Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Clustering High-Dimensional Data
8. Constraint-Based Clustering
9. Outlier Analysis
10. Summary

What is Cluster Analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data
  - Grouping similar data objects into clusters

Major Clustering Approaches

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, ROCK, CHAMELEON
- Density-based approach:
  - Based on some density functions
  - Typical methods: DBSACN, OPTICS

Centroid, Radius, and Diameter of a Cluster (for numerical data sets)

- Centroid: the "middle" of a cluster

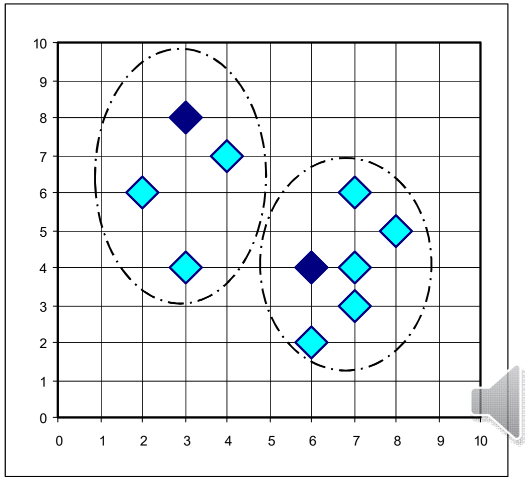
$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

- Radius: square root of an average squared distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}}$$

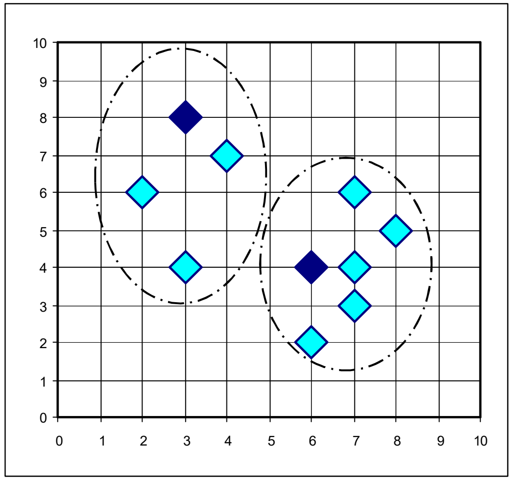
- Diameter: square root of an average squared distance between all possible pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{i=1}^{N} (t_{ip} - t_{iq})^2}{N(N-1)}}$$



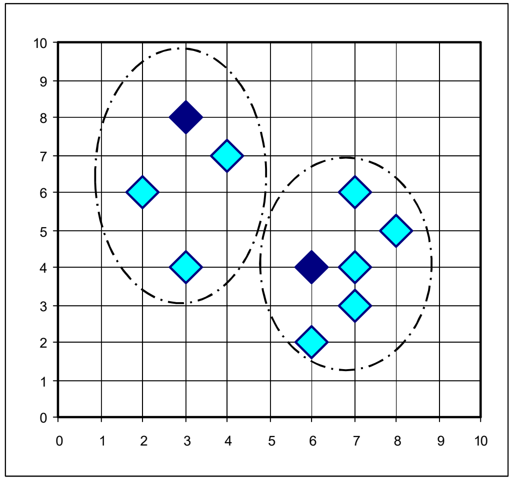
Typical Alternatives to Calculate the **Distance between Clusters**

- Single link: smallest distance between an element in one cluster and an element in the other
  - $dis(K_i, K_j) = min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other
  - $dis(K_i, K_j) = max(t_{ip}, t_{jq})$



Typical Alternatives to Calculate the **Distance between Clusters**

- Average: average distance between an element in one cluster and an element in the other
  - $dis(K_i, K_j) = avg(t_{ip}, t_{jq})$
- Centroid: distance between the centroids of two clusters
  - $dis(K_i, K_j) = dis(C_i, C_j)$
- Medoid: distance between the medoids of two clusters
  - $dis(K_i, K_j) = dis(M_i, M_j)$
  - Medoid: one chosen, centrally located (real) object in the cluster



Chapter 7. Cluster Analysis

1. What is Cluster Analysis?
2. Types of Data in Cluster Analysis
3. A Categorization of Major Clustering Methods
4. Partitioning Methods
5. Hierarchical Methods
6. Density-Based Methods
7. Clustering High-Dimensional Data
8. Constraint-Based Clustering
9. Outlier Analysis
10. Summary

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters, having the minimum sum of squared distances of objects to their representative of a cluster

$$\Sigma_{m=1}^{k}\Sigma_{t_{mi}\in Km}(C_m - t_{mi})^2$$

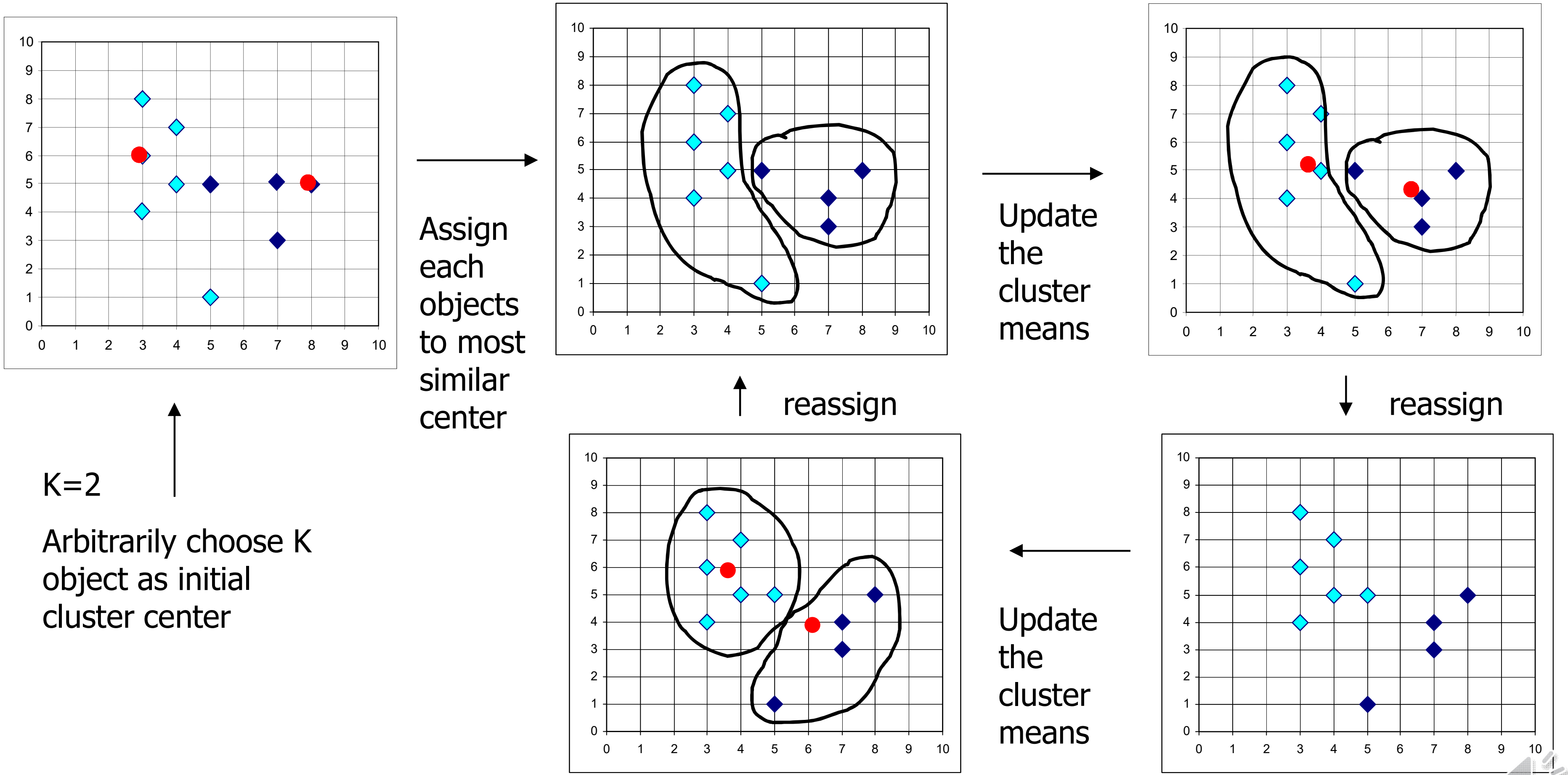
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
    - *k-means*: Each cluster is represented by the centroid of the cluster
    - *k-medoids* or PAM (Partition around medoids): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition
    - The centroid is the center, i.e., *mean point*, of the cluster
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when no more new assignment

The *K-Means* Clustering Method

- Example



Comments on the *K-Means* Method

- Strength: *Relatively efficient*: $O(n*k*t)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t << n$
    - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimum*
- Weakness
    - Applicable only when *mean* is defined (what about categorical data?)
    - Need to specify *k,* the *number* of clusters, in advance
    - Unable to handle noises and *outliers*
    - Not suitable to discover clusters with *non-convex shapes*

Variations of the *K-Means* Method

- Handling categorical data: *k-modes* (Huang'98)
  - Idea: replacing means of clusters with modes
    - X, Y: objects having m categorical attributes
    - Dissimilarity d(X,Y): the number of total mismatches

$$d(X,Y) = \Sigma_{j=1}^{m} \delta(x_j, y_j) \quad \text{where} \quad \delta(x_j, y_j) = \begin{cases} 0 \, (x_j = y_j) \\ 1 \, (x_j \neq y_j) \end{cases}$$

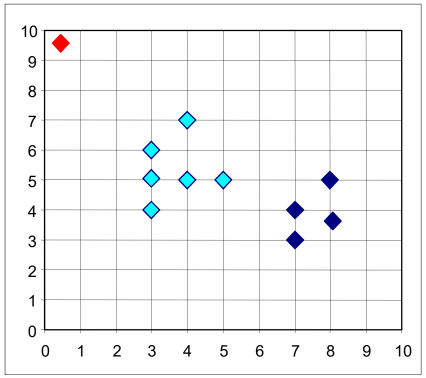
    - *Mode* of X = {X1, X2, …, Xn} is a vector Q = <q1, q2, …., qm> that minimizes

$$D(X,Q) = \Sigma_{i=1}^{n} d(X_i, Q)$$

    - Finding a mode for X
      - Taking the value most frequently occurring for each attribute
      - Using a frequency-based method to update modes of clusters
- A mixture of categorical and numerical data: *k-prototype* method

What Is the Problem of the K-Means Method?

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



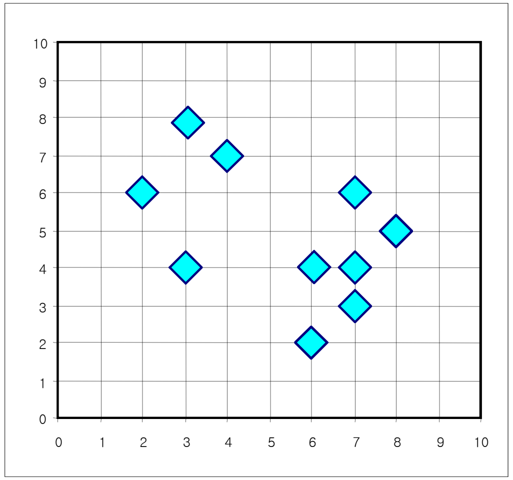
The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
  - *PAM* (Partitioning Around Medoids, 1987)
  - *CLARA* (Kaufmann & Rousseeuw, 1990)
  - *CLARANS* (Ng & Han, 1994): Randomized sampling

PAM (Partitioning Around Medoids) (1987)

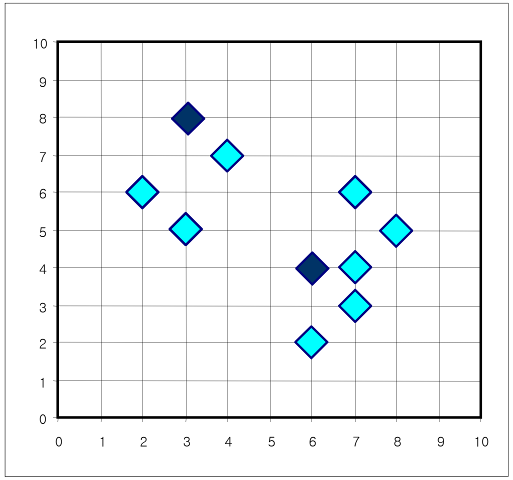
- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use a real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
  - For each pair of non-selected object ***h*** and selected object (i.e., seed) ***i***, calculate the total swapping cost TC_{ih}
  - For each pair of ***i*** and ***h***,
    - If $TC_{ih} < 0$, ***i*** is replaced by ***h***
    - Then, each non-selected object is assigned to the most similar representative object
  - Repeat steps 2-3 until there is no change

A Typical K-Medoids Algorithm (PAM)



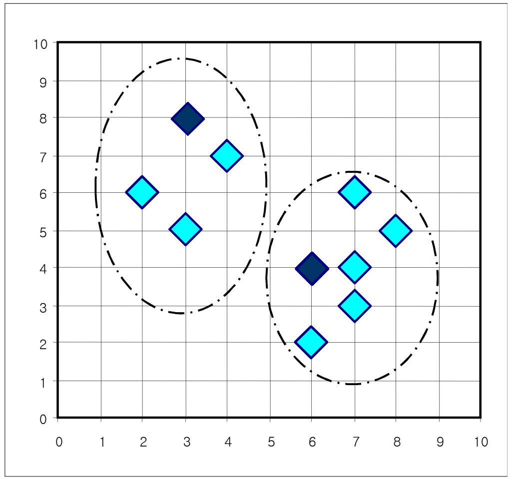
K=2

Arbitrary choose k objects as initial medoids



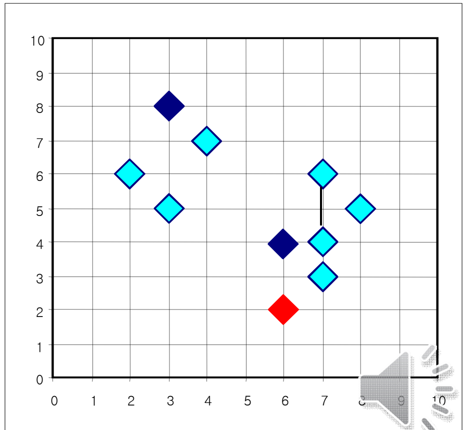
Assign each remaining object to nearest medoids

Total Cost = 20

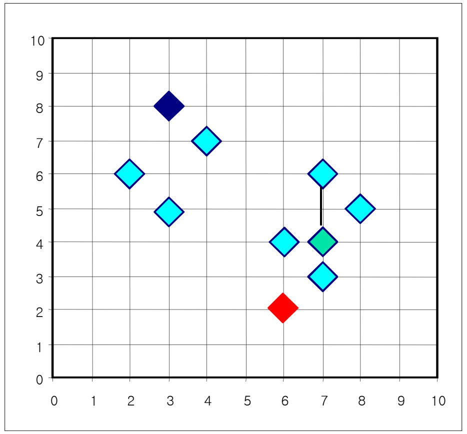


Randomly select a nonmedoid object, O_{ramdom}

Compute total cost of swapping



Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

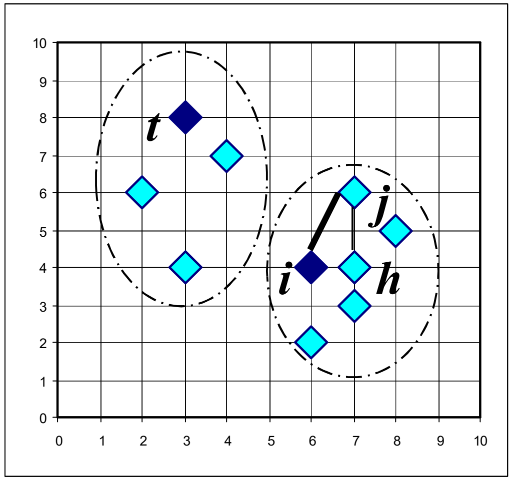
Until no change

PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$

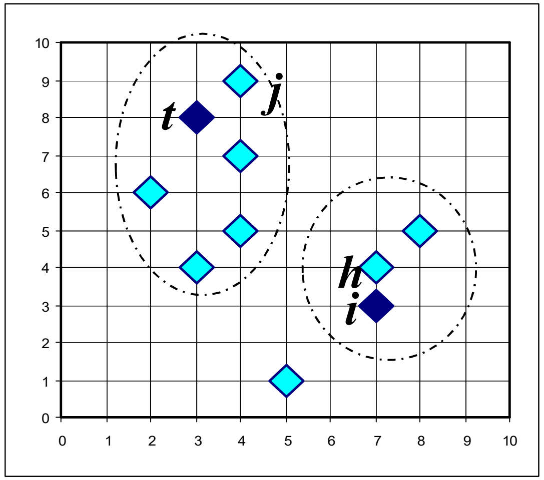
NewC - OldC

i: original seed
h: new seed
t: other seed
j: non-seed

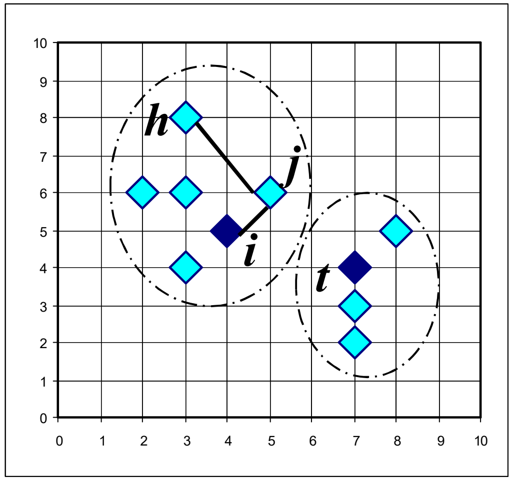
A: j belonged to i and now belongs to h
B: j belonged to t and again belongs to t
C: j belonged to i and now belongs to t
D: j belonged to t and now belongs to h



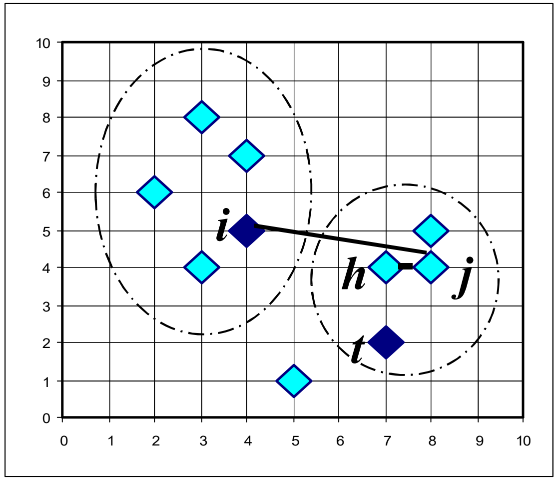
$C_{jih} = d(j, h) - d(j, i)$



$C_{jih} = 0$



$C_{jih} = d(j, t) - d(j, i)$



$C_{jih} = d(j, h) - d(j, t)$

What Is the Problem with PAM?

- PAM is more robust than k-means in the presence of noise and outliers
  - because a medoid is less influenced by outliers or other extreme values than a mean (i.e., centroid)
- PAM works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(i*k*(n-k)^2)$ where *n* is # of data, *k* is # of clusters, *i* is # of iterations

➔ Sampling based method,

CLARA (Clustering LARge Applications)

CLARA (Clustering Large Applications) (1990)

- *CLARA* (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as S+
- It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
  - Efficiency *depends on the sample size*
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set *if the sample is biased*