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. Outlier Analysis
8. 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

What is Cluster Analysis?

- 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

Clustering: Rich Applications and Multidisciplinary Efforts

- Spatial Data Analysis
  - Detect spatial clusters or for other spatial mining tasks
- Economic Science (especially market research)
  - Identify customers whose behaviors are similar
- WWW
  - Cluster documents
  - Cluster Weblog data to discover groups of similar access patterns
- Image Processing & Pattern Recognition

Examples of Clustering Applications

- Marketing:
  - Help marketers discover distinct groups in their customer bases
  - Use this knowledge to develop targeted marketing programs
- Land use:
  - Identification of areas of similar land use in an earth observation database

Examples of Clustering Applications

- Insurance:
  - Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning:
  - Identifying groups of houses according to their house type, value, and geographical location

Quality: What Is Good Clustering?

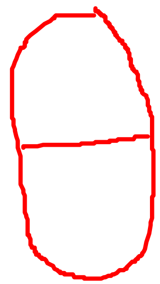
- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

Measure the Quality of Clustering

- Dissimilarity/Similarity metric: Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
- There is a separate "quality" function that measures the "goodness" of a cluster
- The definitions of distance functions
  - Usually very different for interval-scaled, Boolean, categorical, ordinal ratio, and vector variables
  - Weights should be associated with different variables based on applications and data semantics
- Hard to define "similar enough" or "good enough"
  - The answer is typically highly subjective

Requirements of Clustering in Data Mining

- Ability to deal with different types of attributes
- Ability to handle dynamic data
- Discovery of clusters with an arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noises and outliers
- Insensitive to the order of input records
- High dimensionality
- Scalability
- Incorporation of user-specified constraints



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. Outlier Analysis
8. Summary

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

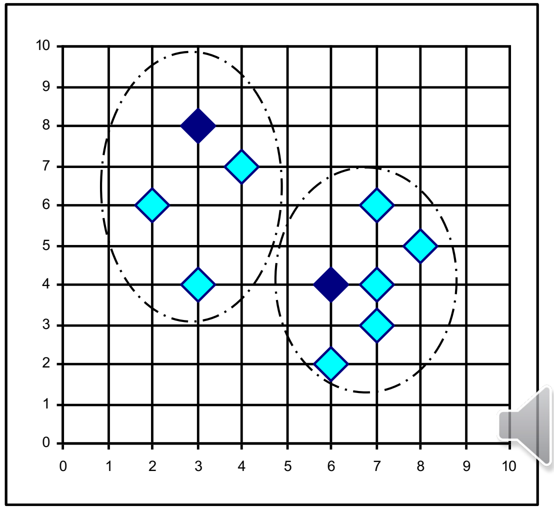
$$C_m = \frac{\Sigma_{i=1}^{N}(t_{ip})}{N}$$

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

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

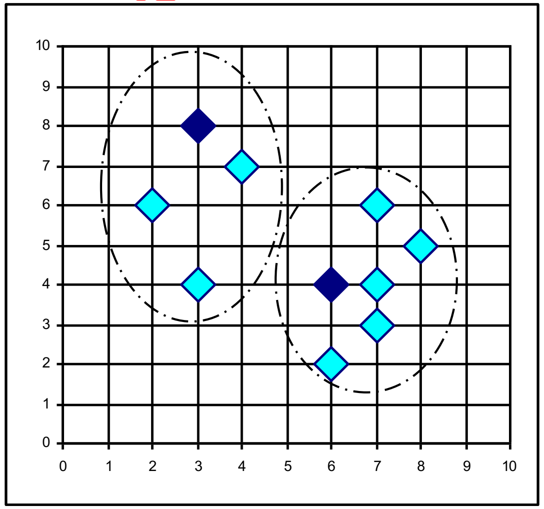
- Diameter: square root of an average squared distance between all possible pairs of points in the cluster

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



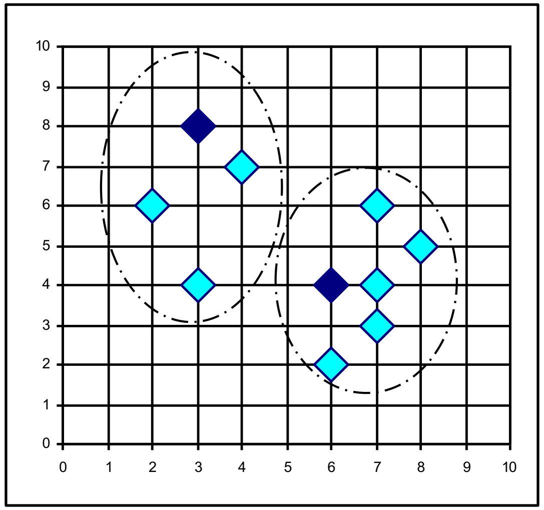
Typical Alternatives to Calculate the **Distance between Clusters**

- 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

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

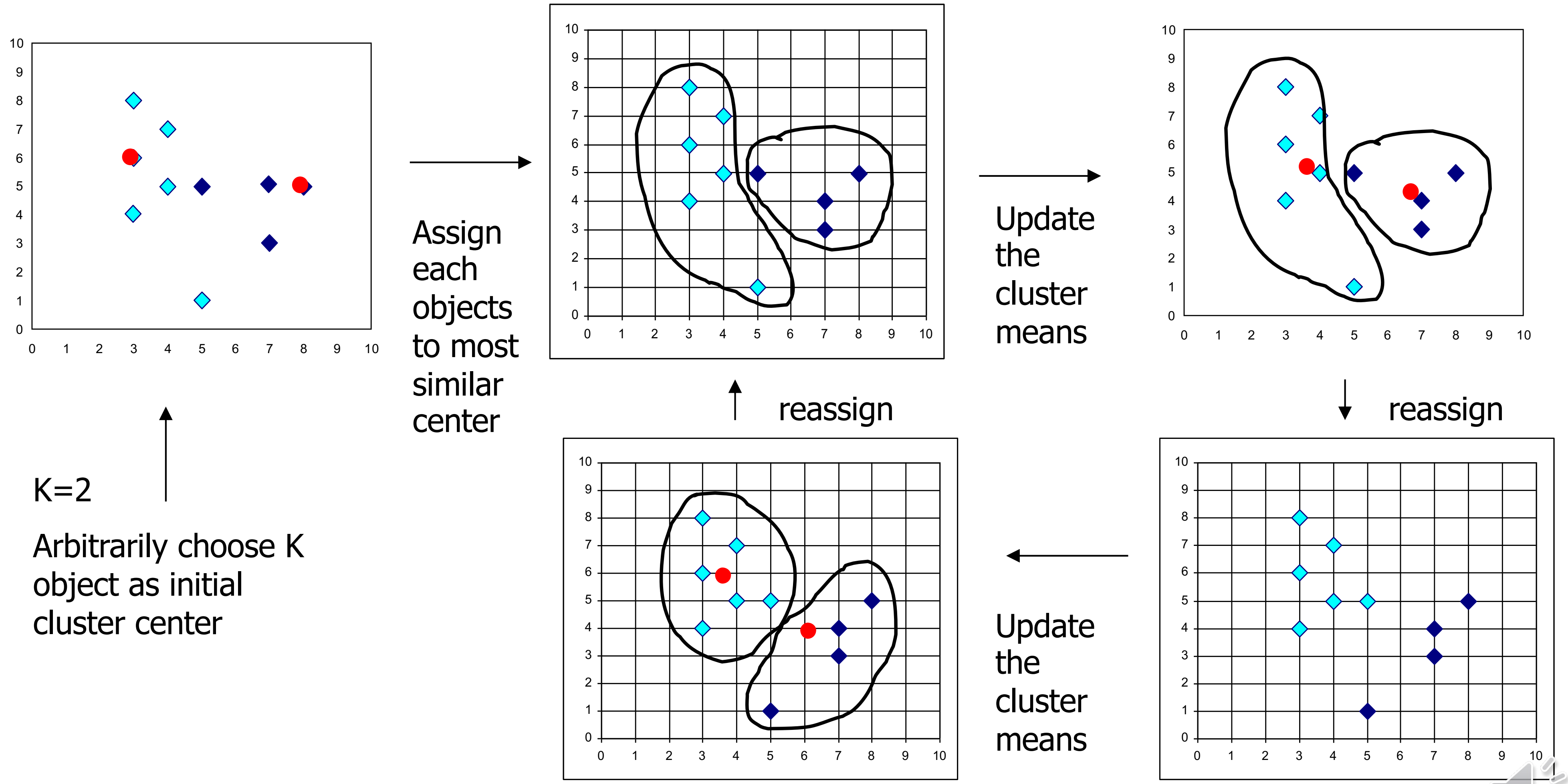
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
    - *k-means*: Each cluster is represented by the centroid of the cluster
    - *k-medoids* 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

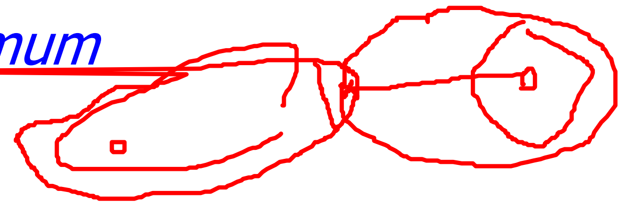
The *K-Means* Clustering Method

- 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