Text Clustering

CPE 393: Text Analytics

Dr. Sansiri Tarnpradab

Department of Computer Engineering
King Mongkut's University of Technology Thonburi

Intro	Pattern Matching	Text Visualization	Web Scraping
Text Preparation	Text Feature Representation	Text Classification	Text Summarization
Text Clustering	Topic Modeling	TBA Advanced Topic	???

Outline

- Introducing clustering concepts
- Similarity
- Distance functions
- Quality of clustering
- Clustering: Partitioning-based method
- K-means clustering
- Lab

Cluster Analysis

Cluster

A collection of data objects

Similarity

Dissimilarity

Cluster Analysis

Finding Similarities

Characteristics

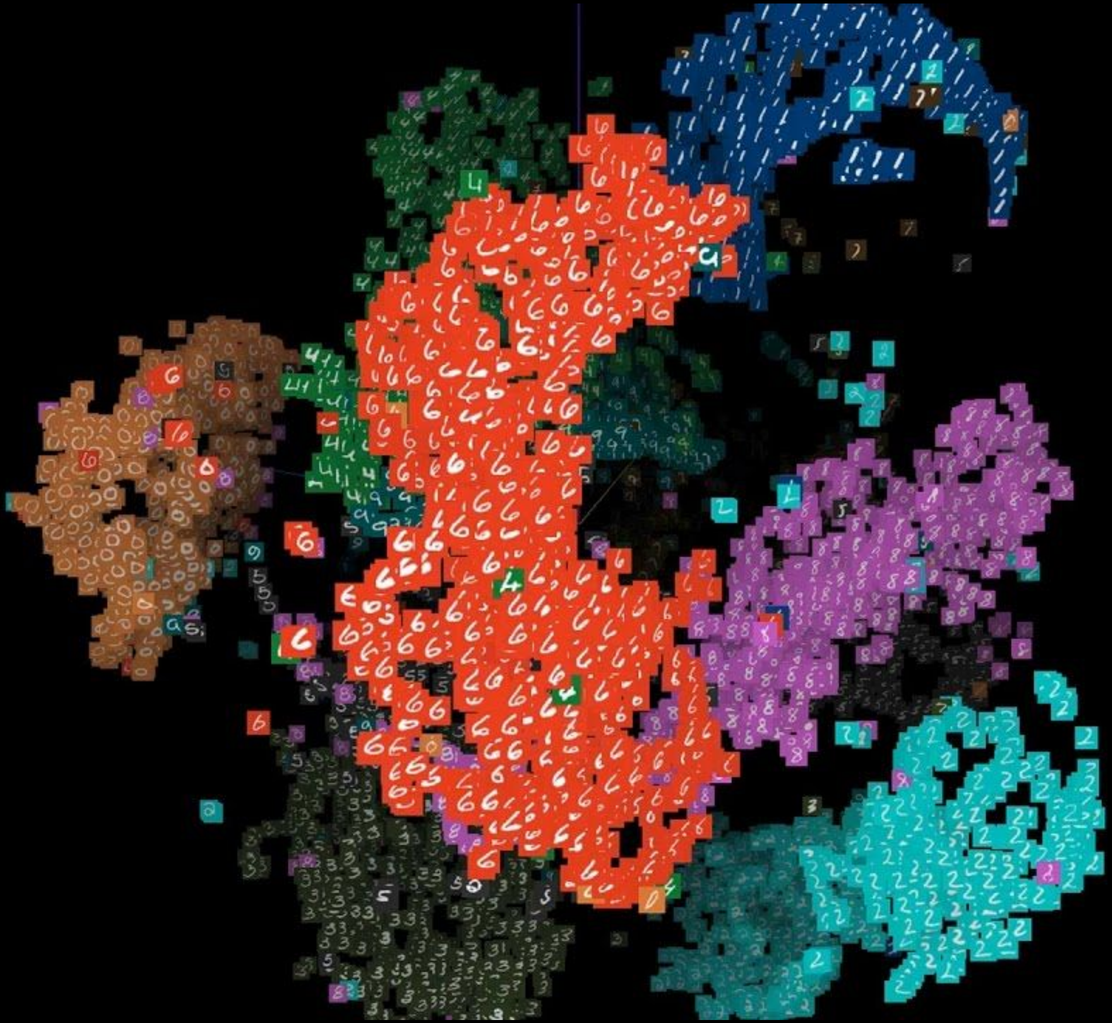
Grouping based on similarities

Similarity

- If two things are similar in some ways, they often share other characteristics.

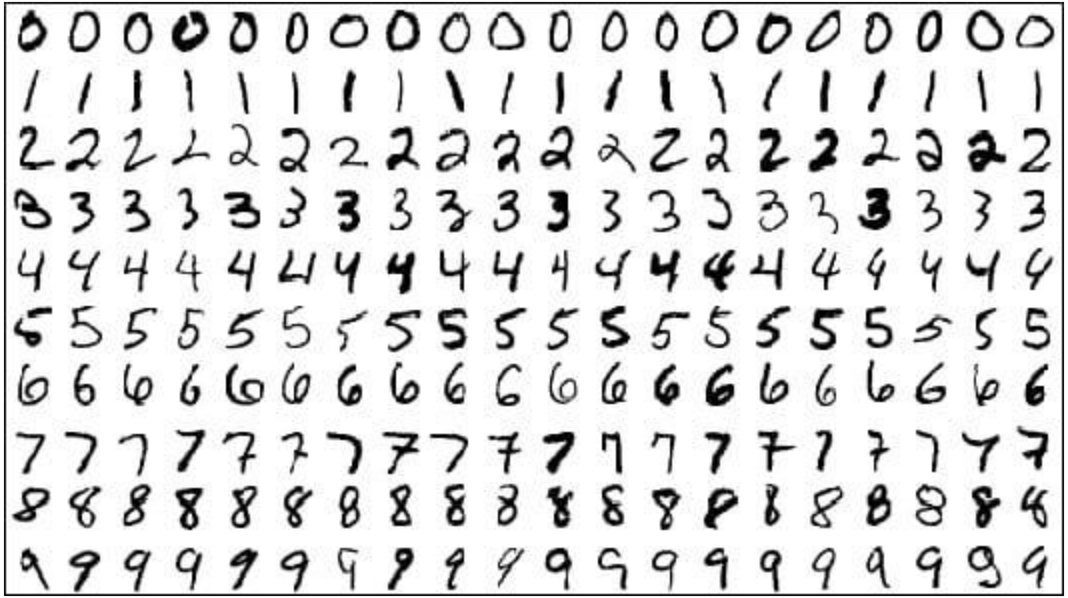
- Applications:
  - **Recommendations**
    - Online Shopping: Amazon
    - Social Media: Facebook
    - Netflix, Hulu, Disney+
  - **Reasoning**
    - Troubleshooting
    - Knowledge Management
  - **Customer Segmentation**
    - What customers have in common
  - Even **Classification** or **Regression**

Those similar to each other are closer in distance.



Ref: https://www.youtube.com/watch?v=wvsE8jm1GzE

MNIST 0-9



Ref: https://medium.com/@navyashree.raghupatro/recognizing-handwritten-digits-with-scikit-learn-8d248dc01b6d

___Recall this?___

Distributed
Text Representation

SIMILARITY MEASURE

Two common methods to measure a distance between vectors in a vector space.

Euclidean Distance

$$euclidean(u, v) = \sqrt{\sum_{i=1}^{n} |u_i - v_i|^2}$$

Cosine Similarity

Dot product of the vectors divided by the product of their magnitudes.

$$\cos(\theta) = \frac{u \cdot v}{\|u\|\|v\|} = \frac{\sum_{i=1}^{n} u_i \times v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}$$

Distance *Functions*

Euclidean Distance (L2 norm)

$$d_{Euclidean}(X,Y) = ||X - Y||_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots}$$

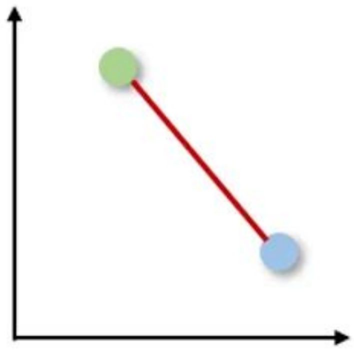
Manhattan Distance (L1 norm)

$$d_{Manhattan}(X,Y) = ||X - Y||_1 = |x_1 - y_1| + |x_2 - y_2| + \ldots$$

Cosine Distance

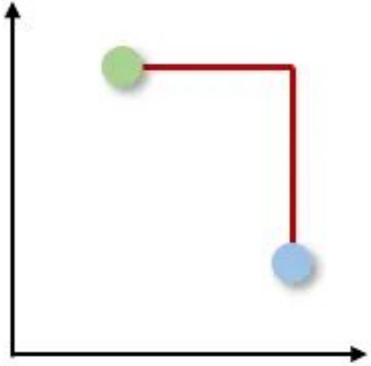
$$d_{Cosine}(X,Y) = \frac{X \cdot Y}{||X|| \times ||Y||}$$

Euclidean



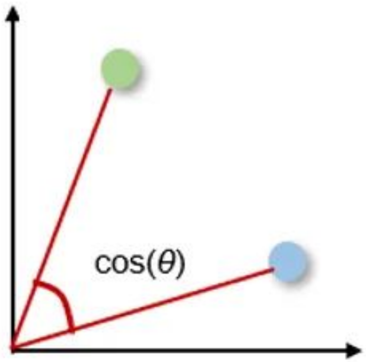
- Shortest distance between two real-valued vectors
- Most common

Manhattan



- Taxicab or City-block distance
- Shortest distance between two real-valued vectors
- Right angles

Cosine



- Cosine between two vectors
- Often used in higher dimensionality
- Measured in Θ
  - Θ = 0° → similar (overlap)
  - Θ = 90° → dissimilar

Ref: https://medium.com/@jodancker/a-brief-introduction-to-distance-measures-ac89cbd2298

Example

Attributes	A	B
Age	23	40
Years residing at the current address	2	10
Residential status (1 = owner, 2 = renter, 3 = others)	2	1

$$d(A, B) = \sqrt{(23 - 40)^2 + (2 - 10)^2 + (2 - 1)^2}$$
$$\approx 18.8$$

Not meaningful

Not enough context (there's only A and B)

Example: More Data Points

Customer	Age	Income (~k)	Cards	Response	Distance from David
David	37	50	2	?	0
John	35	35	3	Y	
Rachel	22	50	2	N	
Bob	63	200	1	N	
Jeffrey	59	170	1	N	
Norah	25	40	4	Y	

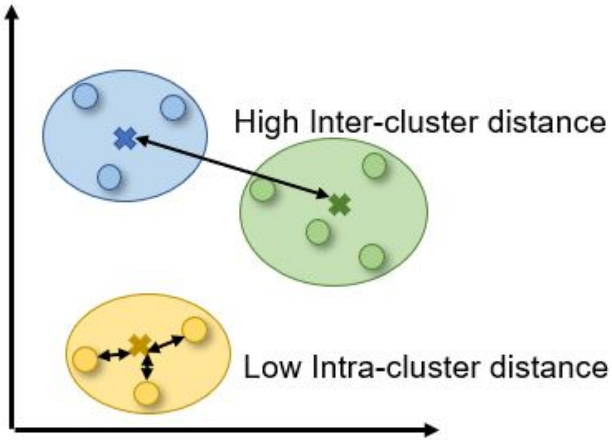
Issues

- Attribute may not all be numeric
- Values are not in the same range
- Some preprocessing is needed
  - Scaled
  - Normalized

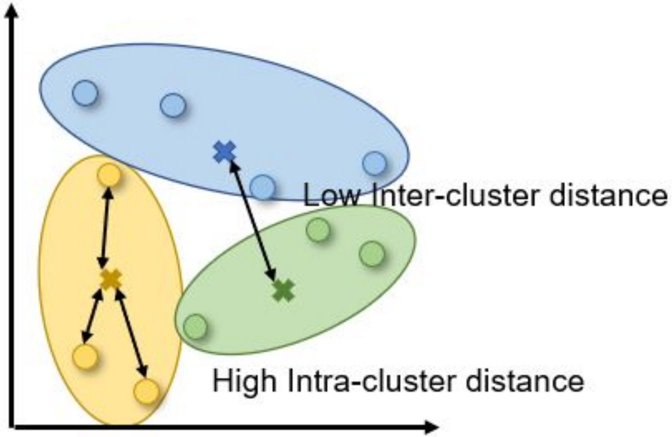
Back to (Text) Clustering...

Quality of Clustering

- High *intra*-class similarity
- Low *inter*-class similarity



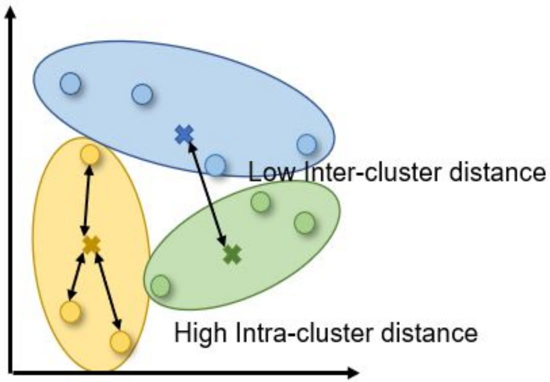
Good Example

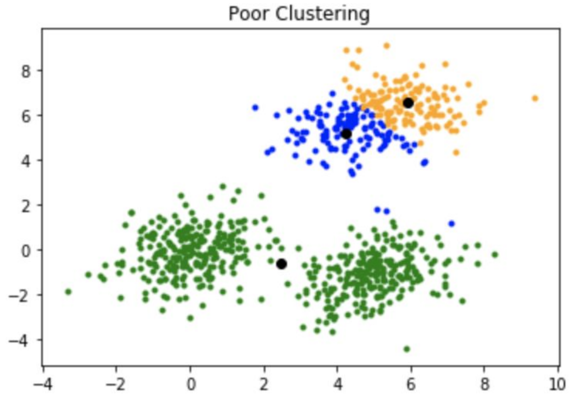


Bad Example

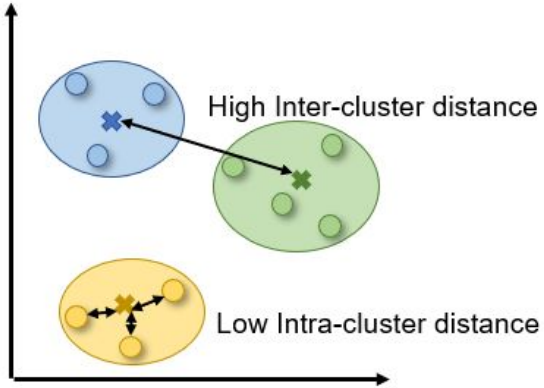
Ref: https://medium.com/@jodancker/a-brief-introduction-to-cluster-validation-ca4215295b06

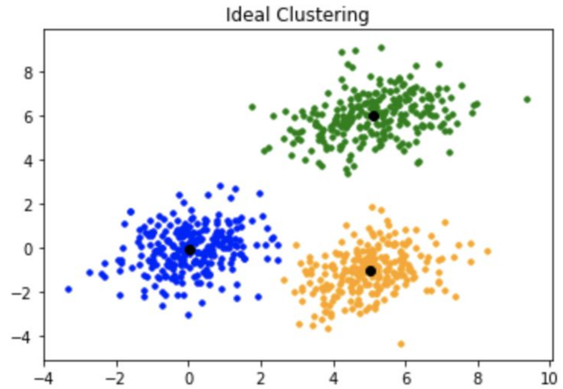
Bad
Example





Good
Example





Refs:
https://www.geeksforgeeks.org/ml-k-means-algorithm/
https://medium.com/@jodancker/a-brief-introduction-to-cluster-validation-ca4215295b06

Partitioning

- Breaking down a large group of data points into partitions
- While still taking into account the distance → minimum

Basic Concept

Construct a partition of a database D of n objects into a set of k clusters, such that sum of squared distance is minimal

Basic Concept

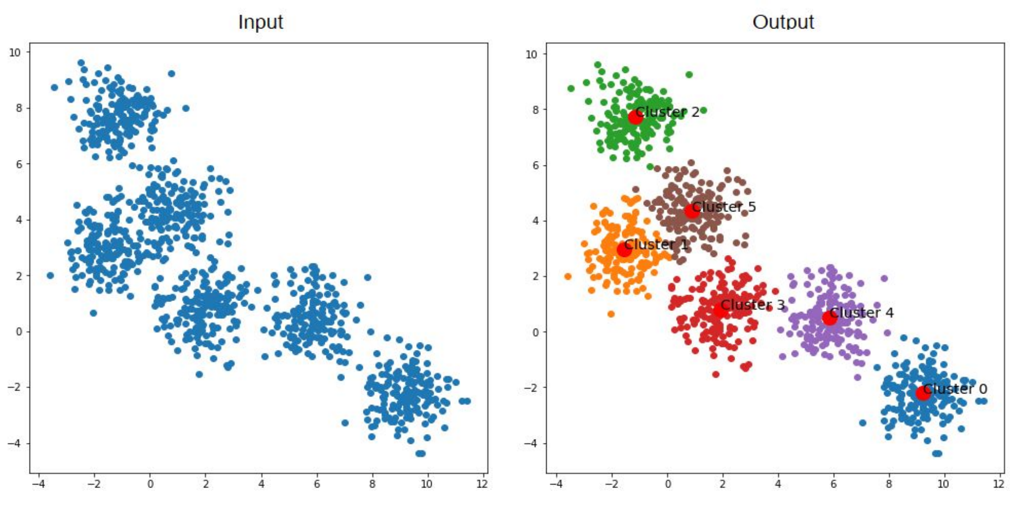
Construct a partition of a database D of n objects into a set of k clusters, such that sum of squared distance is minimal

k=2 k=3 k=4 k=5 k=...

Computationally Infeasible

Partitioning: *K-means*

- Each cluster is represented by the center of the cluster
- **Centroid**
  - Center of the cluster → Average



Ref: https://towardsai.net/p/l/centroid-neural-network-an-efficient-and-stable-clustering-algorithm

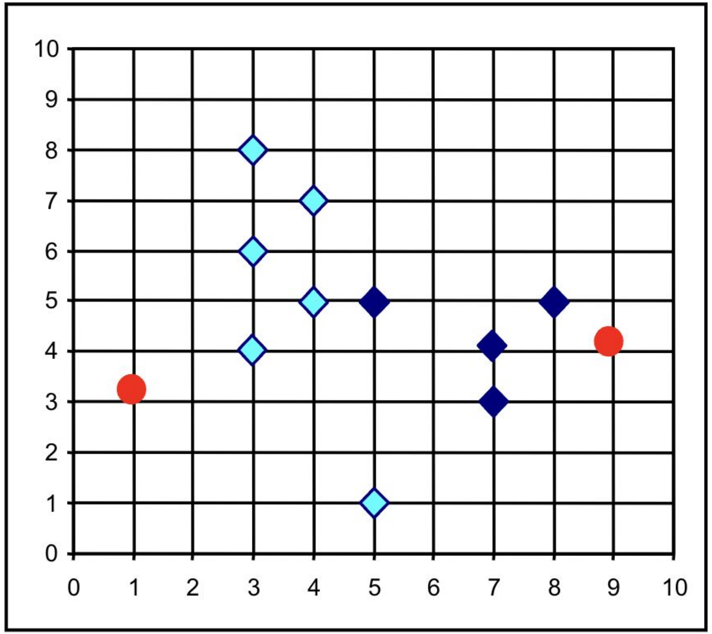
K-means: *Steps*

1. Partition objects into k non-empty subsets.
2. Compute seed points as the centroids of the clusters of the current partition.
3. Assign each object to the cluster with the nearest seed point.
4. Go back to Step 2, stop when no more new assignment.

K-means:
Steps (1-2)

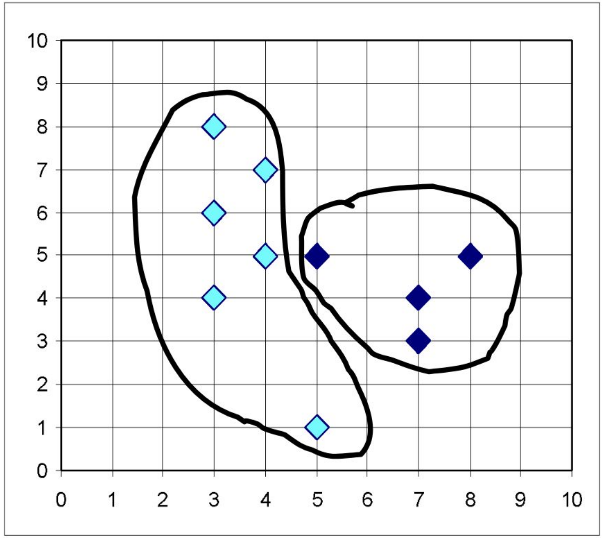
1. Partition objects into k non-empty subsets. (k=2)

2. Compute seed points as the centroids of the clusters of the current partition.



K-means:
Steps (3)

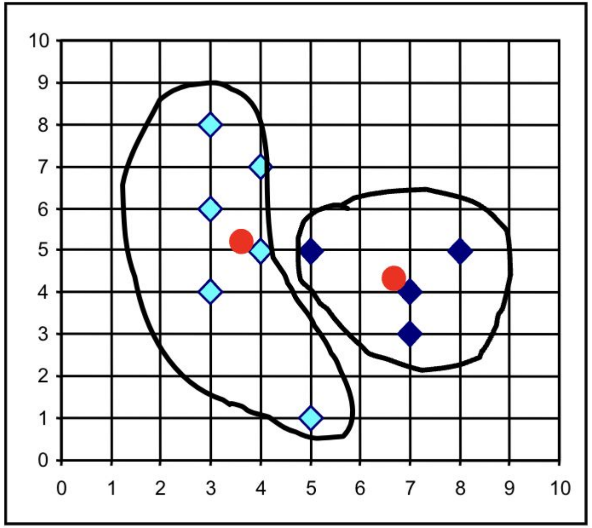
3. Assign each object to the cluster with the nearest seed point.

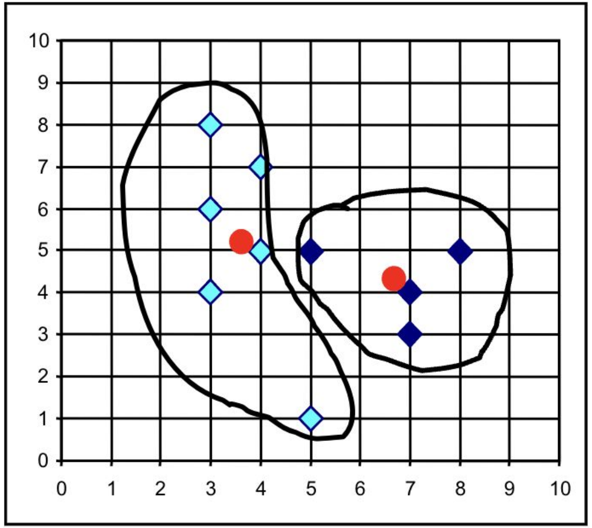


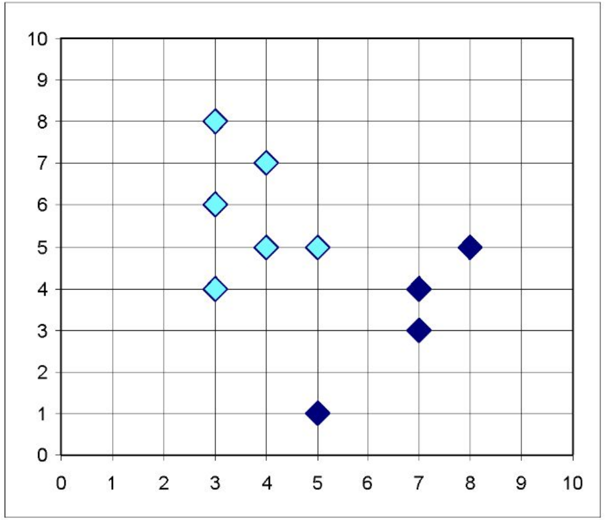
K-means: *Steps (4)*

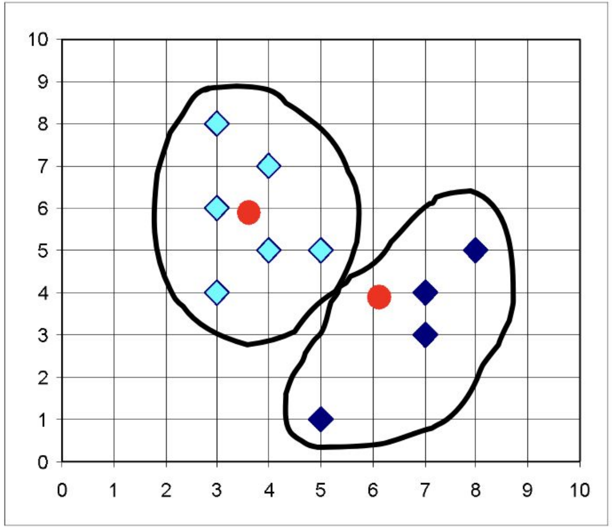
4. Go back to **Step 2**, stop when no more new assignment.

Step 2: Compute seed points as the centroids of the clusters of the current partition.




0 1 2 3 4 5 6 7 8 9 10


0 1 2 3 4 5 6 7 8 9 10


0 1 2 3 4 5 6 7 8 9 10

Clustering:
Customer Complaints

Task:

- Use the customer complaints dataset
- Perform data preprocessing
- Use a proper text representation
- Perform text clustering (k-means)
    - Try different number of k
- Print terms in each cluster
- Interpret results

```
from sklearn.cluster import  KMeans
km = KMeans(n_clusters=11, random_state=42).fit(X)
```

```
order_centroids = km.argsort()[:, ::-1]
terms = vectorizer.get_feature_names_out()

for i in range(k):
    print(f"Cluster {i}: ", end="")
    for ind in order_centroids[i, :10]:
        print(f"{terms[ind]} ", end="")
    print()
```

Conclusion

- Introducing clustering concepts
- Similarity
- Distance functions
- Quality of clustering
- Clustering: Partitioning-based method
- K-means clustering

Q & A