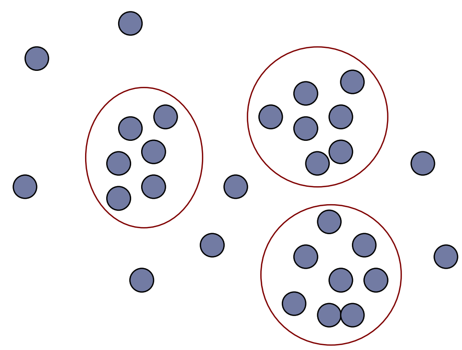
Chapter 3: Cluster Analysis

- **3.1 Basic Concepts of Clustering**
  - 3.1.1 Cluster Analysis
  - 3.1.2 Clustering Categories
- **3.2 Partitioning Methods**
  - 3.2.1 The principle
  - 3.2.2 K-Means Method
  - 3.2.3 K-Medoids Method
  - 3.2.4 CLARA
  - 3.2.5 CLARANS
- **3.3 Hierarchical Methods**
- **3.4 Density-based Methods**
- **3.5 Clustering High-Dimensional Data**
- **3.6 Outlier Analysis**

3.1.1 Cluster Analysis

- Unsupervised learning (i.e., Class label is unknown)
- Group data to form new categories (i.e., clusters), e.g., cluster houses to find distribution patterns
- Principle: Maximizing intra-class similarity & minimizing interclass similarity



- **Typical Applications**
  - → WWW, Social networks, Marketing, Biology, Library, etc.

3.1.2 Clustering Categories

- **Partitioning Methods**
  - → Construct k partitions of the data
- **Hierarchical Methods**
  - → Creates a hierarchical decomposition of the data
- **Density-based Methods**
  - → Grow a given cluster depending on its density (# data objects)
- **Grid-based Methods**
  - → Quantize the object space into a finite number of cells
- **Model-based methods**
  - → Hypothesize a model for each cluster and find the best fit of the data to the given model
- **Clustering high-dimensional data**
  - → Subspace clustering
- **Constraint-based methods**
  - → Used for user-specific applications

Chapter 3: Cluster Analysis

- **3.1 Basic Concepts of Clustering**
  - 3.1.1 Cluster Analysis
  - 3.1.2 Clustering Categories
- **3.2 Partitioning Methods**
  - 3.2.1 The principle
  - 3.2.2 K-Means Method
  - 3.2.3 K-Medoids Method
  - 3.2.4 CLARA
  - 3.2.5 CLARANS
- **3.3 Hierarchical Methods**
- **3.4 Density-based Methods**
- **3.5 Clustering High-Dimensional Data**
- **3.6 Outlier Analysis**

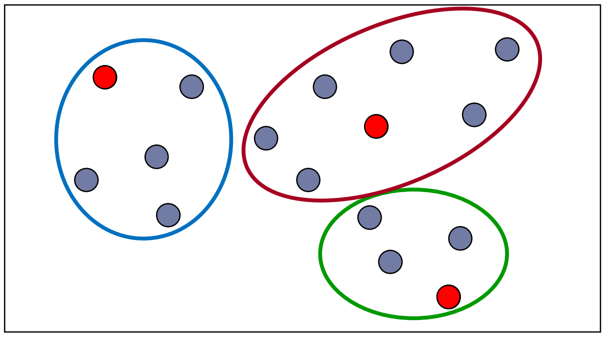
3.2.1 Partitioning Methods: The Principle

- Given
  - A data set of **n** objects
  - **K** the number of clusters to form
- Organize the objects into k partitions **(k<=n)** where each partition represents a cluster
- The clusters are formed to optimize an objective partitioning criterion
  - Objects within a cluster are **similar**
  - Objects of different clusters are **dissimilar**

3.2.2 K-Means Method

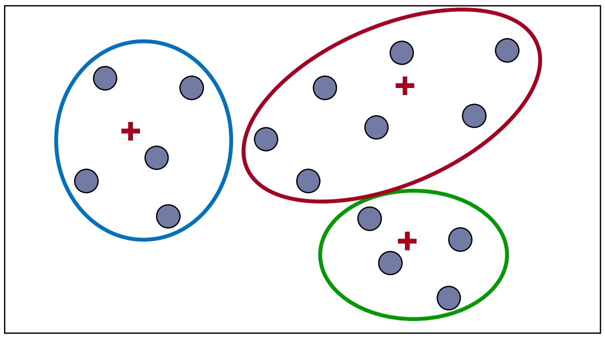
Choose 3 objects (cluster centroids)

Assign each object to the closest centroid to form Clusters



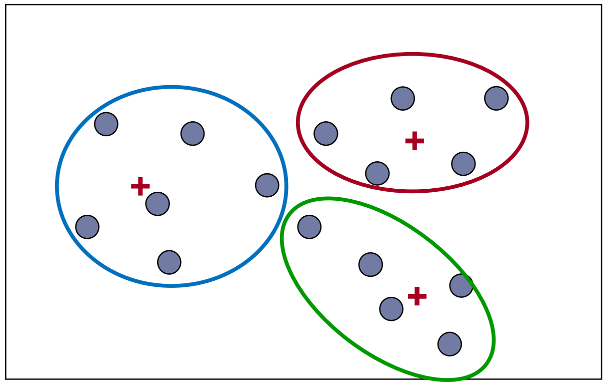
Goal: create 3 clusters (partitions)

Update cluster centroids

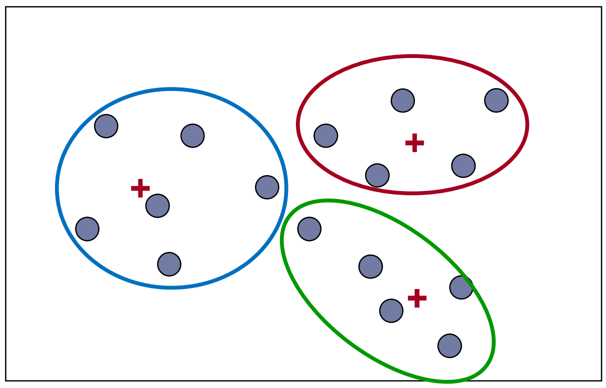


K-Means Method

Recompute Clusters



If Stable centroids, then stop



K-Means Algorithm

- **Input**
  - K: the number of clusters
  - D: a data set containing n objects
- **Output**: A set of k clusters
- **Method**:

(1) Arbitrary choose k objects from D as in initial cluster centers
(2) **Repeat**
(3) Reassign each object to the most similar cluster based on the mean value of the objects in the cluster
(4) Update the cluster means
(5) **Until** no change

K-Means Properties

- The algorithm attempts to determine k partitions that minimize the square-error function

$$E = \sum_{i-1}^{k} \sum_{p \in C_i} (p - m_i)^2$$

  - **E**: the sum of the squared error for all objects in the data set
  - **P**: the data point in the space representing an object
  - **m_i**: is the mean of cluster C_i

- It works well when the clusters are compact clouds that are rather well separated from one another

K-Means Properties

Advantages

- K-means is relatively scalable and efficient in processing large data sets
- The computational complexity of the algorithm is **O(nkt)**
  - → **n:** the total number of objects
  - → **k:** the number of clusters
  - → **t:** the number of iterations
  - → **Normally:** k<<n and t<<n

Disadvantage

- Can be applied only when the mean of a cluster is defined
- Users need to specify k
- K-means is not suitable for discovering clusters with nonconvex shapes or clusters of very different size
- It is sensitive to noise and outlier data points (can influence the mean value)

Variations of the K-Means Method

- A few variants of the ***k-means*** which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: ***k-modes*** (Huang'98)
  - Replacing means of clusters with <u>modes</u>
  - Using new dissimilarity measures to deal with categorical objects
  - Using a <u>frequency</u>-based method to update modes of clusters
  - A mixture of categorical and numerical data

3.2.3 K-Medoids Method

- Minimize the sensitivity of k-means to outliers
- Pick actual objects to represent clusters instead of mean values
- Each remaining object is clustered with the representative object (**Medoid**) to which is the most similar
- The algorithm minimizes the sum of the dissimilarities between each object and its corresponding reference point

$$E = \sum_{i-1}^{k} \sum_{p \in C_i} | p - o_i |$$

  - → **E**: the sum of absolute error for all objects in the data set
  - → **P**: the data point in the space representing an object
  - → $\mathbf{O_i}$: is the representative object of cluster C_i

K-Medoids Method: The Idea

- Initial representatives are chosen randomly

- The iterative process of replacing representative objects by no representative objects continues as long as the quality of the clustering is improved

- For each representative Object O

  - → For each non-representative object R, swap O and R

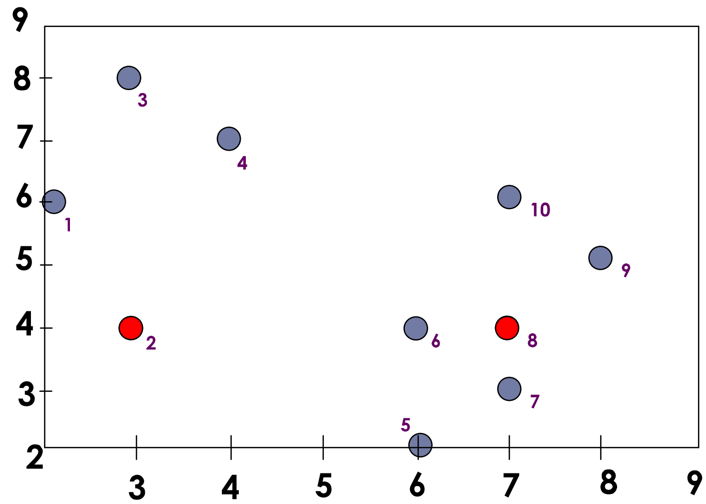
- Choose the configuration with the lowest cost

- Cost function is the difference in absolute error-value if a current representative object is replaced by a non-representative object

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



Goal: create two clusters

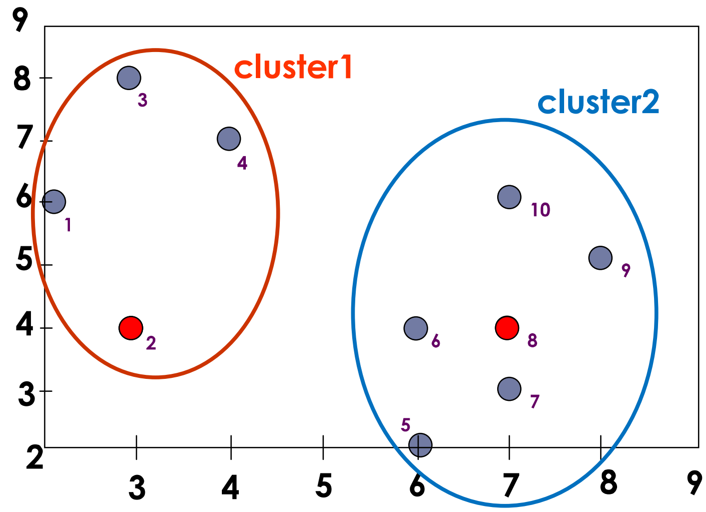
Choose randmly two medoids

$O_2 = (3,4)$
$O_8 = (7,4)$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Assign each object to the closest representative object

→Using L1 Metric (Manhattan), we form the following clusters

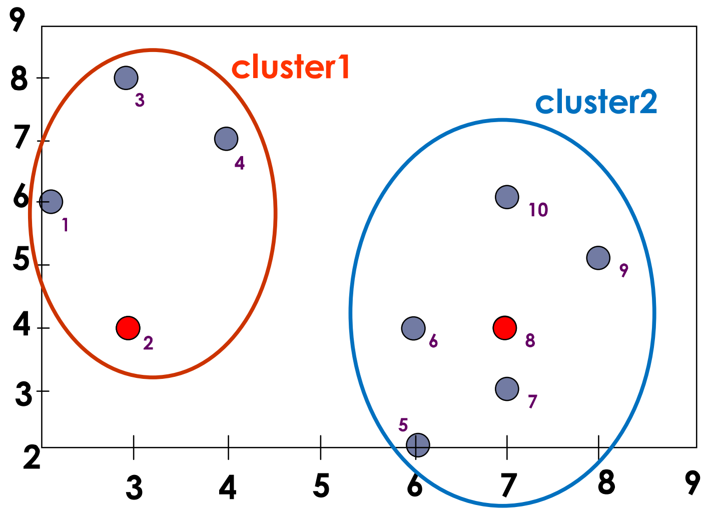
Cluster1 = $\{O_1, O_2, O_3, O_4\}$

Cluster2 = $\{O_5, O_6, O_7, O_8, O_9, O_{10}\}$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Compute the absolute error criterion **[for the set of Medoids (O2,O8)]**

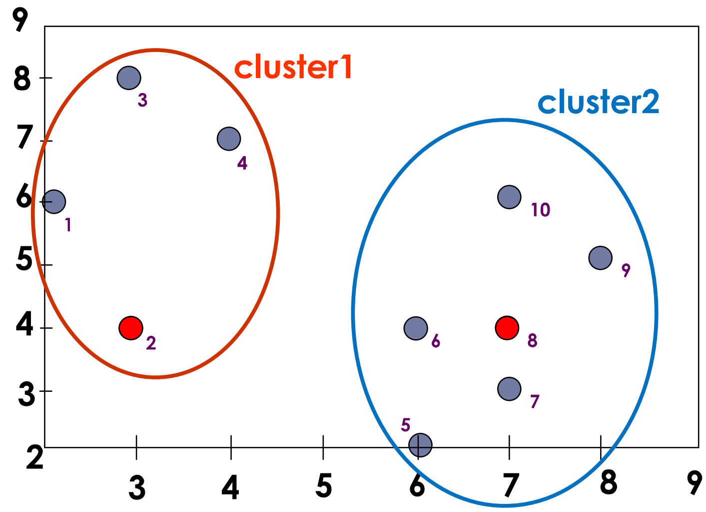
$$E=\sum_{i=1}^{k}\sum_{p\in C_i}|p-o_i|=|o_1-o_2|+|o_3-o_2|+|o_4-o_2|$$

$$+|o_5-o_8|+|o_6-o_8|+|o_7-o_8|+|o_9-o_8|+|o_{10}-o_8|$$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



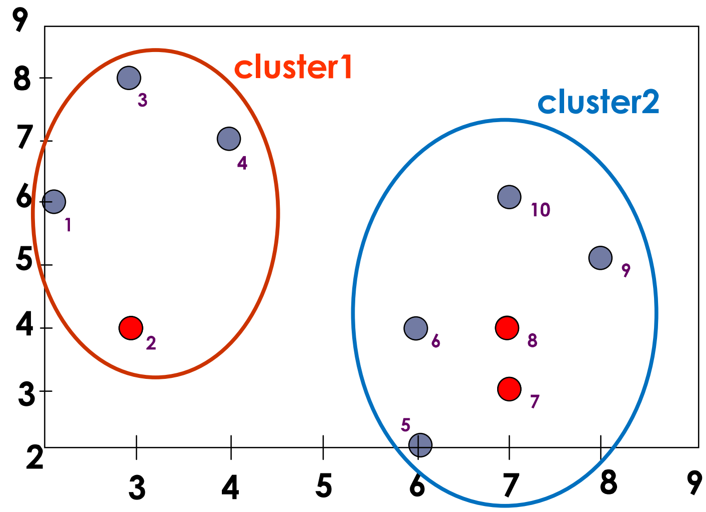
→The absolute error criterion **[for the set of Medoids (O2,O8)]**

$$E = (3+4+4) + (3+1+1+2+2) = 20$$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Choose a random object O_7

→Swap **O8** and **O7**

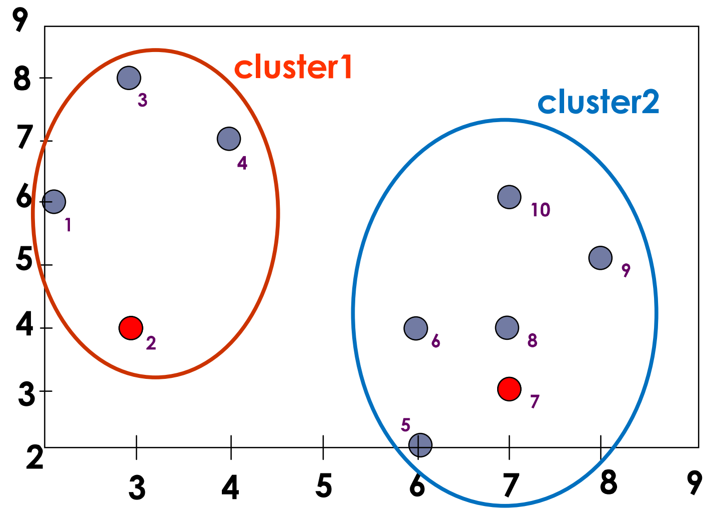
→Compute the absolute error criterion **[for the set of Medoids (O2,O7)]**

$$E = (3+4+4)+(2+2+1+3+3) = 22$$

K-Medoids Method: Example

Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



→Compute the cost function

Absolute error [for O_2,O_7] – Absolute error [O_2,O_8]

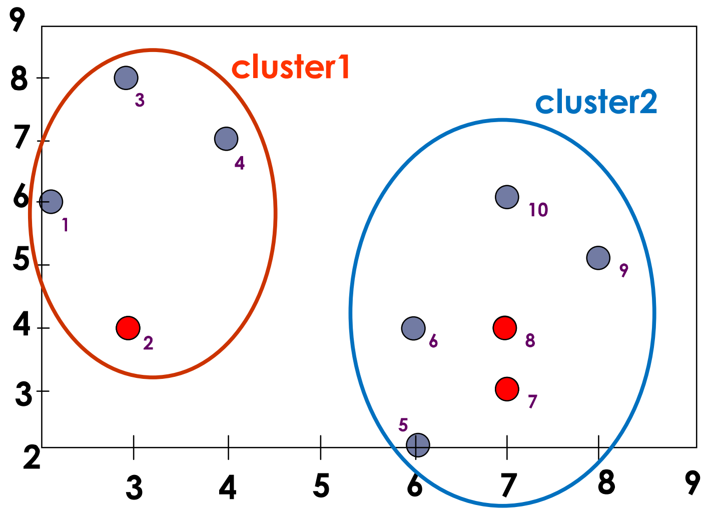
$$S = 22 - 20$$

S> 0 ⇒ it is a bad idea to replace O_8 by O_7

K-Medoids Method

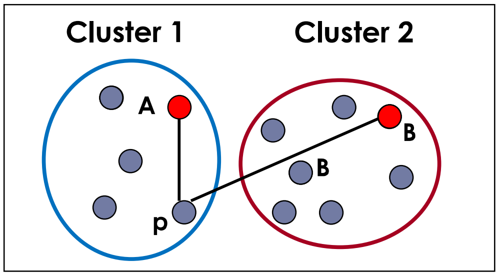
Data Objects

	A_1	A_2
O_1	2	6
O_2	3	4
O_3	3	8
O_4	4	7
O_5	6	2
O_6	6	4
O_7	7	3
O_8	7	4
O_9	8	5
O_{10}	7	6



- In this example, changing the medoid of cluster 2 did not change the assignments of objects to clusters.
- What are the possible cases when we replace a medoid by another object?

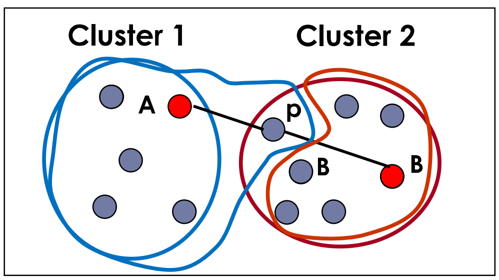
K-Medoids Method



● Representative object
● Random Object
Currently **P** assigned to **A**

First case

The assignment of **P** to **A** does **not change**

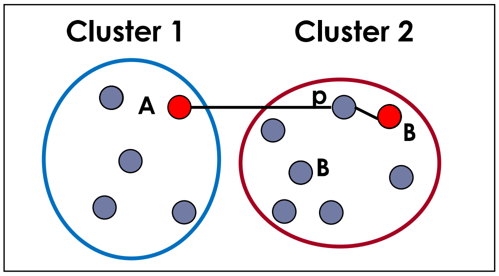


● Representative object
● Random Object
Currently **P** assigned to **B**

Second case

P is **reassigned** to **A**

K-Medoids Method



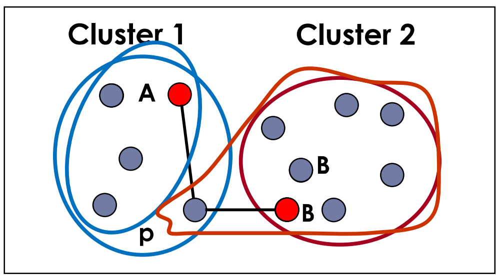
● Representative object

● Random Object

Currently **P** assigned to **B**

Third case

P is **reassigned** to **the new B**



● Representative object

● Random Object

Currently **P** assigned to **A**

Fourth case

P is **reassigned** to **B**