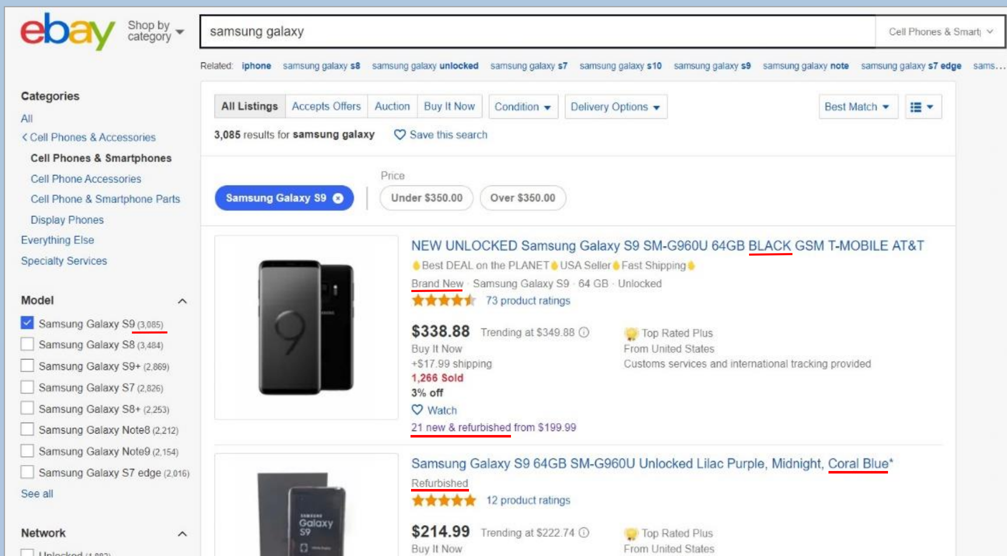
The usefulness of a clustering depends on the goal of the analysis

Example Application 1: Market Segmentation

– Goal: Identify groups of similar customers
– Level of granularity depends on the task at hand
– Relevant customer attributes depend on the task at hand

Example Application 2: E-Commerce

– Identify offers of the same product on electronic markets



Example Application 3: Image Recognition

– Identify parts of an image that belong to the same object

Cluster Analysis as Unsupervised Learning

– Supervised learning: Discover patterns in the data that relate data attributes with a target (class) attribute
  - these patterns are then utilized to predict the values of the target attribute in unseen data instances
  - the set of classes <u>is known</u> before
  - training data is often provided by human annotators

– Unsupervised learning: The data has no target attribute
  - we want to <u>explore the data</u> to find some intrinsic patterns in it
  - the set of classes/clusters <u>is not known</u> before
  - no training data is used

– Cluster Analysis is an unsupervised learning task

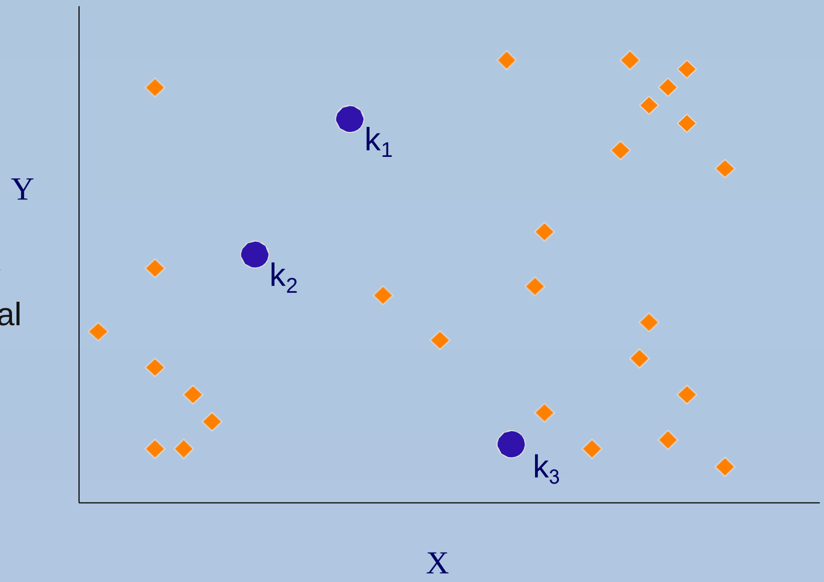
K-Means Clustering

- Partitional clustering algorithm
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters K must be specified manually

- The K-Means algorithm is very simple:

```
Select K points as the initial centroids.
repeat
    Form K clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

K-Means Example, Step 1

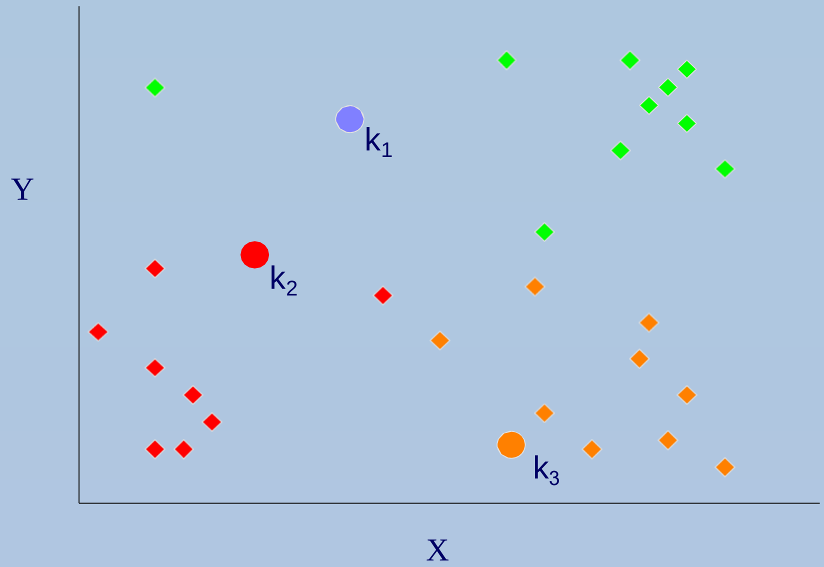


Randomly pick 3 initial centroids

SEEIT

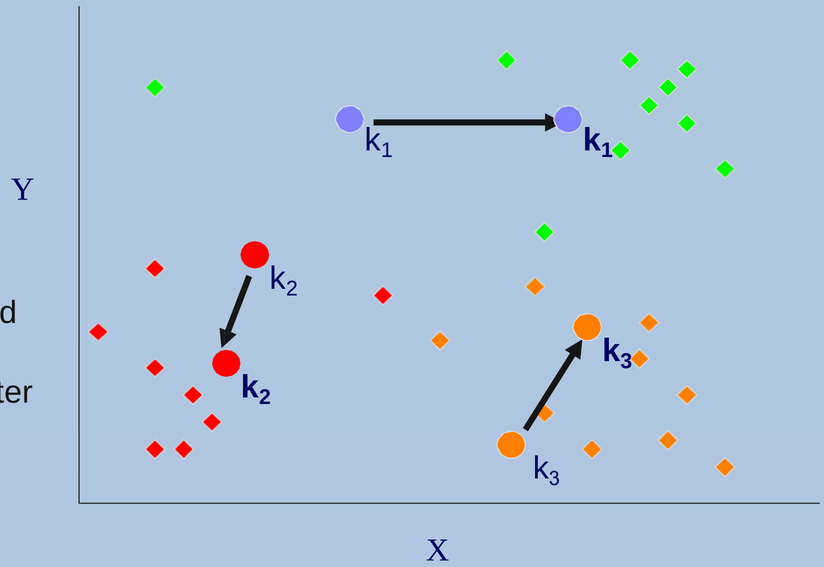
K-Means Example, Step 2

Assign each point to the closest centroid



K-Means Example, Step 3

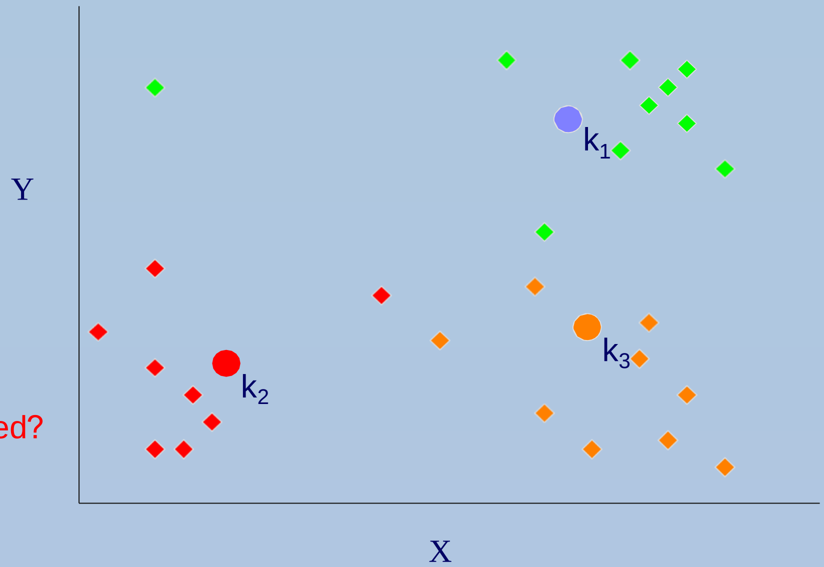
Move
each centroid
to the mean
of each cluster



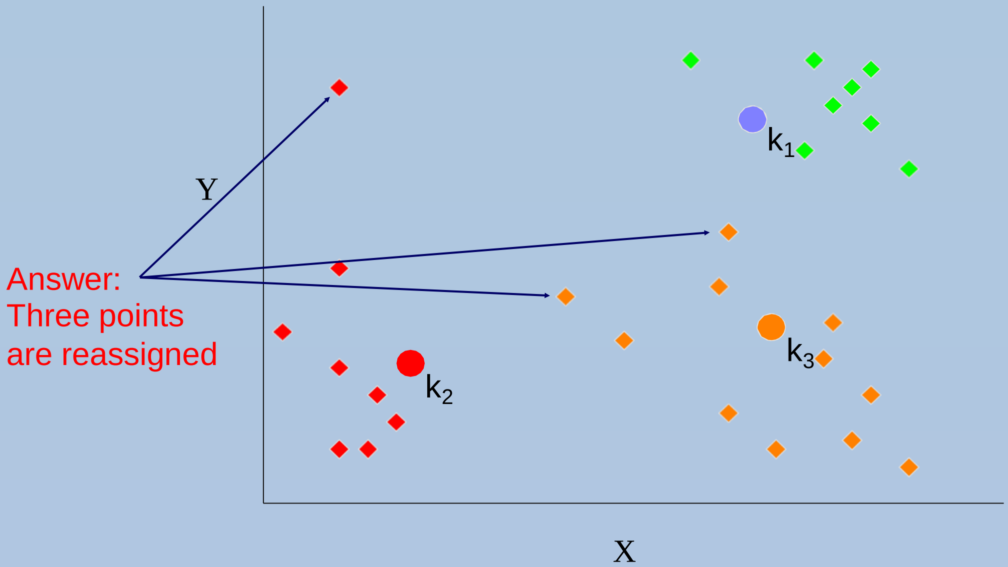
K-Means Example, Step 4

Reassign points if they are now closer to a different centroid

Question: Which points are reassigned?



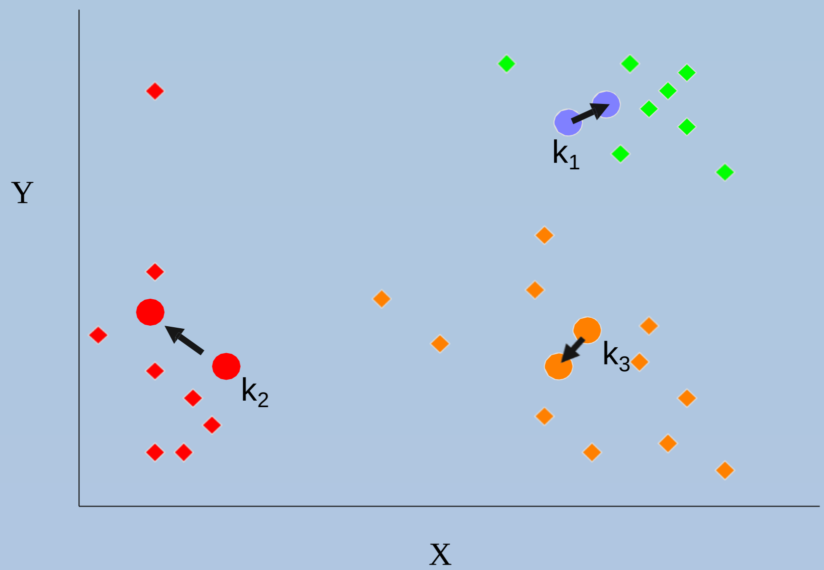
K-Means Example, Step 4



GJU
الجامعة الألمانية الأردنية
German Jordanian University

SEEIT

K-Means Example, Step 5



1. Re-compute cluster means
2. Move centroids to new cluster means

Rana Husni

Convergence Criteria

Default convergence criterion

–no (or minimum) change of centroids

Alternative convergence criteria

1.no (or minimum) re-assignments of data points to different clusters

1.stop after x iterations

2.minimum decrease in the sum of squared error (SSE)

Evaluating K-Means Clusterings

– Widely used cohesion measure: Sum of Squared Error (SSE)

- For each point, the error is the distance to the nearest centroid
- To get SSE, we square these errors and sum them

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

•C_j is the j-th cluster

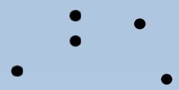
•m_j is the centroid of cluster C_j naem eht)
ni stniop atad eht lla fo rotcevC_j(

•*dist*m ,x)$_j$ atad neewteb ecnatsid eht si (
m diortnec dna x tniop$_j$

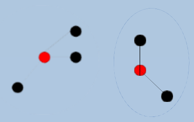
– Given several clusterings (= groupings), we should prefer the one with the smallest SSE

Illustration: Sum of Squared Error

- Cluster analysis problem



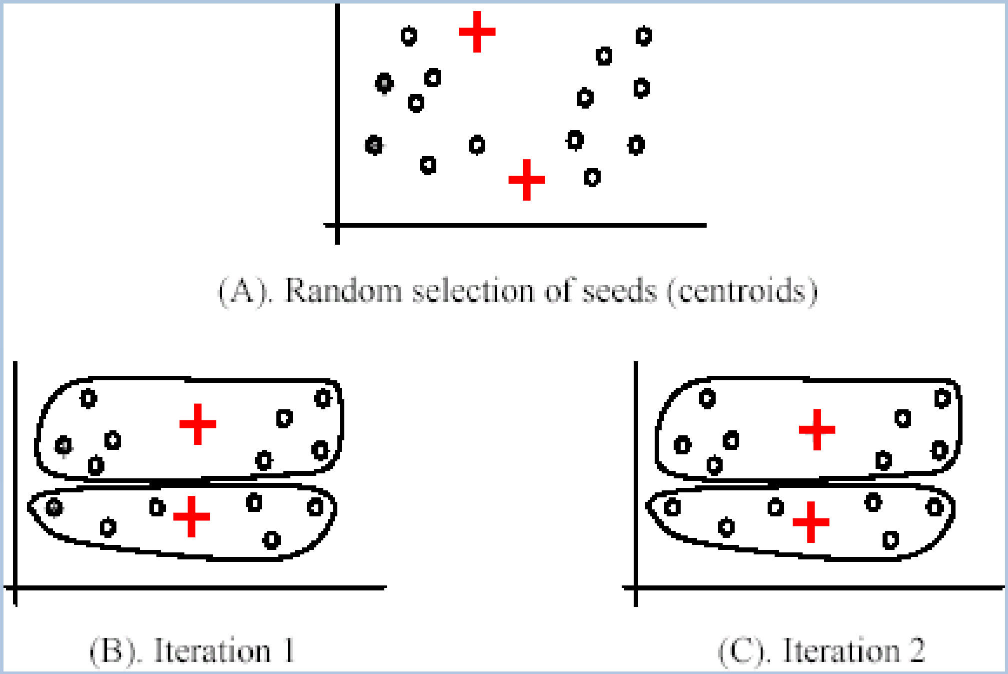
- Good clustering
  - small distances to centroids



- Not so good clustering
  - larger distances to centroids

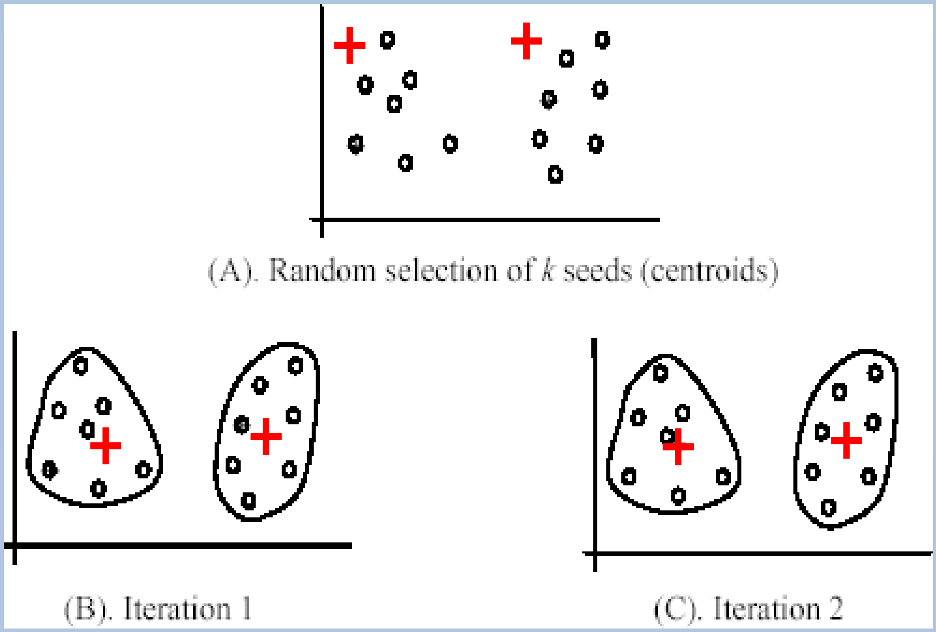
Weaknesses of K-Means: Initial Seeds

Clustering results may vary significantly depending on initial choice of seeds (number and position of seeds)

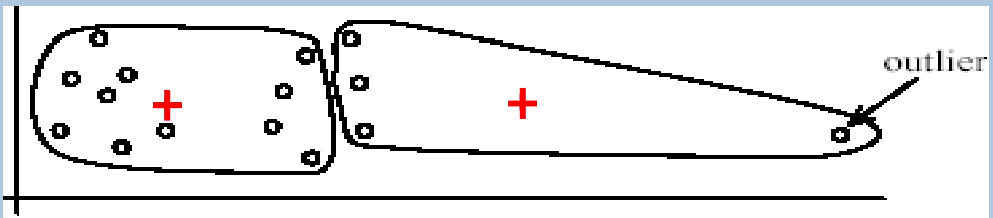


Weaknesses of K-Means: Initial Seeds

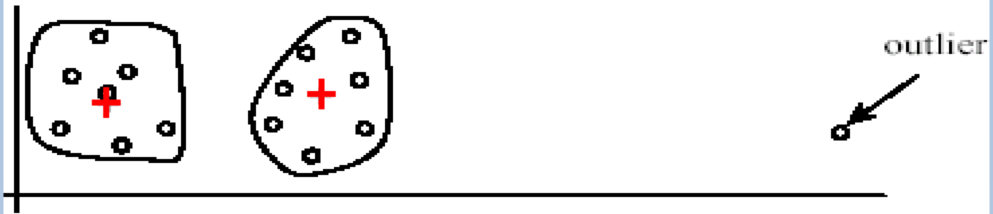
If we use different seeds, we get good results



Weaknesses of K-Means: Problems with Outliers



(A): Undesirable clusters



(B): Better clusters

Weaknesses of K-Means: Problems with Outliers

Approaches to deal with outliers:

1.K-Medoids

- K-Medoids is a K-Means variation that uses the median of each cluster instead of the mean
- Medoids are the most central existing data points in each cluster
- K-Medoids is more robust against outliers as the median is less affected by extreme values:
  - Mean and Median of 9 ,7 ,5 ,3 ,1is 5
  - Mean of 1009 ,7 ,5 ,3 ,1is 205
  - Median of 1009 ,7 ,5 ,3 ,1is 5

2.DBSCAN

- Density-based clustering method that removes outliers

Suppose that the data mining task is to cluster points into three clusters, where the points are A1(2, 10), A2(2, 5), A3(8, 4), B1(5, 8), B2(7, 5), B3(6, 4), C1(1, 2), C2(4, 9). The distance function is Euclidean distance.

Step 1: Define the Data you have eight points with (x, y) coordinates:

- A1 = (2, 10)
- A2 = (2, 5)
- A3 = (8, 4)
- B1 = (5, 8)
- B2 = (7, 5)
- B3 = (6, 4)
- C1 = (1, 2)
- C2 = (4, 9)

Step 2: Choose the Number of Clusters (k) We choose k=3, as specified.

Step 3: Initialize the Centroids You can initialize the centroids randomly or choose specific points as initial centroids. Here, let's pick three initial centroids as follows:

- Centroid 1 = (2, 10) (Cluster 1)
- Centroid 2 = (7, 5) (Cluster 2)
- Centroid 3 = (1, 2) (Cluster 3)

Euclidean Distance Formula

Given two points (x_1, y_1) and (x_2, y_2), the Euclidean distance is calculated as:
$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Calculate Distances for Each Point to Each Centroid

Distances from Centroid 1 = (2, 10)

1. A1 = 0 (it's the centroid)
2. A2 = $\sqrt{(2-2)^2 + (5-10)^2} = 5$
3. A3 = $\sqrt{(8-2)^2 + (4-10)^2} \approx 8.49$
4. B1 = $\sqrt{(5-2)^2 + (8-10)^2} \approx 3.61$
5. B2 = $\sqrt{(7-2)^2 + (5-10)^2} \approx 7.07$
6. B3 = $\sqrt{(6-2)^2 + (4-10)^2} \approx 6.40$
7. C1 = $\sqrt{(1-2)^2 + (2-10)^2} \approx 8.06$
8. C2 = $\sqrt{(4-2)^2 + (9-10)^2} \approx 2.24$

Resulting points for Cluster 1: A1, B1, C2.

Distances from Centroid 2 = (7, 5)

1. A1 = $\sqrt{(2-7)^2+(10-5)^2} \approx 7.07$
2. A2 = $\sqrt{(2-7)^2+(5-5)^2} \approx 5$
3. A3 = $\sqrt{(8-7)^2+(4-5)^2} \approx 1.41$
4. B1 = $\sqrt{(5-7)^2+(8-5)^2} \approx 3.61$
5. B2 = 0 (it's the centroid)
6. B3 = $\sqrt{(6-7)^2+(4-5)^2} \approx 1.41$
7. C1 = $\sqrt{(1-7)^2+(2-5)^2} \approx 6.71$
8. C2 = $\sqrt{(4-7)^2+(9-5)^2} \approx 5$

Resulting points for Cluster 2: A3, B2, B3.

Distances from Centroid 3 = (1, 2)

1. A1 = $\sqrt{(2-1)^2+(10-2)^2} \approx 8.06$
2. A2 = $\sqrt{(2-1)^2+(5-2)^2} \approx 3.16$
3. A3 = $\sqrt{(8-1)^2+(4-2)^2} \approx 7.28$
4. B1 = $\sqrt{(5-1)^2+(8-2)^2} \approx 7.21$
5. B2 = $\sqrt{(7-1)^2+(5-2)^2} \approx 6.71$
6. B3 = $\sqrt{(6-1)^2+(4-2)^2} \approx 5.39$
7. C1 = 0 (it's the centroid)
8. C2 = $\sqrt{(4-1)^2+(9-2)^2} \approx 7.28$

Resulting points for Cluster 3: A2, C1.

Step 5: Update the Centroids

To update the centroids, calculate the mean of the points in each cluster. This involves taking the average of the x- and y-coordinates for each cluster's points.

New Centroid for Cluster 1

- **Mean of x-coordinates:** $(2+5+4)/3 \approx 3.67$
- **Mean of y-coordinates:** $(10+8+9)/3 \approx 9$
- **New Centroid:** $(3.67, 9)$

New Centroid for Cluster 2

- **Mean of x-coordinates:** $(8+7+6)/3 \approx 7$
- **Mean of y-coordinates:** $(4+5+4)/3 \approx 4.33$
- **New Centroid:** $(7, 4.33)$

New Centroid for Cluster 3

- **Mean of x-coordinates:** $(2+1)/2 \approx 1.5$
- **Mean of y-coordinates:** $(5+2)/2 \approx 3.5$
- **New Centroid:** $(1.5, 3.5)$

SEEIT

Step 6: Reassign Points to the New Centroids

In this step, we reassign the points to the cluster of the nearest updated centroid. The process is similar to Step 4, where we calculate the Euclidean distances between each point and each centroid, then assign points to the nearest centroid.

Recalculating Distances for New Centroids

New Centroid 1 = (3.67, 9)

1. A1 = $\sqrt{(2-3.67)^2+(10-9)^2} \approx 1.82$
2. A2 = $\sqrt{(2-3.67)^2+(5-9)^2} \approx 4.38$
3. A3 = $\sqrt{(8-3.67)^2+(4-9)^2} \approx 6.15$
4. B1 = $\sqrt{(5-3.67)^2+(8-9)^2} \approx 1.36$
5. B2 = $\sqrt{(7-3.67)^2+(5-9)^2} \approx 4.38$
6. B3 = $\sqrt{(6-3.67)^2+(4-9)^2} \approx 5.34$
7. C1 = $\sqrt{(1-3.67)^2+(2-9)^2} \approx 7.41$
8. C2 = $\sqrt{(4-3.67)^2+(9-9)^2} \approx 0.33$

New Centroid 2 = (7, 4.33)

1. A1 = $\sqrt{(2-7)^2+(10-4.33)^2} \approx 7.38$
2. A2 = $\sqrt{(2-7)^2+(5-4.33)^2} \approx 5.11$
3. A3 = $\sqrt{(8-7)^2+(4-4.33)^2} \approx 1.05$
4. B1 = $\sqrt{(5-7)^2+(8-4.33)^2} \approx 4.07$
5. B2 = $\sqrt{(7-7)^2+(5-4.33)^2} \approx 0.67$
6. B3 = $\sqrt{(6-7)^2+(4-4.33)^2} \approx 1.05$
7. C1 = $\sqrt{(1-7)^2+(2-4.33)^2} \approx 6.35$
8. C2 = $\sqrt{(4-7)^2+(9-4.33)^2} \approx 5.59$

New Centroid 3 = (1.5, 3.5)

1. A1 = $\sqrt{(2-1.5)^2+(10-3.5)^2} \approx 6.52$
2. A2 = $\sqrt{(2-1.5)^2+(5-3.5)^2} \approx 1.58$
3. A3 = $\sqrt{(8-1.5)^2+(4-3.5)^2} \approx 6.52$
4. B1 = $\sqrt{(5-1.5)^2+(8-3.5)^2} \approx 5.70$
5. B2 = $\sqrt{(7-1.5)^2+(5-3.5)^2} \approx 5.79$
6. B3 = $\sqrt{(6-1.5)^2+(4-3.5)^2} \approx 4.61$
7. C1 = $\sqrt{(1-1.5)^2+(2-3.5)^2} \approx 1.58$
8. C2 = $\sqrt{(4-1.5)^2+(9-3.5)^2} \approx 6.52$

Reassigning Points to Clusters

After calculating distances for each point to the new centroids, we reassign each point to the nearest cluster:

1. **Cluster 1 (Centroid 1 = (3.67, 9))**: Points A1, B1, and C2.
2. **Cluster 2 (Centroid 2 = (7, 4.33))**: Points A3, B2, B3.
3. **Cluster 3 (Centroid 3 = (1.5, 3.5))**: Points A2, C1.

Rana Husni

These new assignments may lead to convergence, where the centroids no longer move significantly, indicating the algorithm has found stable clusters. You can continue updating centroids and reassigning points until convergence is achieved.

In Python

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
# Define the data points
data = np.array([
   [2, 10],  # A1
   [2, 5],   # A2
   [8, 4],   # A3
   [5, 8],   # B1
   [7, 5],   # B2
   [6, 4],   # B3
   [1, 2],   # C1
   [4, 9],   # C2
])
# Set the number of clusters
num_clusters = 3
# Initialize and fit K-means
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
kmeans.fit(data)
# Get the cluster labels for each data point
labels = kmeans.labels_
# Get the cluster centroids
centroids = kmeans.cluster_centers_
# Plotting the data points with cluster assignment
plt.scatter(data[:, 0], data[:, 1], c=labels, label='Data points')
plt.scatter(centroids[:, 0], centroids[:, 1], s=200, c='red', label='Centroids')
# Centroids marked in red
plt.xlabel('X Coordinate')
plt.ylabel('Y Coordinate')
plt.title('K-means Clustering with 3 Clusters')
plt.legend()
plt.show()
```