COMP 345: Data Mining
Clustering Basics

Slides Adapted From : Jiawei Han, Micheline Kamber & Jian Pei
Data Mining: Concepts and Techniques, 3rd ed.

Rhodes College

What is 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

2

1

Applications of Cluster Analysis

- Data reduction
  - Summarization: Preprocessing for regression, PCA, classification, and association analysis
  - Compression: Image processing: vector quantization
- Hypothesis generation and testing
- Prediction based on groups
  - Cluster & find characteristics/patterns for each group
- Finding K-nearest Neighbors
  - Localizing search to one or a small number of clusters
- Outlier detection: Outliers are often viewed as those "far away" from any cluster

3

Basic Steps to Develop a Clustering Task

- Feature selection
  - Select info concerning the task of interest
  - Minimal information redundancy
- Proximity measure
  - Similarity of two feature vectors
- Clustering criterion
  - Expressed via a cost function or some rules
- Clustering algorithms
  - Choice of algorithms
- Validation of the results
  - Validation test (also, *clustering tendency* test)
- Interpretation of the results
  - Integration with applications

4

Quality: What Is Good Clustering?

- A <u>good clustering</u> method will produce high quality clusters
  - high <u>intra-class</u> similarity: cohesive within clusters
  - low <u>inter-class</u> similarity: distinctive between clusters
- The <u>quality</u> 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 <u>hidden</u> patterns

5

Measure the Quality of Clustering

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

6

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)

7

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
- Others
  - Discovery of clusters with arbitrary shape
  - Ability to deal with noisy data
  - Incremental clustering and insensitivity to input order
  - High dimensionality

8

Major Clustering Approaches (I)

- 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, CHAMELEON
- Density-based approach:
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue
- Grid-based approach:
  - based on a multiple-level granularity structure
  - Typical methods: STING, WaveCluster, CLIQUE

9

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)

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

- 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
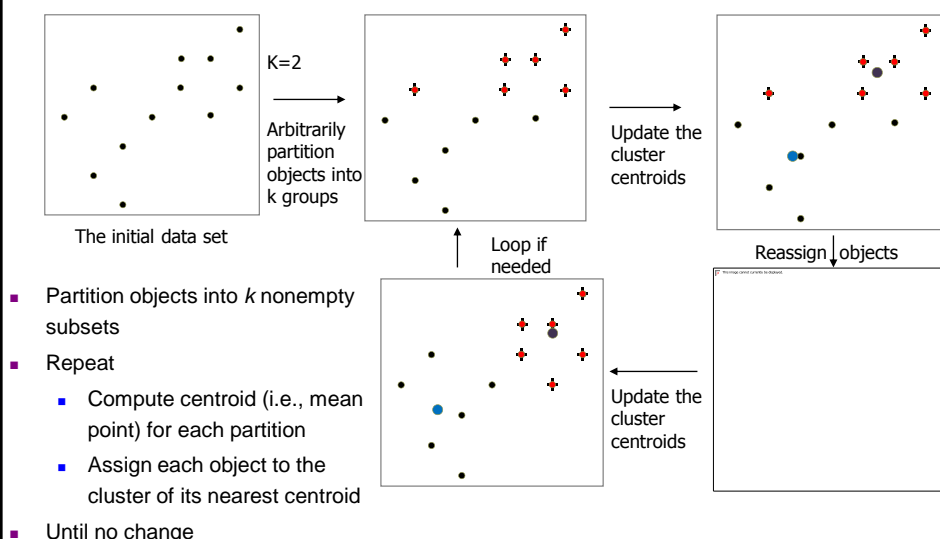
10

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 partitioning (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 the assignment does not change

11

An Example of *K-Means* Clustering

The initial data set

K=2

Arbitrarily partition objects into k groups

Update the cluster centroids

Reassign objects

Loop if needed

Update the cluster centroids

- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

12

Comments on the *K-Means* Method

- Strength: *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: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimal*
- Weakness
  - Applicable only to objects in a continuous n-dimensional space
    - Using the k-modes method for categorical data
    - In comparison, k-medoids can be applied to a wide range of data
  - Need to specify *k,* the *number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
  - Sensitive to noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

13

Example Using K-means

Suppose that the data mining task is to cluster points (with (x, y) representing location) into 3 clusters, where the points are:

A_1 (2, 10), A_2 (2, 5), A_3 (8, 4), B_1 (5, 8), B_2 (7, 5), B_3 (6, 4), C_1 (1, 2), C_2 (4, 9)

The distance function is Euclidean distance. Suppose initially we assign A_1, B_1 and C_1 as the center of each cluster, respectively.

Use the k-means algorithm to show the 3 cluster centers after the first round of execution.

14

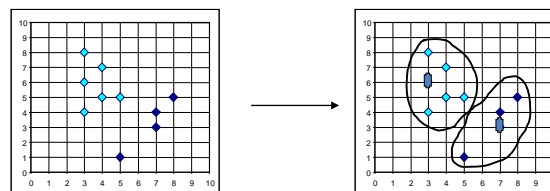# Variations of the *K-Means* Method

- Most of the variants of the *k-means* differ in

  - Selection of the initial *k* means

  - Dissimilarity calculations

  - Strategies to calculate cluster means

- Handling categorical data: *k-modes*

  - 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
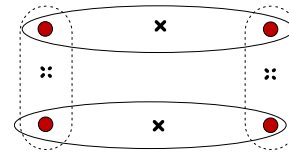
15

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

16

PAM: A Typical K-Medoids Algorithm

Total Cost = 20

K=2

Arbitrary choose k object as initial medoids

Assign each remaining object to nearest medoids

Do loop

Until no change

Total Cost = 26

Swapping O and O_ramdom

If quality is improved.

Compute total cost of swapping

Randomly select a nonmedoid object, O_{ramdom}

17
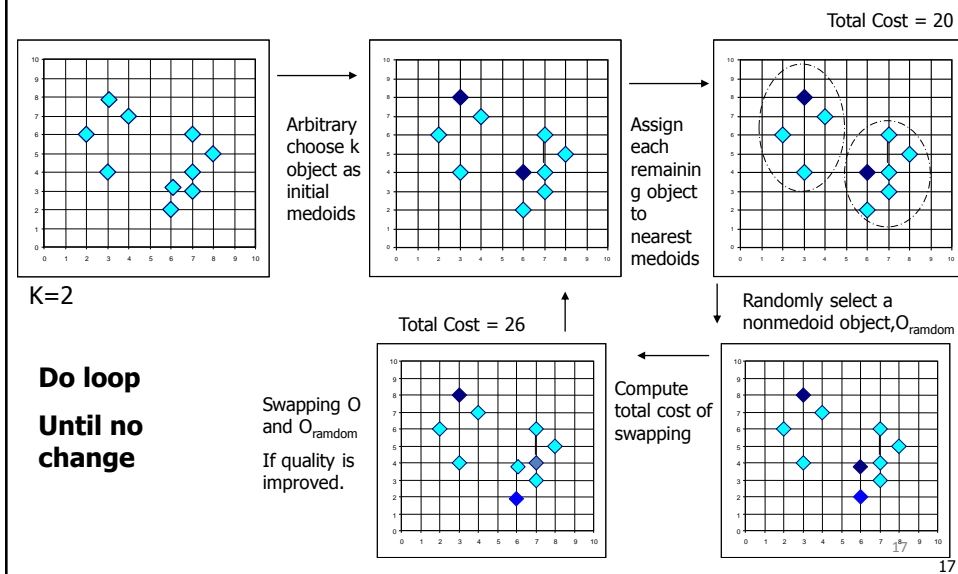
The K-Medoid Clustering Method

- *K-Medoids* Clustering: Find *representative* objects (<u>medoids</u>) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 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 (due to the computational complexity)
- Efficiency improvement on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

18

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
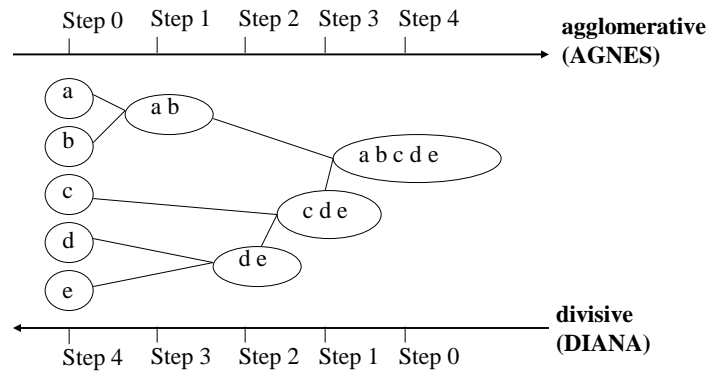
19

AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- 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
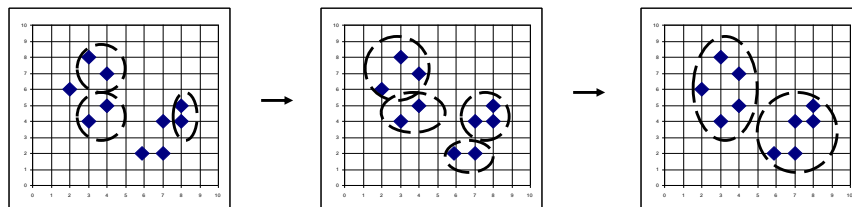
20

Dendrogram: **Shows How Clusters are Merged**

Decompose data objects into 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
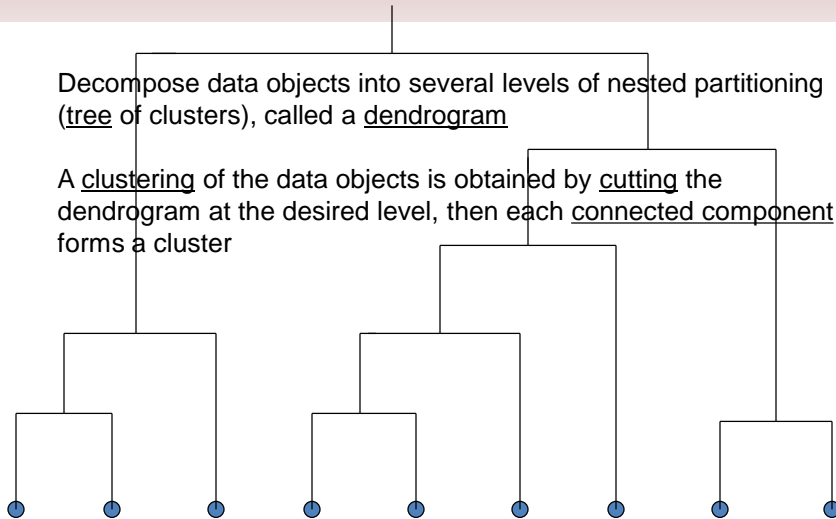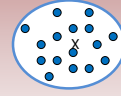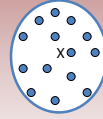
21

DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)

- Implemented in statistical analysis packages, e.g., Splus

- Inverse order of AGNES

- Eventually each node forms a cluster on its own
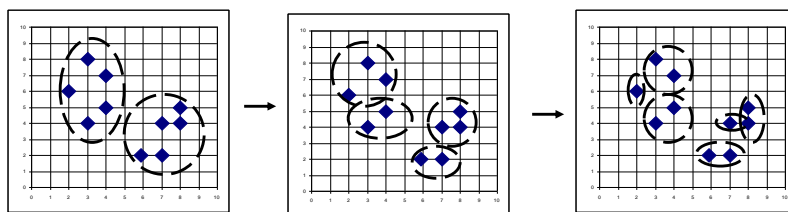

22

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})
- Centroid: distance between the centroids of two clusters, i.e., dist(K_i, K_j) = dist(C_i, C_j)
- Medoid: distance between the medoids of two clusters, i.e., dist(K_i, K_j) = dist(M_i, M_j)
  - Medoid: a chosen, centrally located object in the cluster

23

23

Centroid, Radius and Diameter of a Cluster (for numerical data sets)
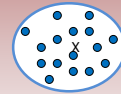
- Centroid: the "middle" of a cluster

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

- 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

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

24

24

Extensions to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
  - Can never undo what was done previously
  - Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
- Integration of hierarchical & distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
 - CHAMELEON (1999): hierarchical clustering using dynamic modeling

25

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

- Zhang, Ramakrishnan & Livny, SIGMOD'96
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness:* handles only numeric data, and sensitive to the order of the data record

26

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling (1999)
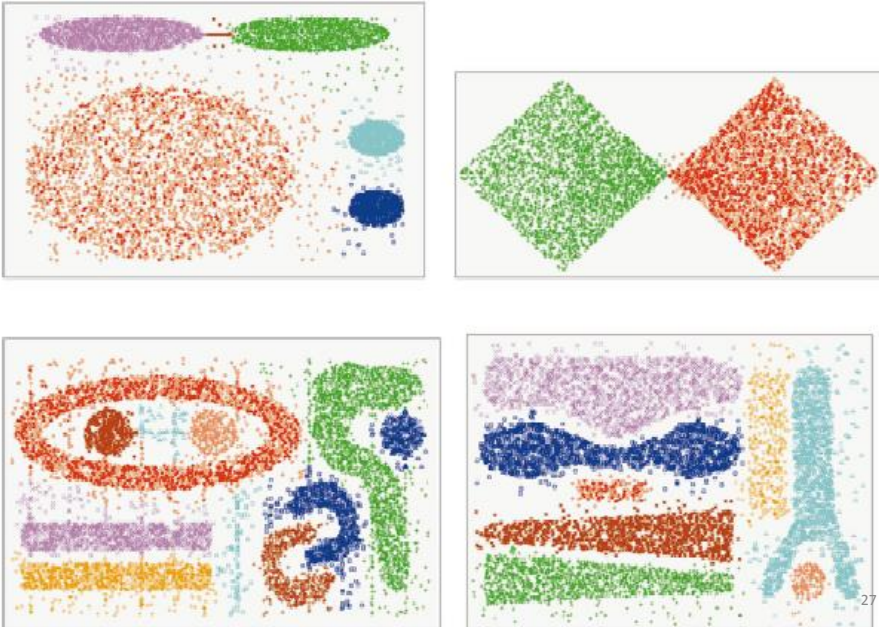
Summary

- 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
- K-means and K-medoids algorithms are popular partitioning-based clustering algorithms
- Birch and Chameleon are interesting hierarchical clustering algorithms, and there are also probabilistic hierarchical clustering algorithms