Chapter 7 Clustering Analysis (2)

Outline

- Cluster Analysis
- Partitioning Clustering
- Hierarchical Clustering
- Summary

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

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

- 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 center of the cluster
  - *k-medoids*: Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

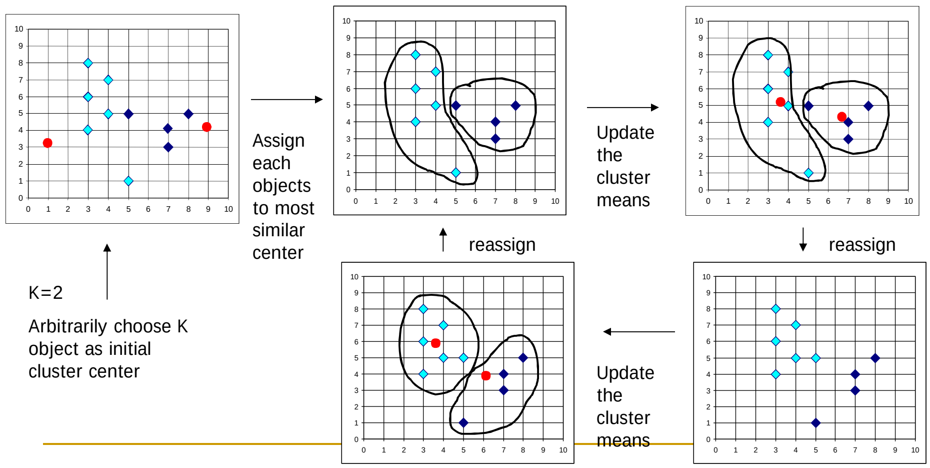
- Centroid of a cluster for numerical values: the mean value of all objects in a cluster

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

- Given k, the k-means algorithm is implemented in four steps:
  1. Select *k* seed points from D as the initial centroids.
  2. Assigning:
     - Assign each object of D to the cluster with the nearest centroid.
  3. Updating:
     - Compute centroids of the clusters of the current partition.
  4. Go back to Step 2 and continue, stop when no more new assignment.

The *K-Means* Clustering Method

- Example



Calculation of Centroids and Distance

- If cluster C_1 has three data points $d_1(x_1, y_1)$, $d_2(x_2, y_2)$, $d_3(x_3, y_3)$, the centroid of cluster C_1 $cen_1(X_1, Y_1)$ is calculated as:

$$X_1 = (x_1+x_2+x_3) / 3$$

$$Y_1 = (y_1+y_2+y_3) / 3$$

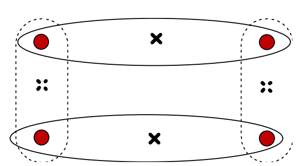
- Euclidean distance could be used to measure the distance between a data point and a centroid.

Comments on the *K-Means* Method

- Strength: *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
- Comment: Often terminates at a *local optimum*. The algorithm is very sensitive to the selection of initial centroids.
- Weakness
  - 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*

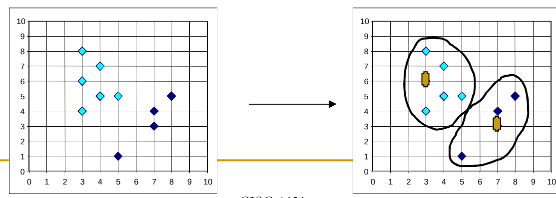
Variations of the *K-Means* Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means



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 (e.g., minimizing the sum of the dissimilarity between each object and the representative object of its cluster)
  - *PAM* works effectively for small data sets, but does not scale well for large data sets

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)

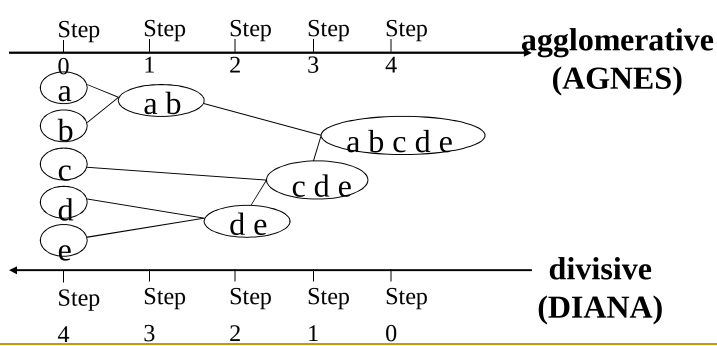
- *CLARA* (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as SPlus
  - It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
  - Efficiency 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

Outline

- Cluster Analysis
- Partitioning Clustering
- Hierarchical Clustering
- Summary

Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition

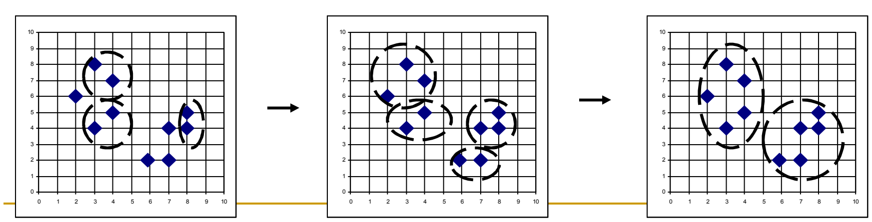


Calculation of Distance between Clusters

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

AGNES (Agglomerative Nesting)

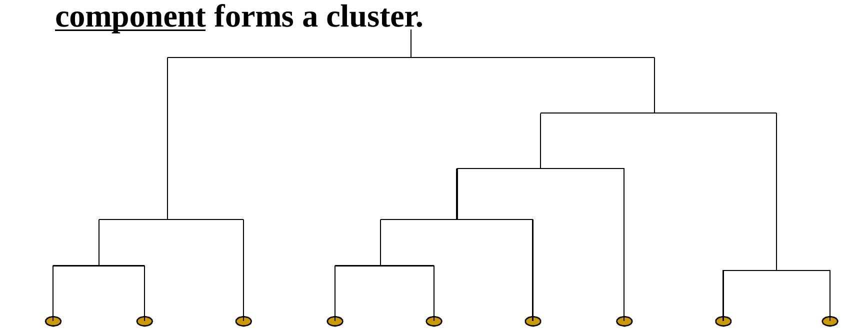
- Use the Single-Link method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



Dendrogram: Shows How the Clusters are Merged

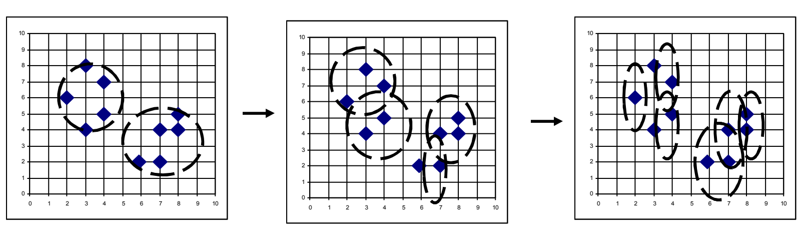
Decompose data objects into a several levels of nested partitioning (<u>tree</u> of clusters), called a <u>dendrogram</u>.

A <u>clustering</u> of the data objects is obtained by <u>cutting</u> the dendrogram at the desired level, then each <u>connected component</u> forms a cluster.



DIANA (Divisive Analysis)

- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Distance Function

- Nearest-neighbor clustering algorithm uses min. distance, $d_{min}(C_i, C_j)$ to measure.
  - **Single-linkage** algorithm terminates the process when the distance between nearest clusters exceeds a predefined threshold.
- Farthest-neighbor clustering algorithm uses max. distance, $d_{max}(C_i, C_j)$ to measure.
  - **Complete-linkage** algorithm terminates the process when the distance between nearest clusters exceeds a predefined threshold.
  - Good for true clusters which are rather compact and about same size.

Extensions to Hierarchical Clustering

- Major weakness of hierachical clustering methods
  - Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - Difficult to select the merge or split points.
  - Can never undo what was done previously
- Integration of hierarchical & partitioning clustering
  - Bisecting k-means algorithm
  - CHAMELEON: Hierarchical Clustering Using Dynamic Modeling

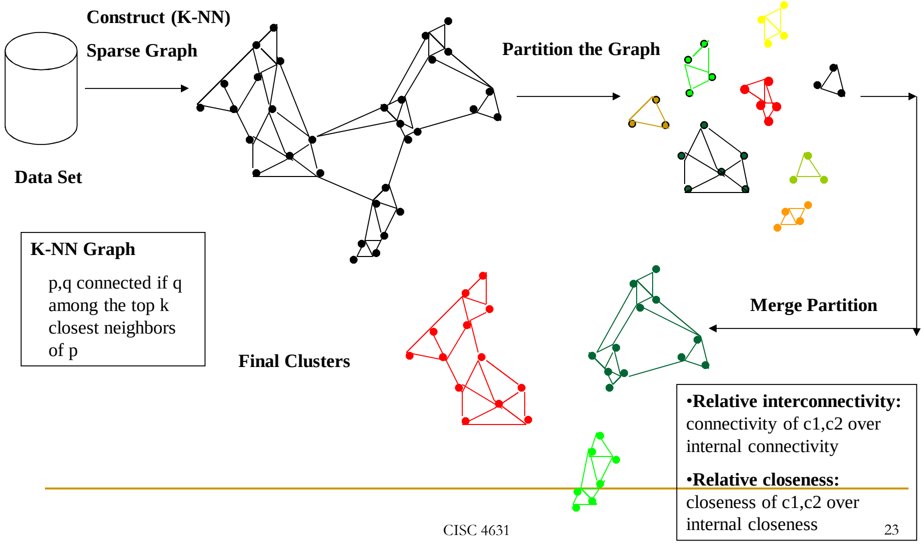
Bisecting K-means Algorithm

- Given *k*, the bisecting *k-means* algorithm is implemented in four steps:
  1. Take the database D as a cluster.
  2. Select a cluster to split.
  3. Perform k-means algorithm on the selected cluster with k=2.
     a. Select **2** seed points from as the initial centroids.
     b. Assigning:
        a. Assign each object within the selected cluster to the cluster with the nearest centroid.
     c. Updating:
        a. Compute centroids of these 2 clusters of the current partition.
     d. Go back to step b and continue. Stop when no more new assignment.
  4. Go back to Step 2 and continue, stop when there are k clusters.

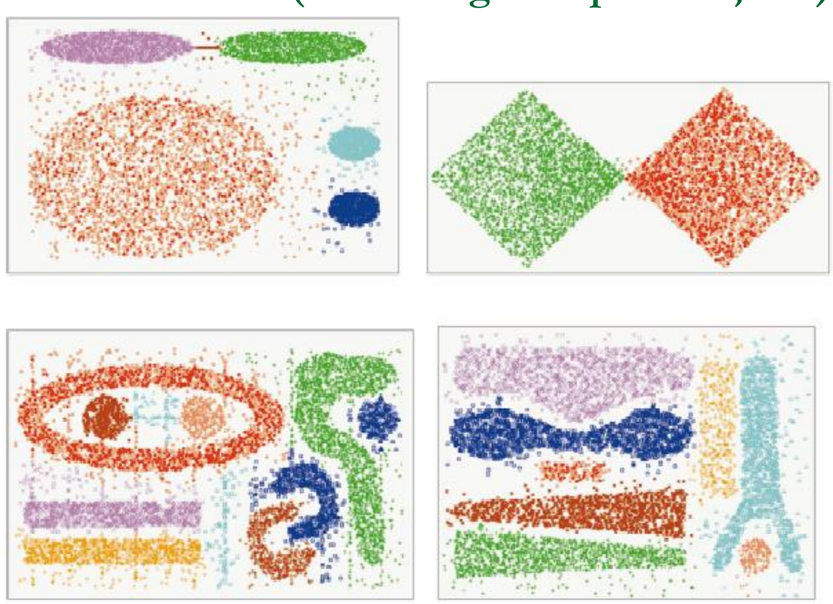
CHAMELEON: Hierarchical Clustering Using Dynamic Modeling

- Measures the similarity based on a dynamic model
  - Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters
- A two-phase algorithm
  1. Use a graph partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
  2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters

Overall Framework of CHAMELEON



CHAMELEON (Clustering Complex Objects)



Outline

- Cluster Analysis
- Partitioning Clustering
- Hierarchical Clustering
- Summary

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high intra-class similarity: cohesive within clusters
  - low inter-class similarity: distinctive between clusters
- 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 of Clustering Accuracy

- Measured by manually labeled data
  - We manually assign tuples into clusters according to their properties (e.g., professors in different research areas)
  - Precision, Recall and F-measure

Precision, Recall and F-measure

- If n_i is the number of the members of class i, n_j is the number of the members of cluster j, and n_{ij} is the number of the members of class i in cluster j, then P(i, j) (precision) and R(i, j) (recall) can be defined as

$$P(i,j) = {n_{ij}}/{n_j} \qquad R(i,j) = {n_{ij}}/{n_i}$$

- F-measure is defined as

$$F-measure(i,j) = {2P(i,j)*R(i,j)}/{P(i,j)+R(i,j)}$$

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- There are still lots of research issues on cluster analysis