Unit IV : Clustering

Cluster Analysis – Partitioning Methods – Hierarchical Methods – Density Based Methods – Grid Based Methods – Outlier Analysis

Cluster Analysis

1. A Categorization of Major Clustering Methods

2. Partitioning Methods

3. Hierarchical Methods

4. Density-Based Methods

5. Grid-Based Methods

Major Clustering Approaches (I)

- <u>Partitioning approach</u>:

    - 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

- <u>Hierarchical approach</u>:

    - Create a hierarchical decomposition of the set of data (or objects) using some criterion

    - Typical methods: Diana, Agnes, BIRCH, ROCK, CAMELEON

- <u>Density-based approach</u>:

    - Based on connectivity and density functions

    - Typical methods: DBSACN, OPTICS, DenClue

Major Clustering Approaches (II)

- Grid-based approach:

  - based on a multiple-level granularity structure

  - Typical methods: STING, WaveCluster, CLIQUE

- Model-based:

  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other

  - Typical methods: EM, SOM, COBWEB

- Frequent pattern-based:

  - Based on the analysis of frequent patterns

  - Typical methods: pCluster

- User-guided or constraint-based:

  - Clustering by considering user-specified or application-specific constraints

  - Typical methods: COD (obstacles), constrained clustering

Typical Alternatives to Calculate the Distance between Clusters

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

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

- **Average:** avg distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = avg(t_{ip}, t_{jq})$

- **Centroid:** distance between the centroids of two clusters, i.e., $dis(K_i, K_j) = dis(C_i, C_j)$

- **Medoid:** distance between the medoids of two clusters, i.e., $dis(K_i, K_j) = dis(M_i, M_j)$

  - Medoid: one chosen, centrally located object in the cluster

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

- Centroid: the "middle" of a cluster

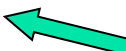$$C_m = \frac{\sum_{i=1}^{N}(t_{ip})}{N}$$

- Radius: square root of average 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 average mean squared distance between all pairs of points in the cluster

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

Partitioning Methods

1. A Categorization of Major Clustering Methods

2. Partitioning Methods

3. Hierarchical Methods

4. Density-Based Methods

5. Grid-Based Methods

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database **D** of **n** objects into a set of **k** clusters, s.t., min sum of squared distance

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

- Given a k, find a partition of $k\ clusters$ that optimizes the chosen partitioning criterion

  - Global optimal: exhaustively enumerate all partitions

  - Heuristic methods: $k\text{-}means$ and $k\text{-}medoids$ algorithms

  - $k\text{-}means$ (MacQueen'67): Each cluster is represented by the center of the cluster

  - $k\text{-}medoids$ or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster
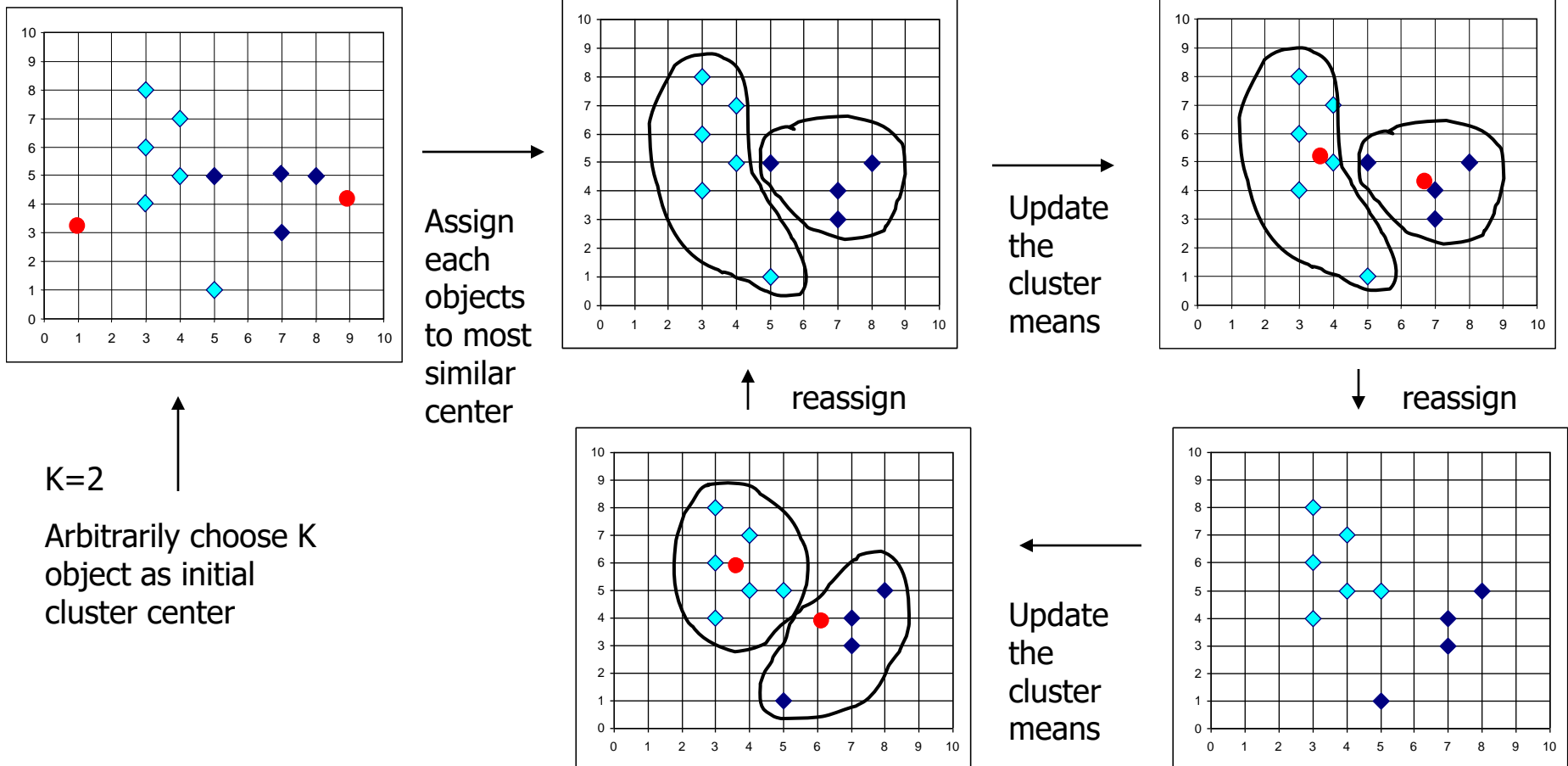
The *K-Means* Clustering Method

- 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

Assign each objects to most similar center

Update the cluster means

Update the cluster means

reassign

reassign

K=2

Arbitrarily choose K object as initial cluster center

Comments on the *K-Means* Method

- <u>Strength:</u> *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, k, $t << n$.

  - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$

- <u>Comment:</u> Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*

- <u>Weakness</u>

  - Applicable only when *mean* is defined, then what about categorical data?

  - Need to specify k, the *number* of clusters, in advance

  - Unable to handle noisy data and *outliers*

  - Not suitable to discover clusters with *non-convex shapes*

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: *k-prototype* method

What Is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to 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.
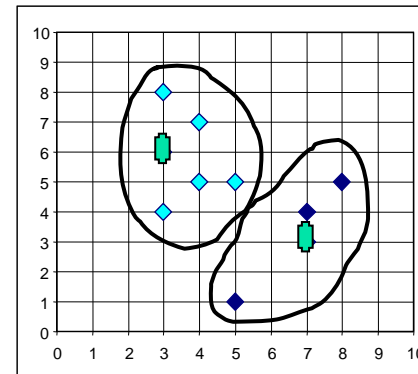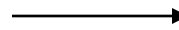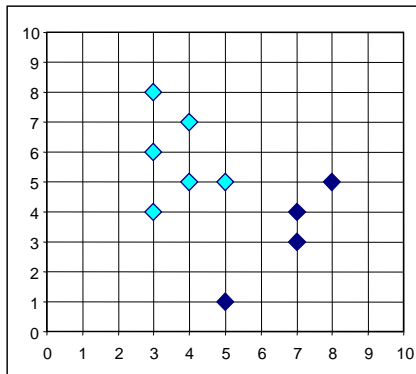
The *K-Medoids* Clustering Method

- Find *representative* objects, called <u>medoids</u>, in clusters

- *PAM* (Partitioning Around Medoids, 1987)

  - 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

  - *PAM* works effectively for small data sets, but does not scale well for large data sets

- *CLARA* (Kaufmann & Rousseeuw, 1990)

- *CLARANS* (Ng & Han, 1994): Randomized sampling

- Focusing + spatial data structure (Ester et al., 1995)
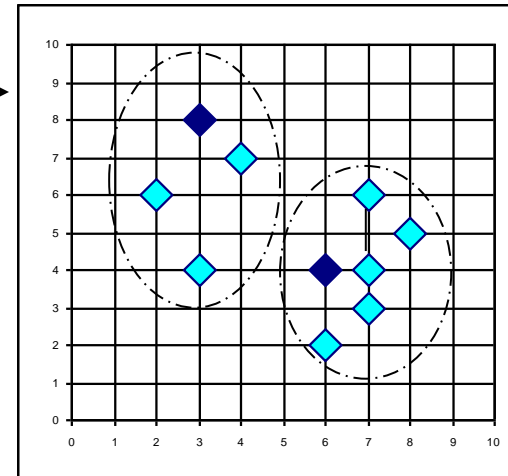
A Typical K-Medoids Algorithm (PAM)

Total Cost = 20
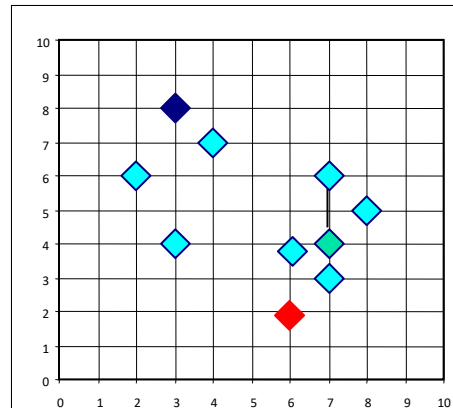

K=2
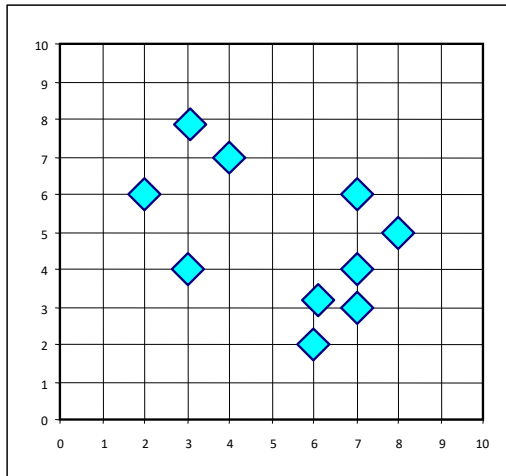
Arbitrary choose k object as initial medoids
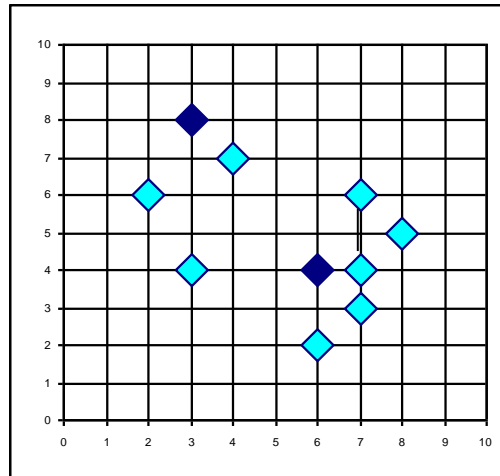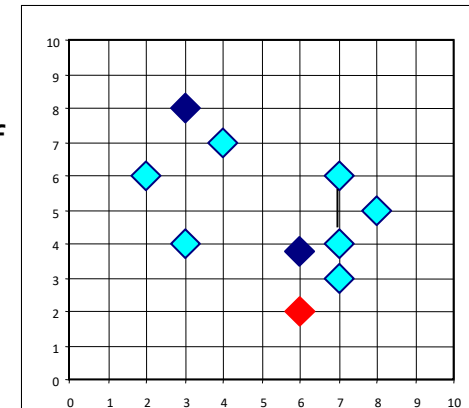
Assign each remaining object to nearest medoids

Randomly select a nonmedoid object, O_{ramdom}

Do loop

Until no change

Total Cost = 26

Swapping O and O_{ramdom}

If quality is improved.

Compute total cost of swapping

PAM (Partitioning Around Medoids) (1987)

- PAM (Kaufman and Rousseeuw, 1987), built in Splus

- Use real object to represent the cluster

  - Select k representative objects arbitrarily

  - For each pair of non-selected object h and selected object 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 assign each non-selected object to the most similar representative object

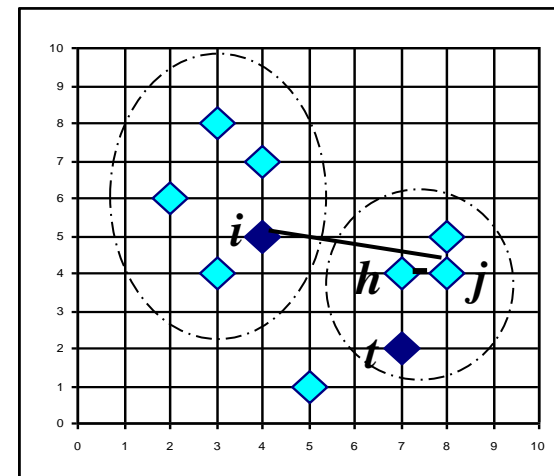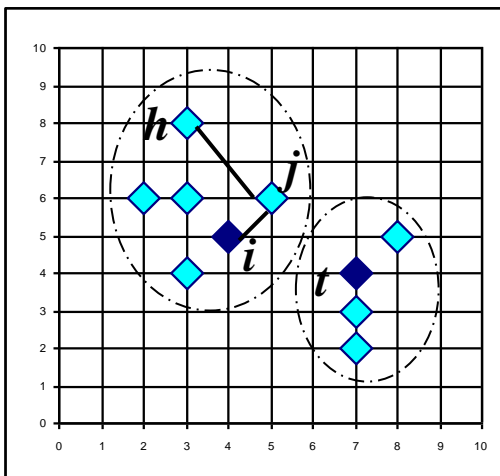 - repeat steps 2-3 until there is no change

PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$
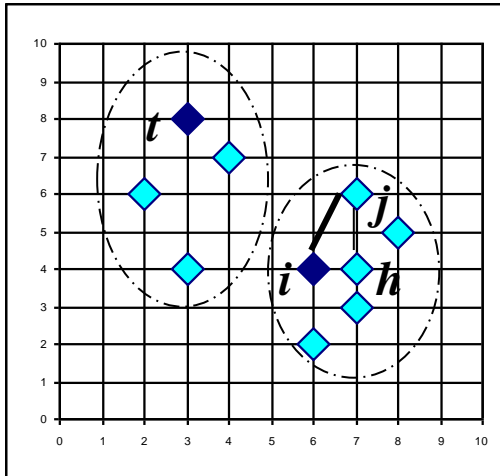
$$C_{jih} = d(j, h) - d(j, i)$$
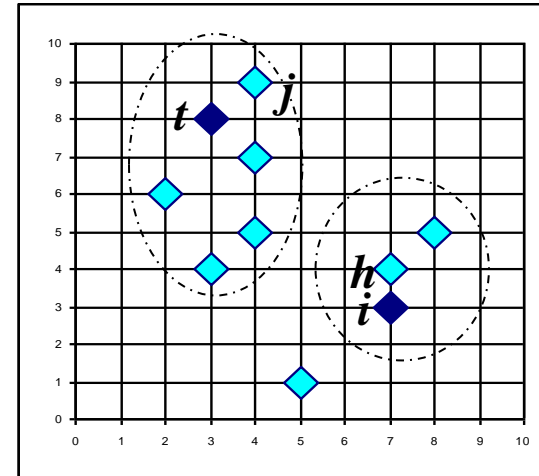
$$C_{jih} = 0$$

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

- Pam works efficiently for small data sets but does not **scale well** for large data sets.

  - $O(k(n-k)^2)$ for each iteration

    where n is # of data,k is # of clusters

➜ 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

- <u>Strength</u>: deals with larger data sets than *PAM*

- <u>Weakness</u>:

  - 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

CLARANS ("Randomized" CLARA) *(1994)*

- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)

- CLARANS draws sample of neighbors dynamically

- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids

- If the local optimum is found, *CLARANS* starts with new randomly selected node in search for a new local optimum

- It is more efficient and scalable than both *PAM* and *CLARA*

- Focusing techniques and spatial access structures may further improve its performance (Ester et al.'95)