Data Mining & Data Warehouse

Dr. Raed Ibraheem Hamed

University of Human Development,
College of Science and Technology
Department of Information Technology

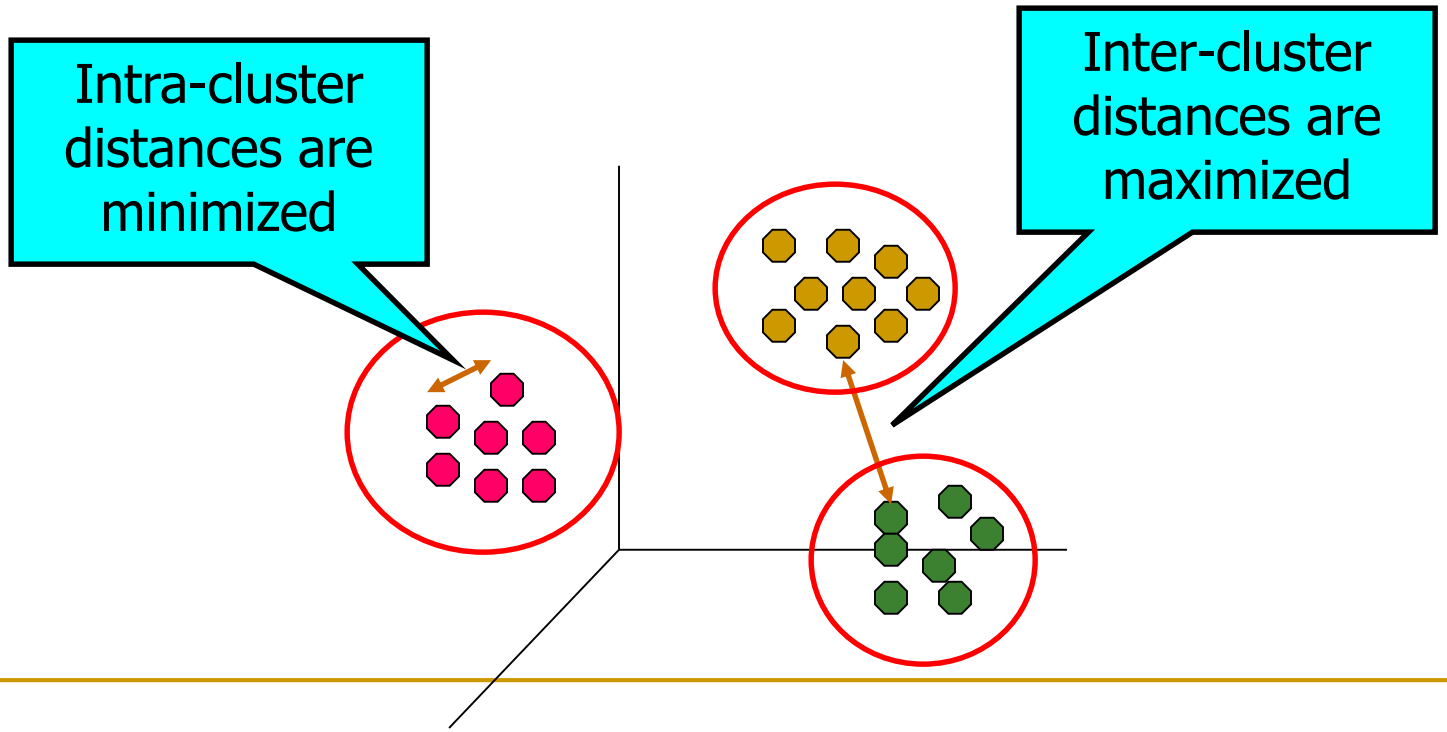
2016 – 2017

Road map

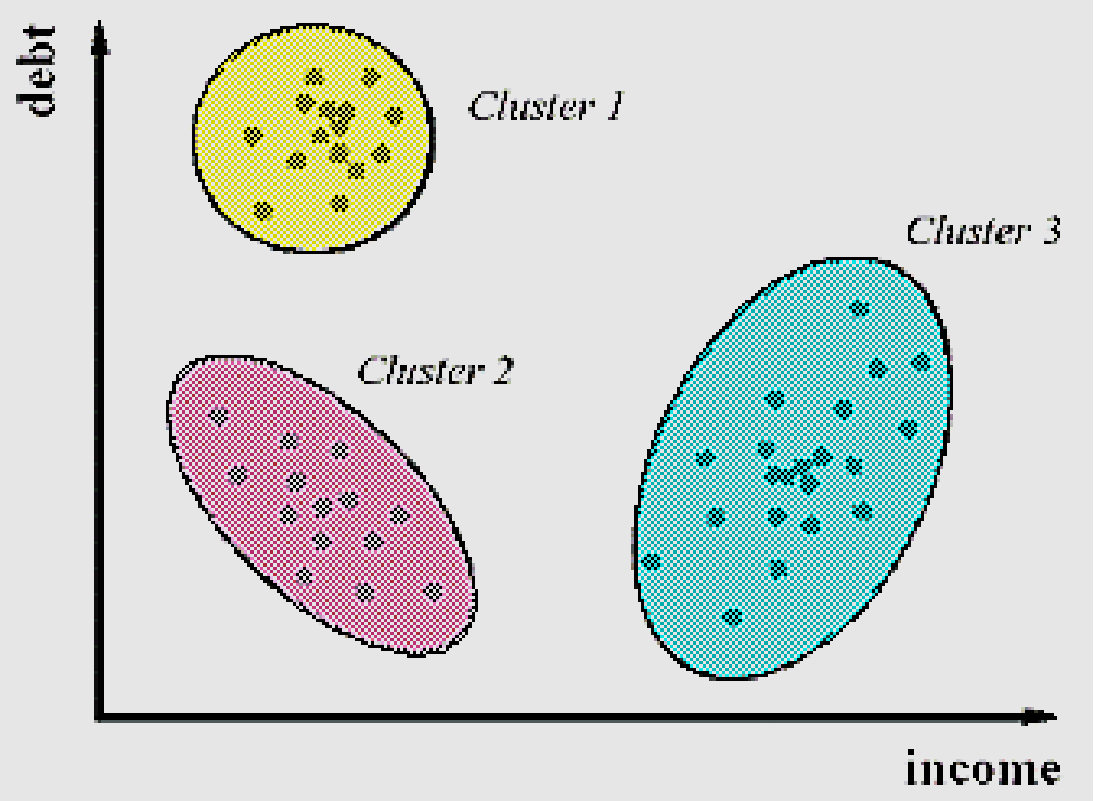
- What is Cluster Analysis?
- Characteristics of Clustering
- Applications of Cluster Analysis
- Clustering: Application Examples
- Basic Steps to Develop a Clustering Task
- Quality: What Is Good Clustering?
- k-Means Clustering
- The *K-Means* Clustering Method
- What is the problem of k-Means Method?
- The K-Medoids Clustering Method

What is Cluster Analysis?

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar to each other than to those in other groups (clusters).

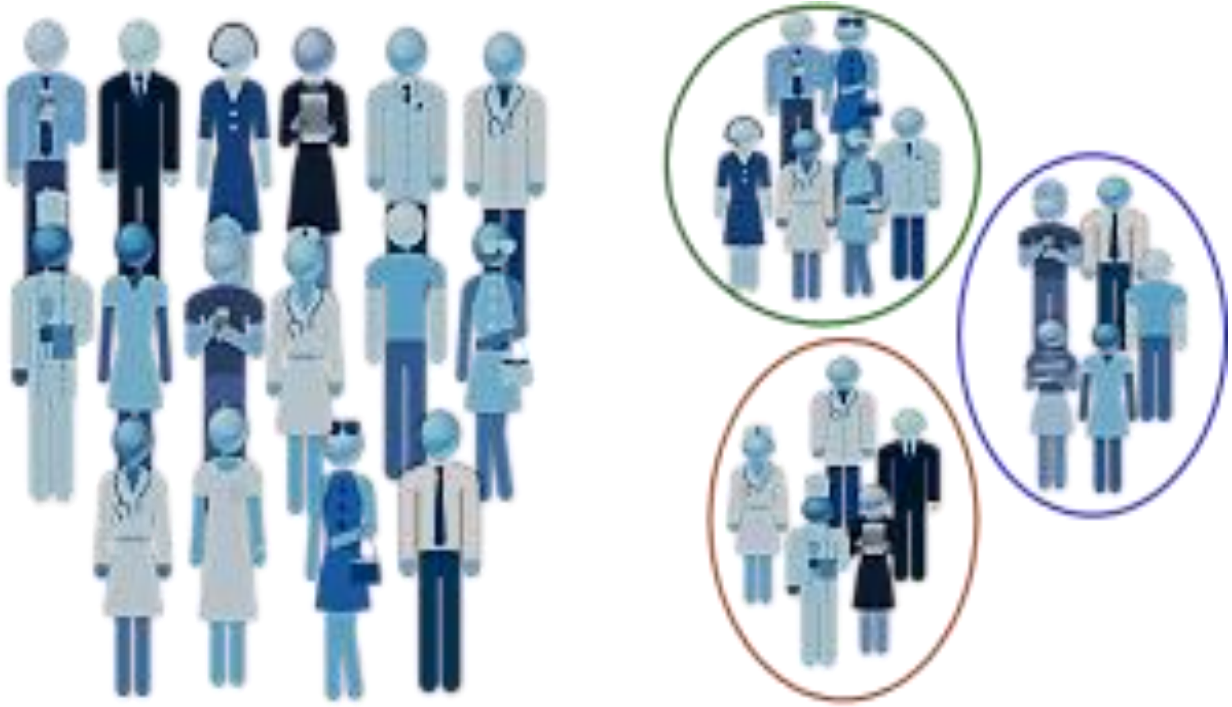


Understanding data mining clustering methods



Understanding data mining clustering methods

you could first cluster your customer data into groups that have similar structures and then plan different marketing operations for each group.



What is the difference between supervised and unsupervised learning?

In **supervised learning**, the output datasets are provided which are used to train the machine and get the desired outputs.

whereas in **unsupervised learning** no output datasets are provided, instead the data is clustered into different classes .

Characteristics of Clustering

- **Cluster analysis (or clustering)**
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters

- **Unsupervised learning**: no predefined classes .

- **Typical applications**
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Applications of Cluster Analysis

1. **Data reduction**
   - Summarization and Compression
2. **Prediction based on groups**
   - Find characteristics/patterns for each group
3. **Finding K-nearest Neighbors**
   - Centralize the search to one or a small number of clusters
4. **Outlier detection**: Outliers are often viewed as those "far away" from any cluster

Clustering: Application Examples

1. **Biology**: family, genes , species, …
2. **Information retrieval**: document clustering
3. **Land use**: Identification of areas of similar land use in an earth observation database
4. **Marketing**: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs.

Clustering: Application Examples

5. **City-planning**: Identifying groups of houses according to their house type, value, and geographical location
6. **Earth-quake studies**: Observed earth quake epicenters should be clustered.
7. **Climate**: understanding earth climate, find patterns of atmospheric and ocean
8. **Economic Science**: market research

Basic Steps to Develop a Clustering Task

1. **Feature selection**
   - ✓ Select information concerning the task of interest
2. **Proximity measure**
   - ✓ Similarity of two feature vectors
3. **Clustering criterion**
   - ✓ Expressed via a cost function or some rules
4. **Clustering algorithms**
   - ✓ Choice of algorithms
5. **Validation of the results**
   - ✓ Validation test
6. **Interpretation of the results**
   - ✓ Integration with applications

Quality: What Is Good Clustering?

1. A good clustering method will produce high quality clusters
   - high intra-class similarity
   - low inter-class similarity
2. The quality of a clustering method depends on
   - the similarity measure used by the method
   - its implementation, and
   - Its ability to discover some or all of the hidden patterns

Major Clustering Approaches

1. Partitioning approach:
2. Hierarchical approach:
3. Density-based approach:
4. Grid-based approach:

Partitioning Algorithms: Basic Concept

Partitional clustering decomposes a data set into a set of disjoint clusters. Given a data set of ***N*** points, a partitioning method constructs ***K*** partitions of the data, with each partition representing a cluster. That is, it classifies the data into ***K*** groups by satisfying the following requirements:

(1) Each group contains at least one point,

(2) Each point belongs to exactly one group.

Partitioning Algorithms: Basic Concept

Methods: *k-means* and *k-medoids* algorithms:-

1. k-means : Each cluster is represented by the **center** of the cluster.

2. k-medoids or PAM (**Partition Around Medoids**): Each cluster is represented by one of the objects in the cluster

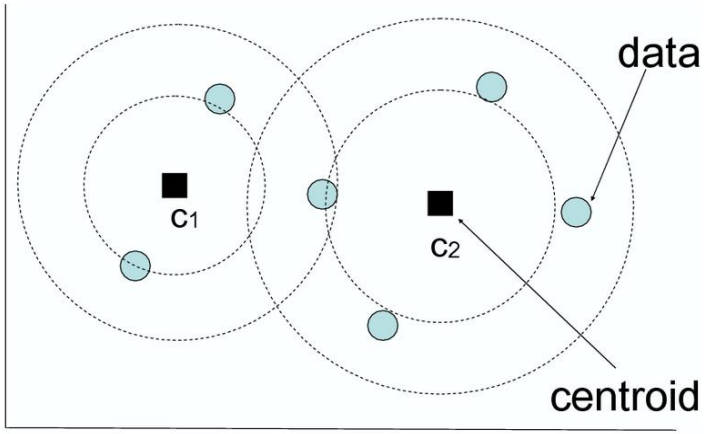
$$\text{cost}(x, c) = \sum_{i=1}^{d} |x_i - c_i| \rightarrow \text{cost}((3,4),(2,6)) = 3$$

k-Means Clustering

- **K-clustering algorithm**
  - **Result**: Given the input set *S* and a fixed integer *k*, a partition of *S* into *k* subsets must be returned.
  - K-means clustering is the most common partitioning algorithm.
  - **K-Means** re-assigns each record in the dataset to **only one** of the new clusters formed. A record or data point is assigned to the **nearest cluster** (the cluster which it is most similar to) using a measure of distance or similarity

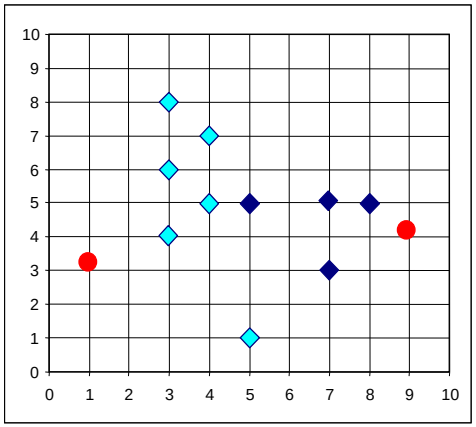
K-Means Clustering Algorithm

1. Separate the objects (data points) into **K** clusters.
2. Cluster center (centroid) = the average of all the data points in the cluster.
3. Assigns each data point to the cluster whose centroid is nearest (using distance function.)
4. Go back to Step 2, stop when the assignment does not change.



The *K-Means* Clustering Method

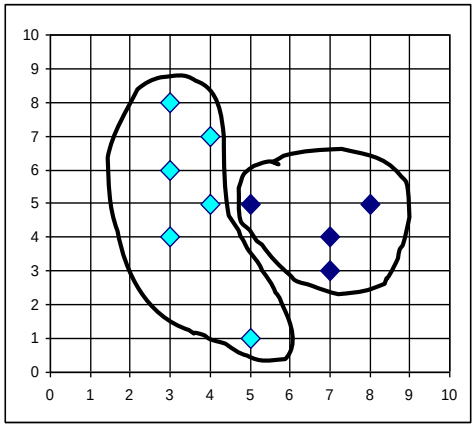
- Example



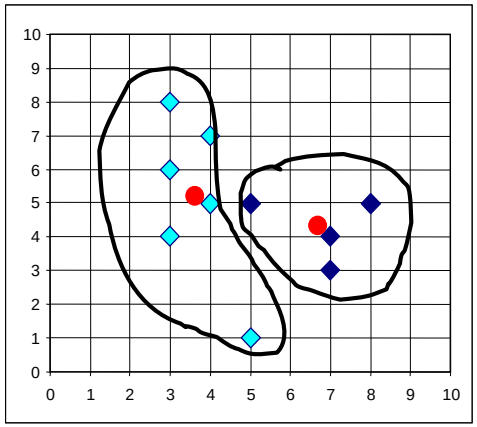
K=2

Arbitrarily choose K object as initial cluster center

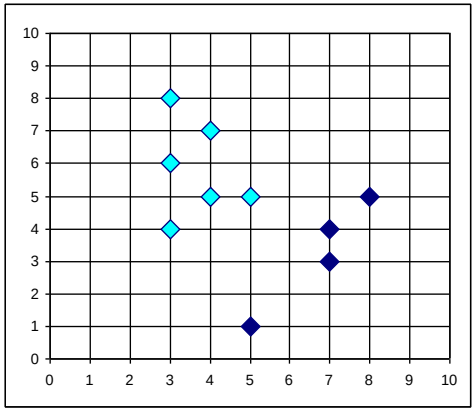
Assign each objects to most similar center



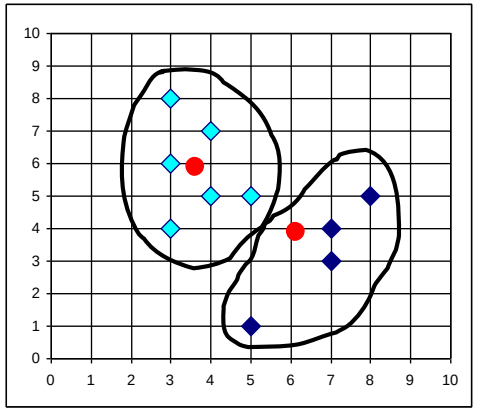
Update the cluster means



reassign



Update the cluster means



reassign

K-mean algorithm

1. It accepts the **number of clusters** to group data into, and the **dataset** to cluster as input values.
2. It then creates the first **K initial clusters** (K= number of clusters needed) from the dataset by choosing K rows of data randomly from the dataset. ***For Example,*** if there are **10,000** rows of data in the dataset and 3 clusters need to be formed, then the first **K=3 initial clusters** will be created by selecting **3 records randomly** from the dataset as the initial clusters. Each of the 3 initial clusters formed will have just one row of data.

K-mean algorithm

3. The K-Means algorithm calculates the **Arithmetic Mean** of each cluster formed in the dataset. *The Arithmetic Mean of a cluster is the mean of all the individual records in the cluster*. In each of the first K initial clusters, their is only one record. The Arithmetic Mean of a cluster with one record is the set of values that make up that record. ***For Example*** if the dataset we are discussing is a set of **Height, Weight and Age** measurements for students in a **UHD**, where a record **P** in the dataset **S** is represented by a Height, Weight and Age measurement, then P = {Age, Height, Weight). Then a record containing the measurements of a student John, would be represented as John = {20, 170, 80} where John's Age = 20 years, Height = 1.70 metres and Weight = 80 Pounds. Since there is only one record in each initial cluster then the Arithmetic Mean of a cluster with only the record for John as a member = {20, 170, 80}.

K-mean algorithm

4. Next, K-Means assigns each record in the dataset to **only one** of the initial clusters. Each record is assigned to the **nearest cluster** (the cluster which it is most similar to) using a measure of distance or similarity like the **Euclidean Distance Measure** or **Manhattan/City-Block Distance Measure**.

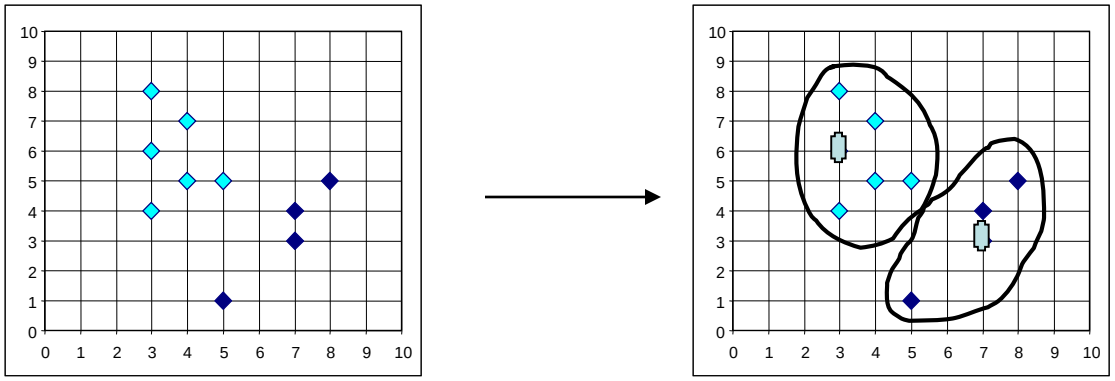
5. K-Means re-assigns each record in the dataset to the most similar cluster and re-calculates the arithmetic mean of all the clusters in the dataset. The arithmetic mean of a cluster is the arithmetic mean of all the records in that cluster. ***For Example***, if a cluster contains two records where the record of the set of measurements for John = {20, 170, 80} and Henry = {30, 160, 120}, then the arithmetic mean $\mathbf{P}_{\mathbf{mean}}$ is represented as $\mathbf{P}_{\mathbf{mean}}$ = {$\mathbf{Age}_{\mathbf{mean}}$, $\mathbf{Height}_{\mathbf{mean}}$, $\mathbf{Weight}_{\mathbf{mean}}$). $\mathbf{Age}_{\mathbf{mean}}$ = (20 + 30)/2, $\mathbf{Height}_{\mathbf{mean}}$ = (170 + 160)/2 and $\mathbf{Weight}_{\mathbf{mean}}$ = (80 + 120)/2. **The arithmetic mean of this cluster = {25, 165, 100}**. This new arithmetic mean becomes the center of this new cluster. Following the same procedure, new **cluster centers** are formed for all the existing clusters.

K-mean algorithm

6. K-Means re-assigns each record in the dataset to **only one** of the new clusters formed. A record or data point is assigned to the **nearest cluster** (the cluster which it is most similar to) using a measure of distance or similarity
7. The preceding steps are repeated until **stable clusters** are formed and the K-Means clustering procedure is completed. Stable clusters are formed when new iterations or repetitions of the K-Means clustering algorithm does not create new clusters as the cluster center or Arithmetic Mean of each cluster formed is the same as the old cluster center. There are different techniques for determining when a stable cluster is formed or when the k-means clustering algorithm procedure is completed.

What is the problem of k-Means Method?

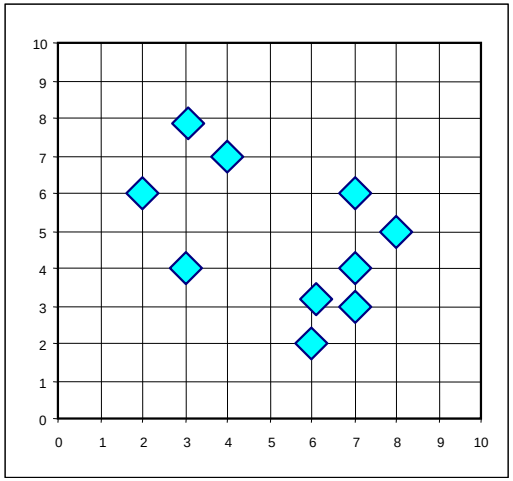
- The k-means algorithm is sensitive to noise and outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.



The K-Medoids Clustering Method

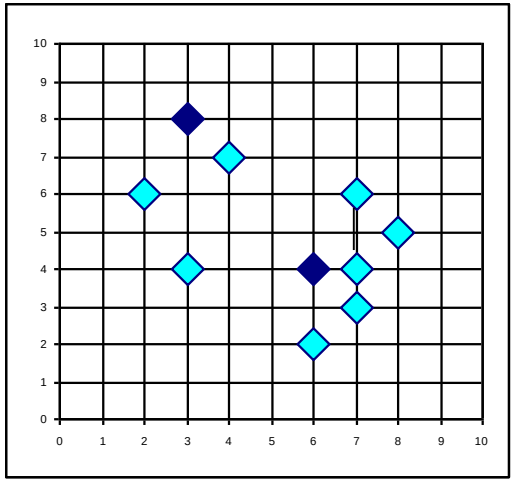
- Find representative objects, called medoids, in clusters
- PAM (Partitioning Around Medoids, 1987):
  1. Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering.
  2. PAM works effectively for small data sets, but does not scale well for large data sets

Typical k-medoids algorithm (PAM)



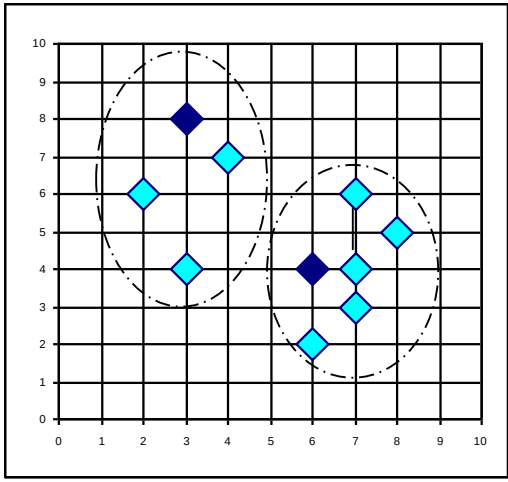
K=2

Arbitrary choose k object as initial medoids

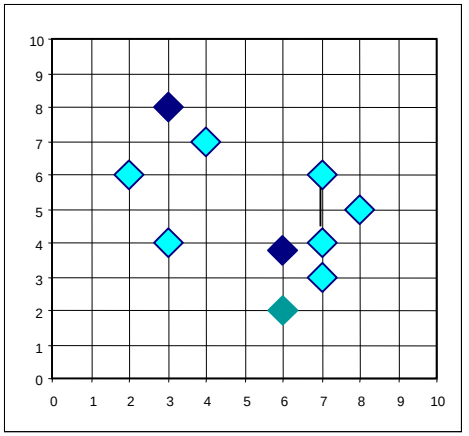


Assign each remaining object to nearest medoids

Total Cost = 20

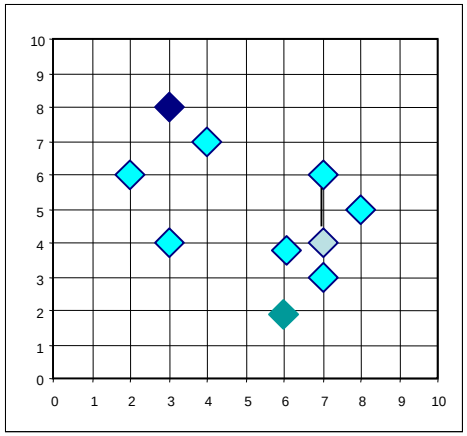


Randomly select a nonmedoid object,O_{ramdom}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

Thank you