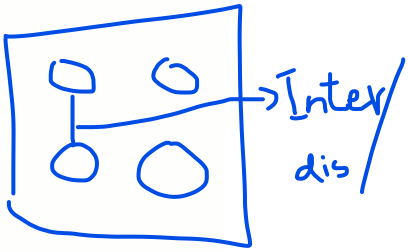
CLUSTER ANALYSIS

- Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups
- Cluster analysis (or *clustering*, *data segmentation*, ...)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Applications

- **Biology**: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- **Information retrieval:** document clustering
- **Marketing:** Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- **City-planning:** Identifying groups of houses according to their house type, value, and geographical location
- **Earth-quake studies:** Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- **Economic Science:** market research
- **Preprocessing Tool:** Summarization, Compression, Outlier detection

- 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 method depends on
  - the similarity measure used by the method
  - its implementation, and
  - Its ability to discover some or all of the hidden patterns
- Similarity is expressed in terms of a distance function
  - The definitions of distance functions are usually rather different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables



Considerations for Cluster Analysis

- Partitioning criteria
  - Single level vs. hierarchical partitioning (often, multi-level hierarchical partitioning is desirable)
- Separation of clusters
  - Exclusive (e.g., one customer belongs to only one region) vs. non-exclusive (e.g., one document may belong to more than one class)
- Similarity measure
  - Distance-based (e.g., Euclidian, road network, vector) vs. connectivity-based (e.g., density or contiguity)
- Clustering space
  - Full space (often when low dimensional) vs. subspaces (often in high-dimensional clustering)

Requirements and Challenges

- Scalability
  - Clustering all the data instead of only on samples
- Ability to deal with different types of attributes
  - Numerical, binary, categorical, ordinal, linked, and mixture of these
- Constraint-based clustering
  - User may give inputs on constraints
  - Use domain knowledge to determine input parameters
- Interpretability and usability
- Ability to deal with noisy data

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, CAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSACN, OPTICS, DenClue
- others

cluster analysis.mp4

Classification of Clustering Approaches

- Hierarchical vs partitional
- Exclusive vs overlapping vs fuzzy
- Complete vs partial

27:04 / 1:38:44

Partitioning Algorithms: Basic Concept

- Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

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

- Given *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* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

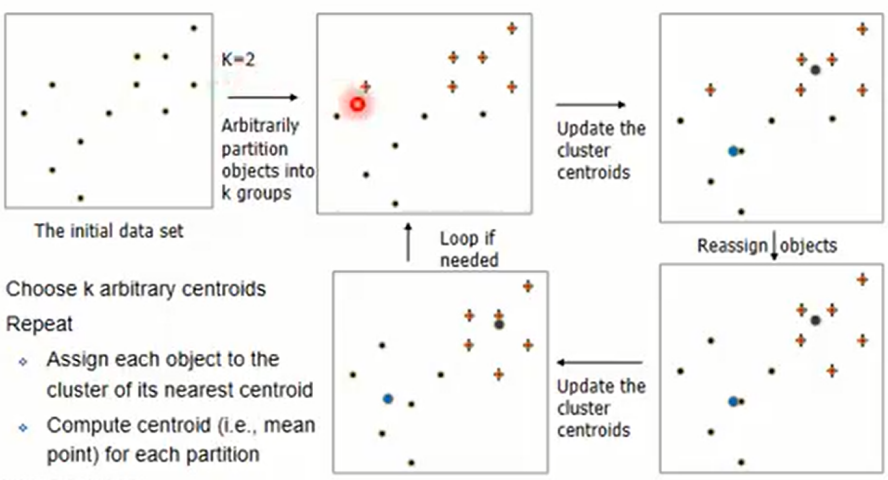
==========

K-MEANS

1. arbitrarily choose *k* objects from *D* as the initial cluster centers;
2. **repeat**
   a) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
   b) update the cluster means, that is, calculate the mean value of the objects for each cluster;

   while no change;

An Example of *K-Means* Clustering



- Choose k arbitrary centroids
- Repeat
  - Assign each object to the cluster of its nearest centroid
  - Compute centroid (i.e., mean point) for each partition
- Until no change

Working of the K-Mean Algorithm

Instance #:	1	2	3	4	5	6
X:	1	1	2	2	3	3
Y:	1.5	4.5	1.5	3.5	2.5	6.0

- Let's pick Instances #1 and #3 as the initial centroids.

	Distance with Centroid1	Distance with Centroid2
• Instance #2	3.00	3.16
• Instance #4	2.24	2.00
• Instance #5	2.24	1.41
• Instance #6	6.02	5.41

- **New centroids are (1, 3) and (2.5, 3.4)**

cluster analysis.mp4

- The *k*-means method is not guaranteed to converge to the global optimum and often terminates at a local optimum.
- Complexity of the *k*-means is $O(nkt)$, where n is #of objects, k is # of clusters, and t is # of iterations. Normally, $k<n$ and $t<n$. Therefore, the method is relatively scalable and efficient in processing large data sets.
- Variants of *k*-means: selection of the initial *k*-means, dissimilarity metric, and the strategies for calculating cluster means.
- The *k*-means method can be applied only when the mean of a set of objects is defined. When data with nominal attributes are involved, the ***k*-modes method** is a variant of *k*-means.
- The *k*-means and the *k*-modes methods are integrated for mixed attributes
- The necessity to specify *k* in advance is a disadvantage.
- it is sensitive to noise and outlier data

51:03 / 1:38:44

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
- **Partitioning AroundMedoids (PAM)** algorithm is a popular realization of *k*-medoids clustering

Algorithm

1. arbitrarily choose *k* objects in *D* as the initial representative objects;
2. **repeat**
   a) assign each remaining object to the cluster with the nearest representative object;
   b) randomly select a non-representative object, o_{random};
   c) compute the total cost, *S*, of swapping representative object, o_j, with o_{random};
   d) **if** $S < 0$ **then** swap o_j with o_{random} to form the new set of *k* representative objects;

   until no change;

Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.

✓ External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
  - Entropy

✓ Internal Index: Used to measure the goodness of a clustering structure without respect to external information.
  - Sum of Squared Error (SSE)

✓ Relative Index: Used to compare two different clusterings or clusters.
  - Often an external or internal index is used for this function, e.g., SSE or entropy

- Cluster Cohesion:
  - ❖Measures how closely related are objects in a cluster
  - ❖Cohesion is measured by the within cluster sum of squares (SSE)

$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

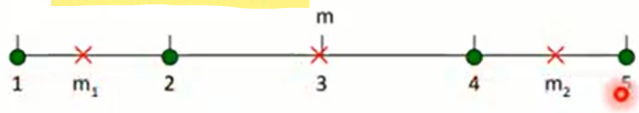
- Cluster Separation:
  - ❖Measure how distinct or well-separated a cluster is from other clusters
  - ❖Separation is measured by the between cluster sum of squares

$$BSS = \sum_i |C_i| (m - m_i)^2$$

Where $|C_i|$ is the size of cluster i

Cohesion and Separation: Example

- Example:
  - BSS + WSS = constant



K=1 cluster:

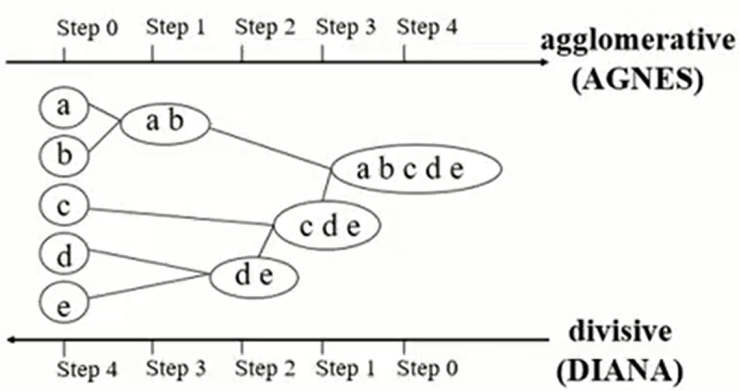
$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$
$$BSS = 4 \times (3-3)^2 = 0$$
$$Total = 10 + 0 = 10$$

K=2 clusters:

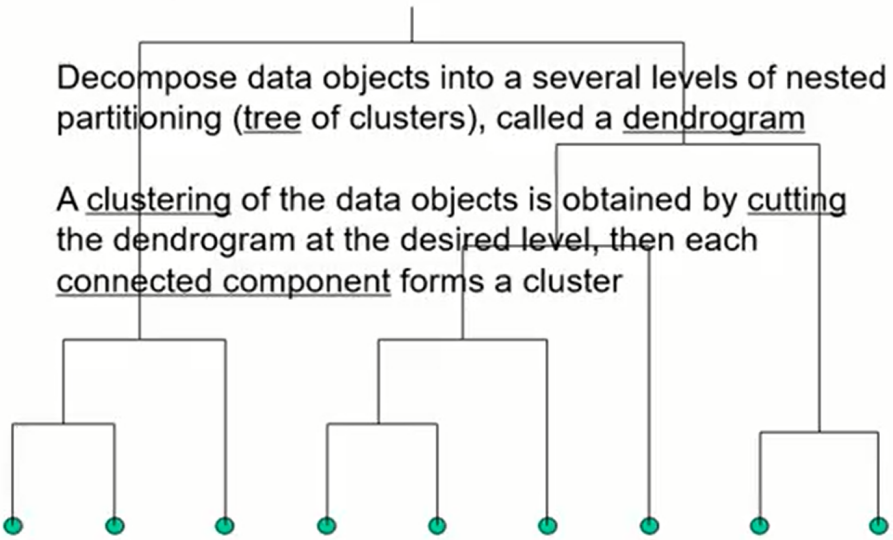
$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$
$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$
$$Total = 1 + 9 = 10$$

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



Dendrogram: Shows How Clusters are Merged



Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

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

Two main types of hierarchical clustering :

- Agglomerative:
  - ✓ Start with the points as individual clusters
  - ✓ At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
- Divisive:
  - ✓ Start with one, all-inclusive cluster
  - ✓ At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - ✓ Merge or split one cluster at a time

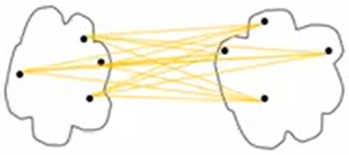
$\Rightarrow$

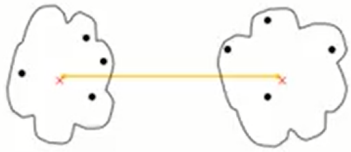
Calculation of Proximity/Similarity between Clusters

- Group Average:

  The proximity of the clusters is defined as the average pairwise proximity among of all pairs of points in the clusters.

Dist({3,6,4},{2,5})={dist(3,2)+dist(3,5)+dist(6,2)+dist(6,5)+dist(4,2)+dist(4,5)} / 3*2



Calculation of Proximity/Similarity between Clusters

- Distance between the Centroids