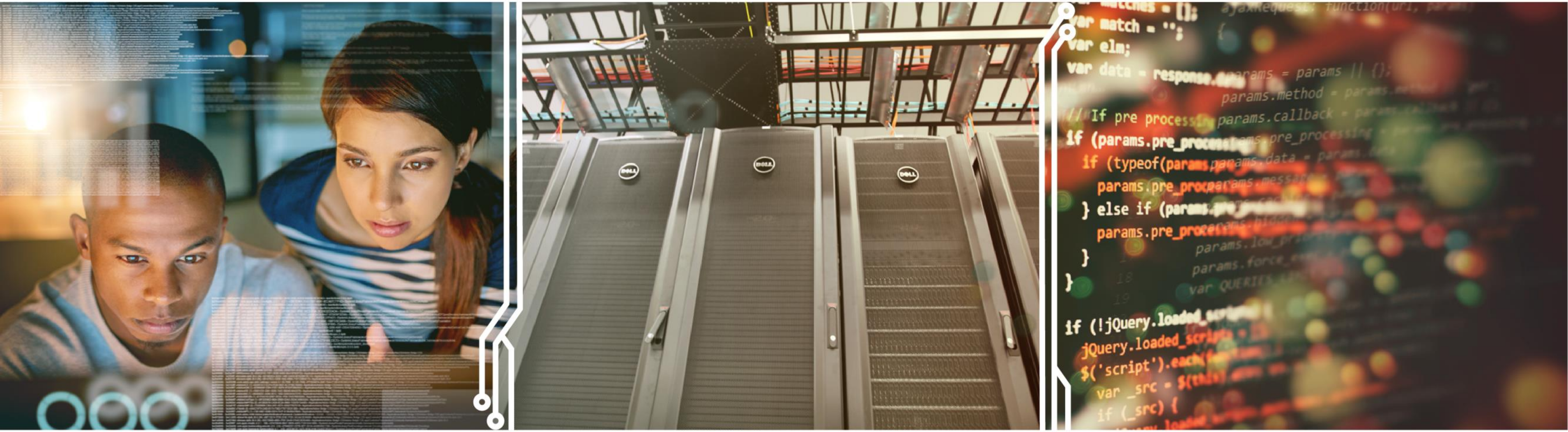
Clustering



Dr. Christan Grant
Dr. Laura Melissa Cruz Castro
CAP5771 – Introduction to Data Science
University of Florida

Outline

- Basic Concepts of Cluster Analysis
- Partitioning Methods
- Hierarchical Methods
- Density methods
- Density-based and grid-based methods

What is Cluster Analysis?

- What is a cluster?
  - A cluster is a collection of data objects which are
    - Similar (or related) to one another within the same group (i.e., cluster)
    - Dissimilar (or unrelated) to the objects in other groups (i.e., clusters)
- Cluster analysis (or clustering, data segmentation, …)
  - Given a set of data points, partition them into a set of groups (i.e., clusters) which are as similar as possible
- Cluster analysis is unsupervised learning (i.e., no predefined classes)
  - This contrasts with classification (i.e., supervised learning)
- Typical ways to use/apply cluster analysis
  - As a stand-alone tool to get insight into data distribution, or
  - As a preprocessing (or intermediate) step for other algorithms

Broad Applications of Cluster Analysis

- A key intermediate step for other data mining tasks
  - Generating a compact summary of data for classification, pattern discovery, hypothesis generation and testing, etc.
  - Outlier detection: Outliers—those “far away” from any cluster
- Data summarization, compression, and reduction
  - Ex. Image processing: Vector quantization
- Collaborative filtering, recommendation systems, or customer segmentation
  - Find like-minded users or similar products
- Dynamic trend detection
  - Clustering stream data and detecting trends and patterns
- Multimedia data analysis, biological data analysis and social network analysis
  - Ex. Clustering images or video/audio clips, gene/protein sequences, etc.

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 method depends on

- The similarity measure used by the method
- Its implementation, and
- 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)

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

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

Others

Discovery of clusters with arbitrary shape

Ability to deal with noisy data

Incremental clustering and insensitivity to input order

High dimensionality

Typical Clustering Methodologies

- Distance-based methods
  - Partitioning algorithms: K-Means, K-Medians, K-Medoids
  - Hierarchical algorithms: Agglomerative vs. divisive methods
- Density-based and grid-based methods
  - Density-based: Data space is explored at a high-level of granularity and then post-processing to put together dense regions into an arbitrary shape
  - Grid-based: Individual regions of the data space are formed into a grid-like structure
- Probabilistic and generative models: Modeling data from a generative process
  - Assume a specific form of the generative model (e.g., mixture of Gaussians)
  - Model parameters are estimated with the Expectation-Maximization (EM) algorithm (using the available dataset, for a maximum likelihood fit)
  - Then estimate the generative probability of the underlying data points
- High-dimensional clustering

Outline

- Basic Concepts of Cluster Analysis
- Partitioning Methods
- Hierarchical Methods
- Density methods
- Density-based and grid-based methods

Partitioning Algorithms: Basic Concept

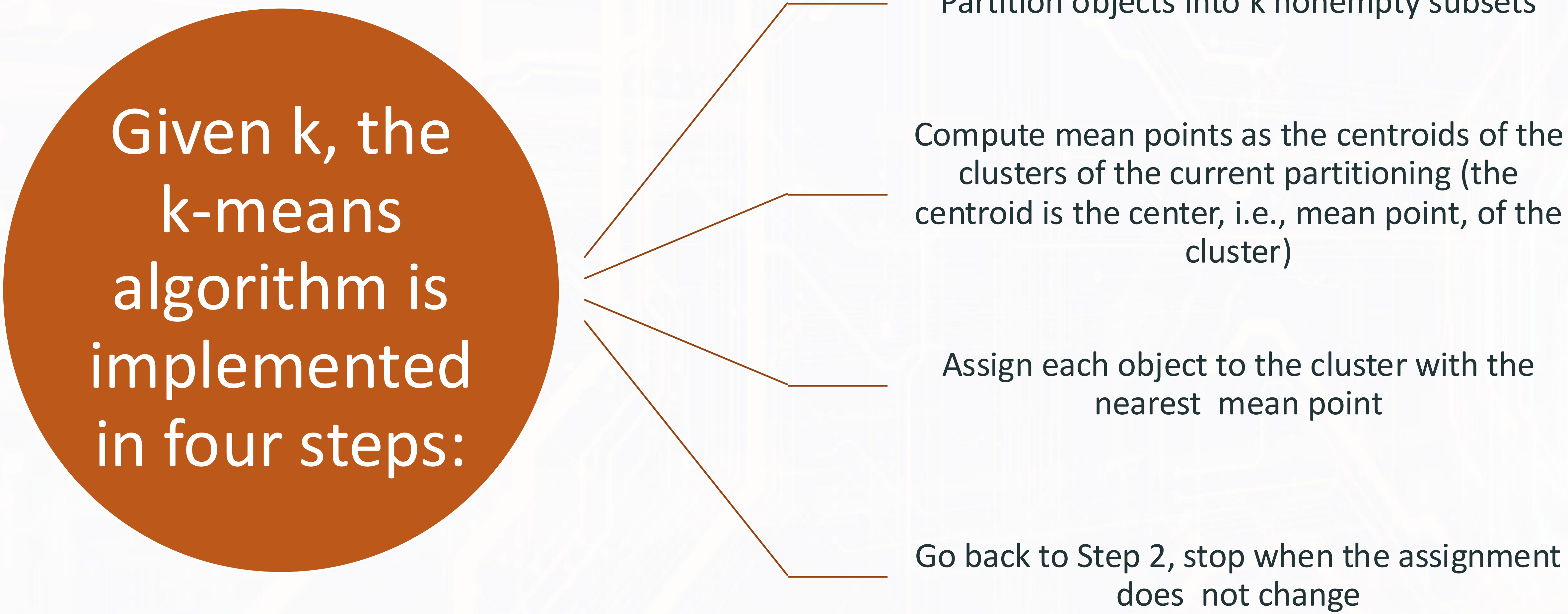
Partitioning method: Partitioning a database D of n objects into a set of k clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

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

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: Each cluster is represented by the center 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:

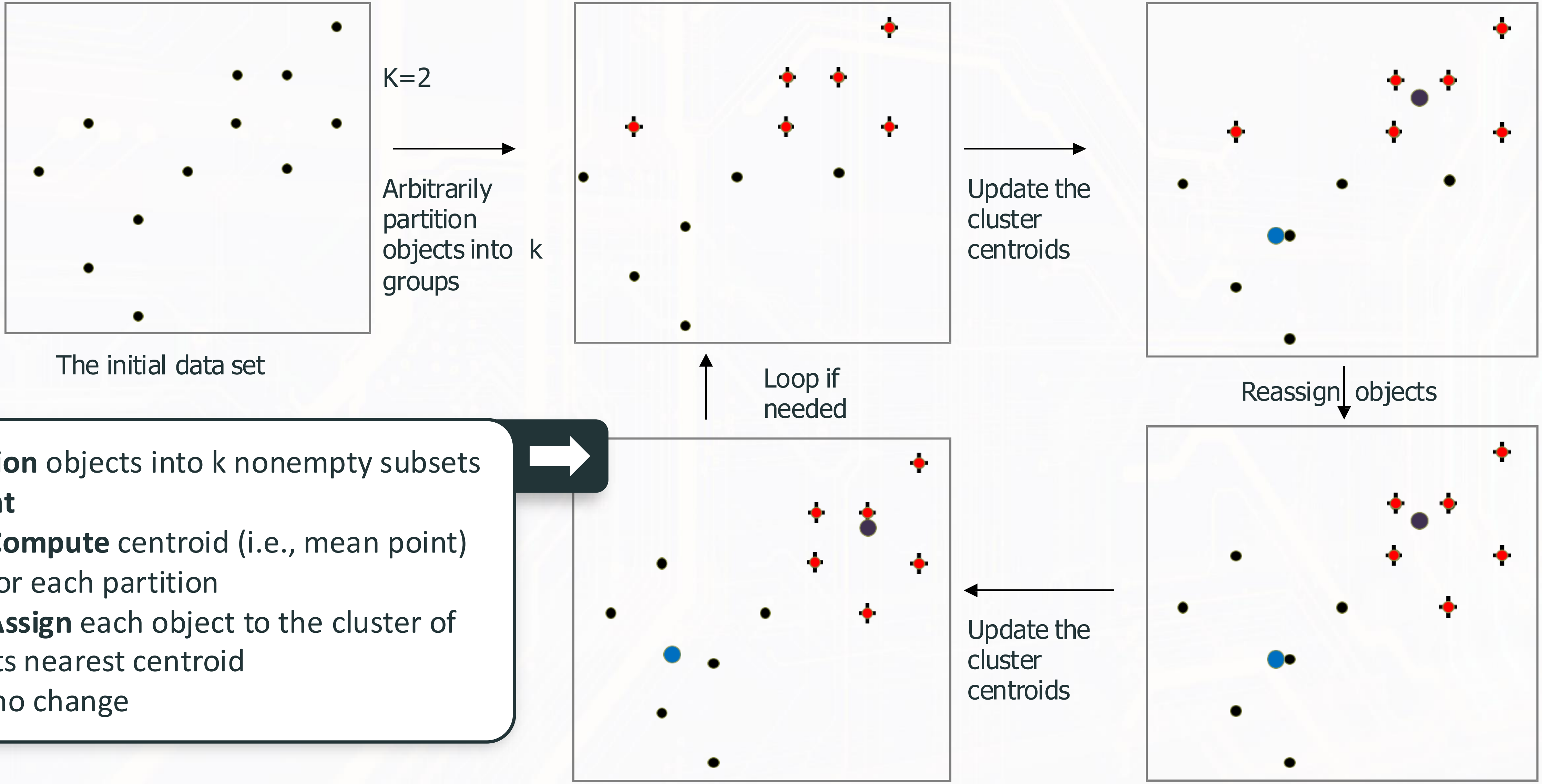
Partition objects into k nonempty subsets

Compute mean 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 mean point

Go back to Step 2, stop when the assignment does not change

An Example of K-Means Clustering



K=2

Arbitrarily partition objects into k groups

The initial data set

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
```

Comments on the K-Means Method

Strength

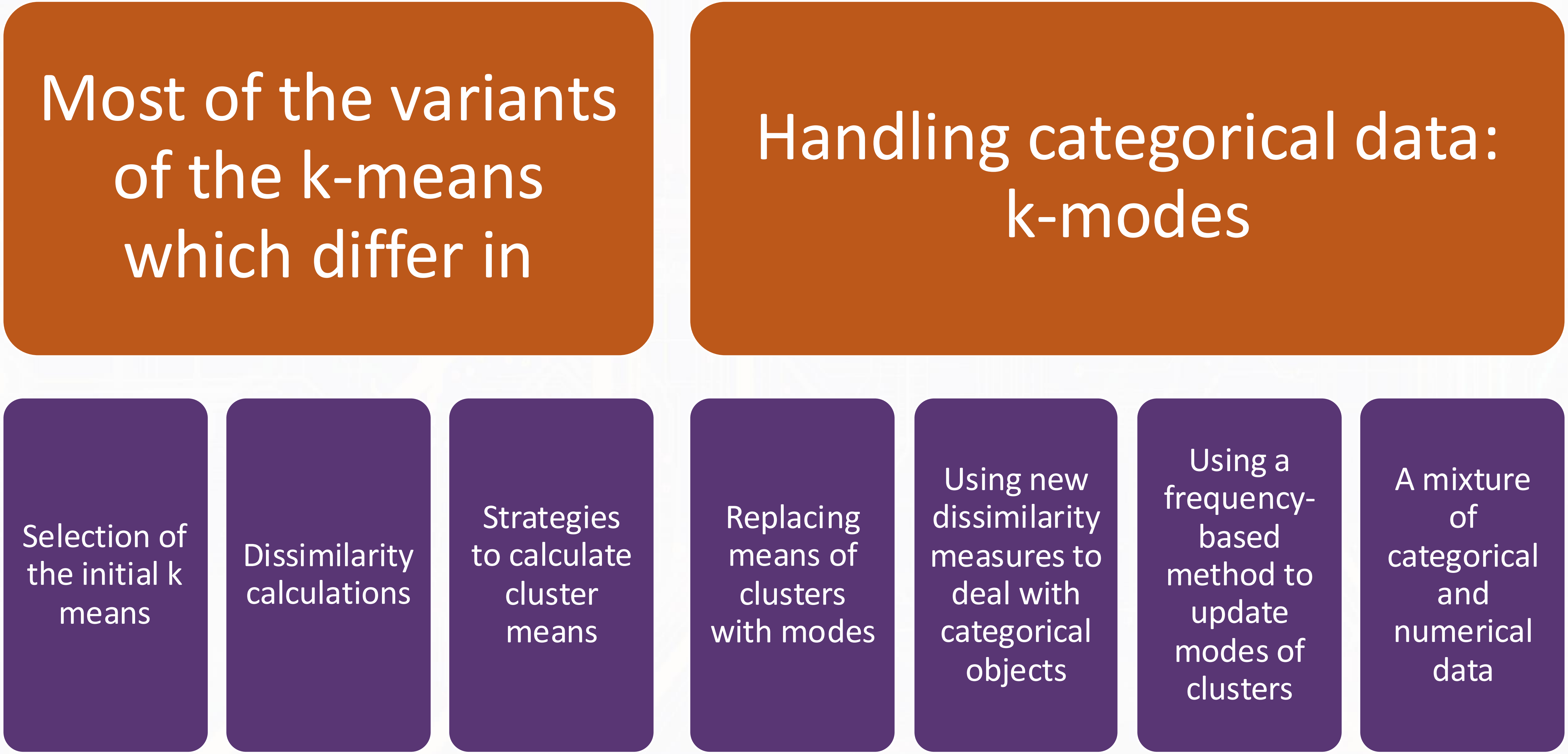
- **Efficient**: O(tkn), where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n.

  Comparing: PAM: O(k(n-k)2), CLARA: O(ks2 + 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

Variations of the K-Means Method



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

- Selection of the initial k means
- Dissimilarity calculations
- Strategies to calculate cluster means

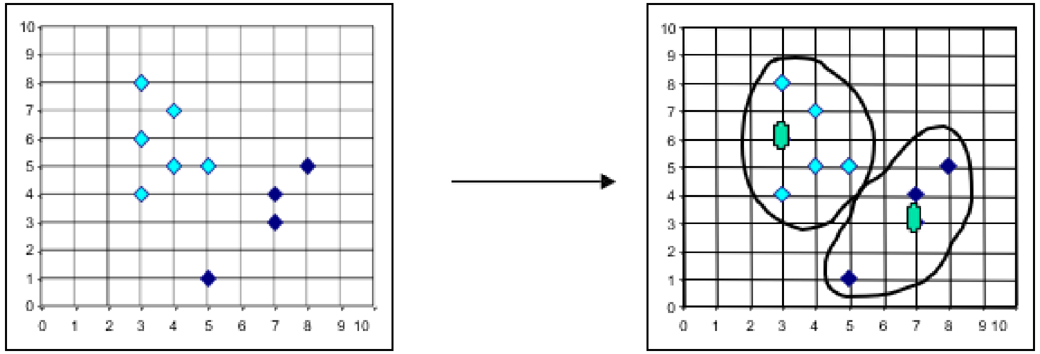
Handling categorical data: k-modes

- Replacing means of clusters with modes
- Using new dissimilarity measures to deal with categorical objects
- Using a frequency-based method to update modes of clusters
- A mixture of categorical and numerical data

What Is the Problem of the K-Means Method?

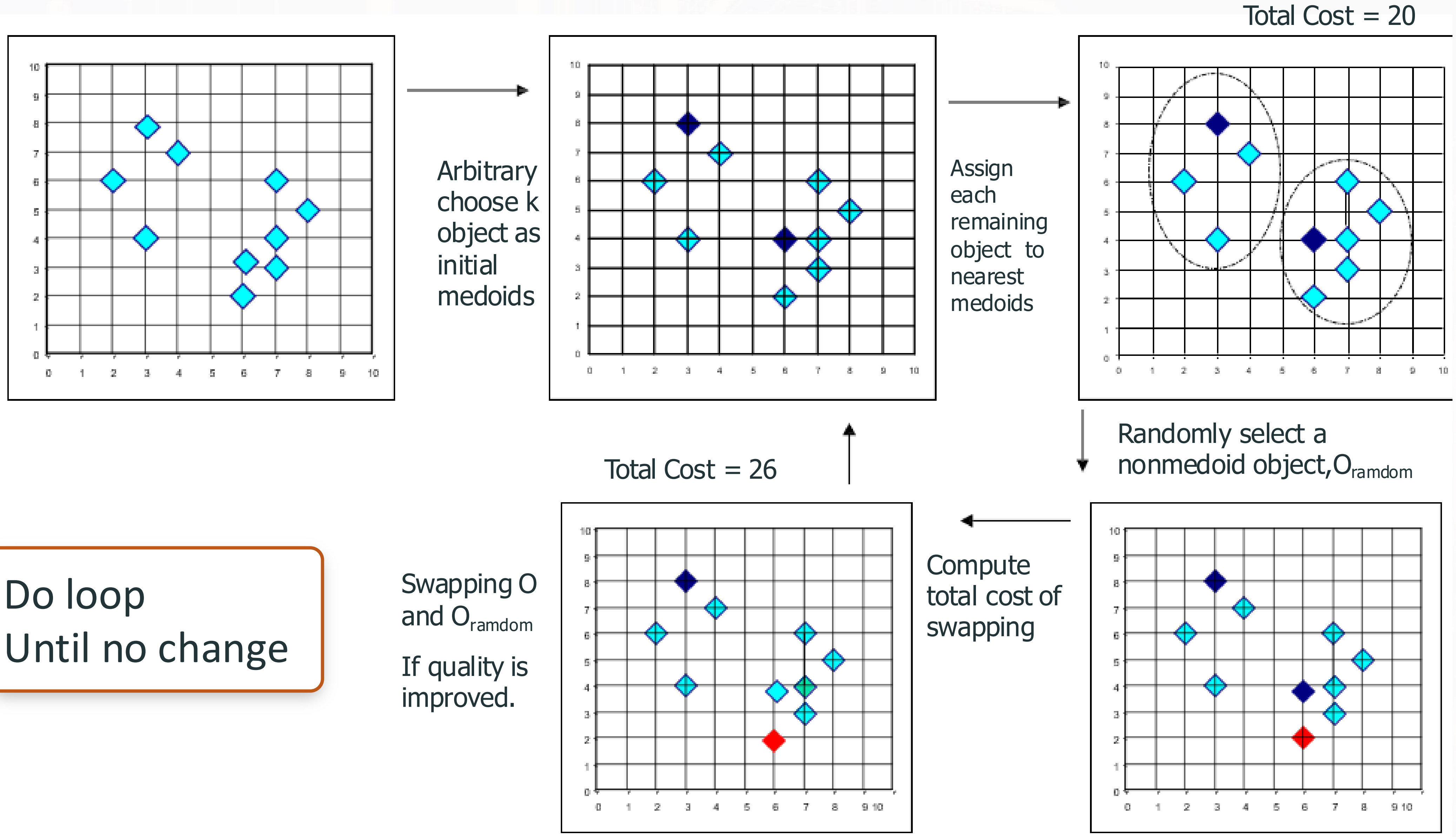
The k-means algorithm is sensitive to outliers !

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

PAM: A Typical K-Medoids Algorithm



Arbitrary choose k object as initial medoids

Assign each remaining object to nearest medoids

Total Cost = 20

Randomly select a nonmedoid object,O_{ramdom}

Compute total cost of swapping

Total Cost = 26

Swapping O and O_{ramdom}

If quality is improved.

!

Do loop
Until no change

The K-Medoid Clustering Method

K-Medoids Clustering: Find representative objects (medoids) in clusters

PAM (Partitioning Around Medoids)

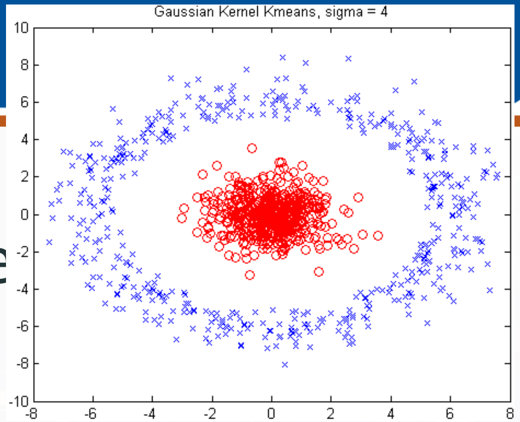
- Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
- PAM works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)

Efficiency improvement on PAM

CLARA: PAM on samples

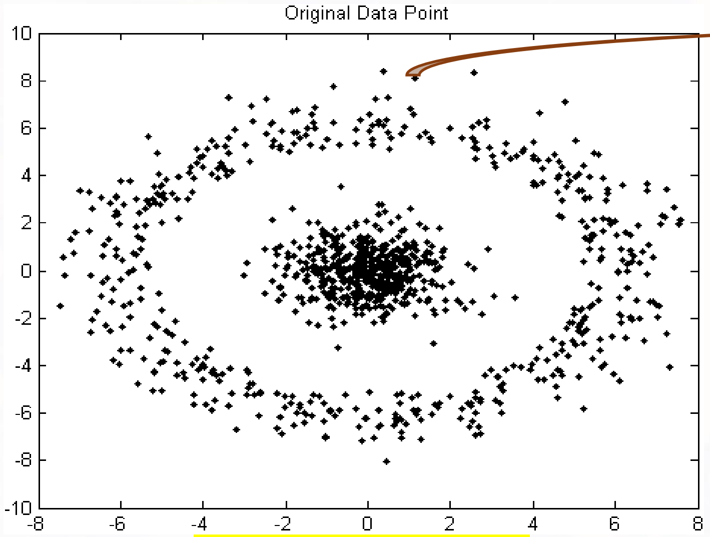
CLARANS: Randomized re-sampling

Kernel K-Means Clustering

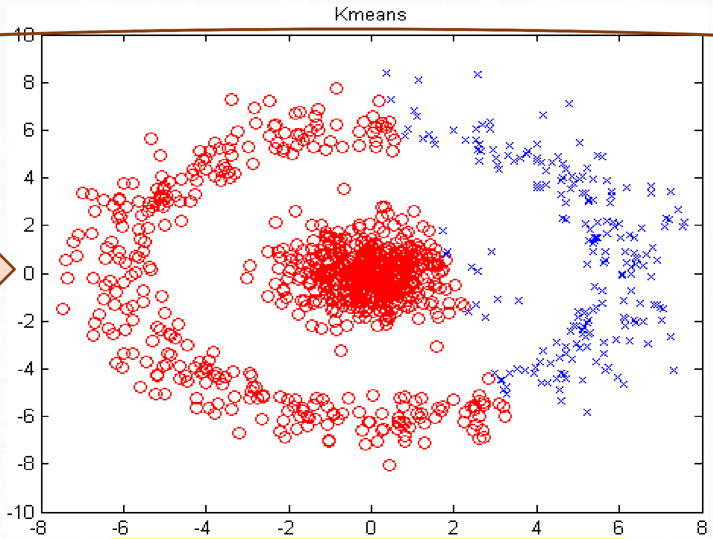


- Kernel K-Means can be used to detect non-convex cluste
  - K-Means can only detect clusters that are linearly separable
- Idea: Project data onto the high-dimensional kernel space, and then perform K-Means clustering
  - Map data points in the input space onto a high-dimensional feature space using the kernel function
  - Perform K-Means on the mapped feature space
- Computational complexity is higher than K-Means
  - Need to compute and store n x n kernel matrix generated from the kernel function on the original data
- The widely studied spectral clustering can be considered as a variant of Kernel K-Means clustering

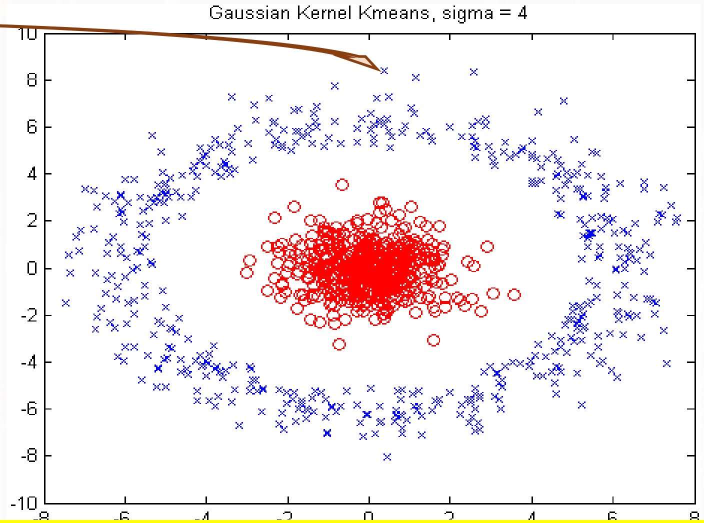
Example: Kernel K-Means Clustering



The original data set



The result of K-Means clustering



The result of Gaussian Kernel K-Means clustering

- ❑ The above data set cannot generate quality clusters by K-Means since it contains non-covex clusters
- ❑ Gaussian RBF Kernel transformation maps data to a kernel matrix K for any two points x_i, x_j: $K_{x_i x_j} = \phi(x_i) \bullet \phi(x_j)$ and Gaussian kernel: $K(\boldsymbol{X}_i, \boldsymbol{X}_j) = e^{-\|X_i - X_j\|^2 / 2\sigma^2}$
- ❑ K-Means clustering is conducted on the mapped data, generating quality clusters

Outline

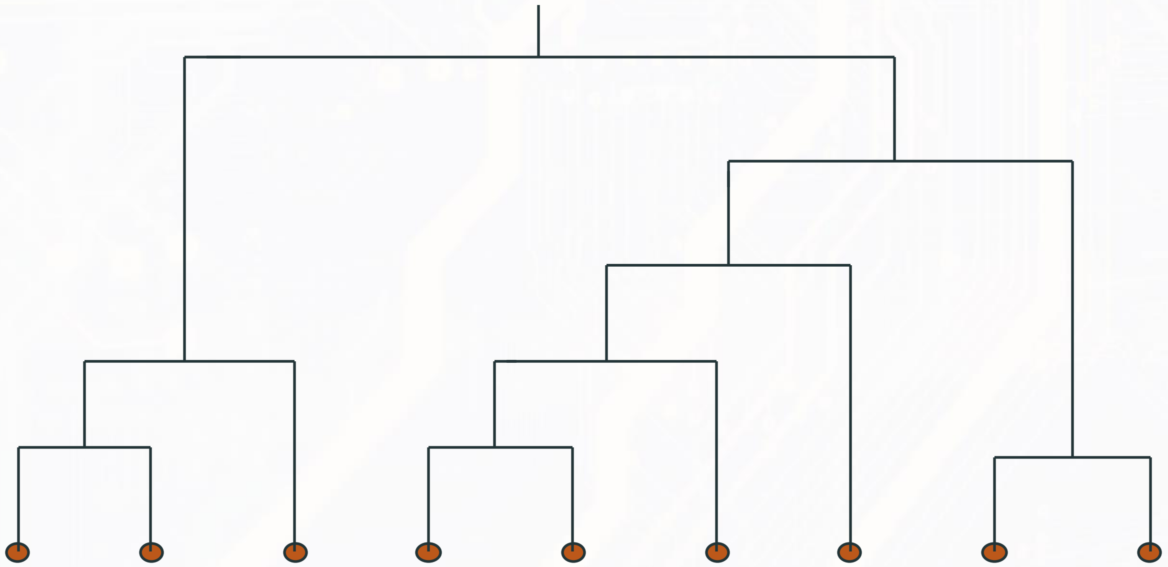
- Basic Concepts of Cluster Analysis
- Partitioning Methods
- Hierarchical Methods
- Density methods
- Density-based and grid-based methods

Hierarchical Clustering: Basic Concepts

- Hierarchical clustering
  - Generate a clustering hierarchy (drawn as a dendrogram)
  - Not required to specify K, the number of clusters
  - More deterministic
  - No iterative refinement
- Two categories of algorithms
  - Agglomerative: Start with singleton clusters, continuously merge two clusters at a time to build a bottom-up hierarchy of clusters
  - Divisive: Start with a huge macro-cluster, split it continuously into two groups, generating a top-down hierarchy of clusters

Dendrogram: How Clusters are Merged

- Dendrogram: Decompose a set of data objects into a tree of clusters by multi-level nested partitioning
- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster

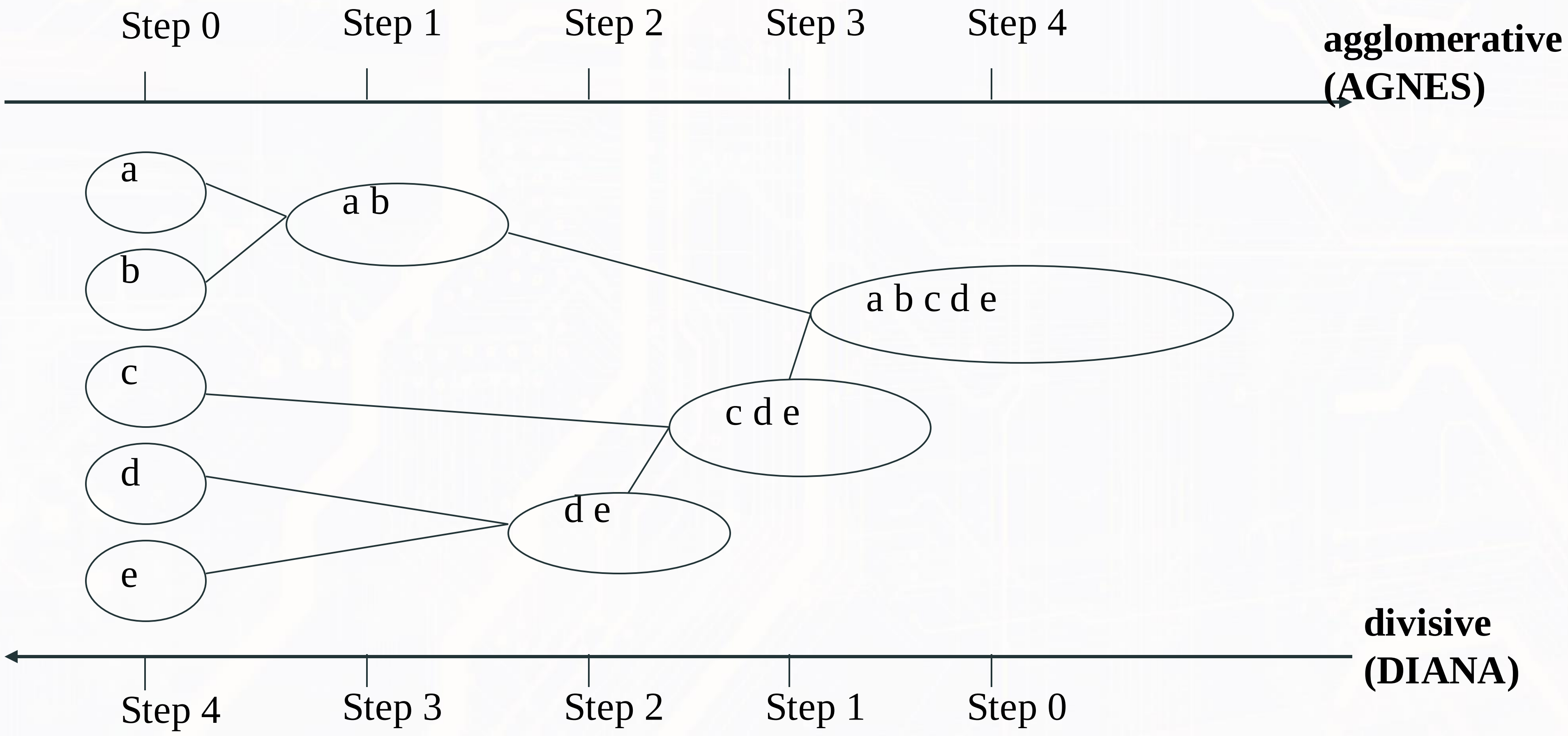


Hierarchical clustering generates a dendrogram (a hierarchy of clusters)

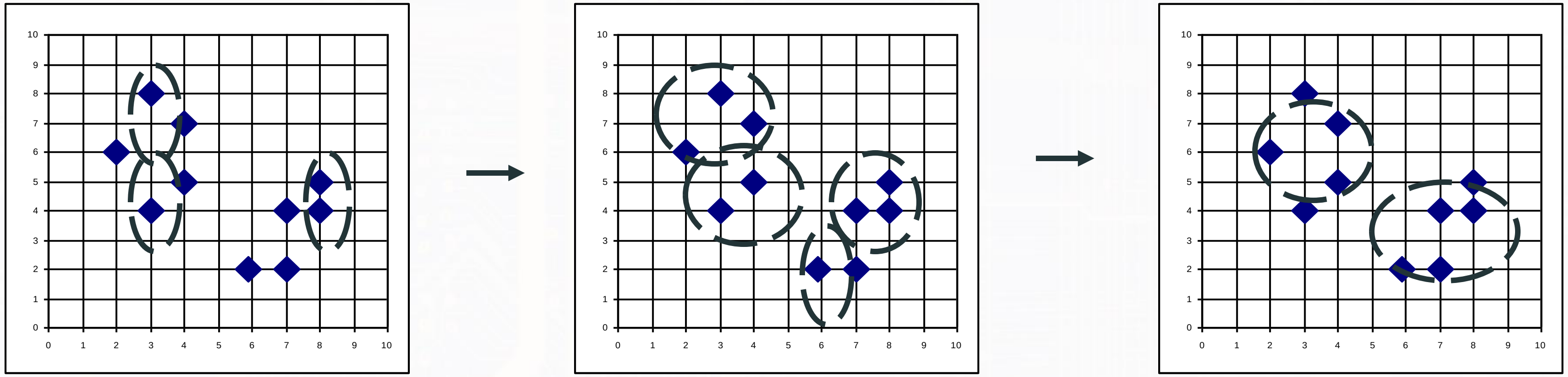
Agglomerative Clustering Algorithm

- AGNES (AGglomerative NESting) (Kaufmann and Rousseeuw, 1990)
  - Use the single-link method and the dissimilarity matrix
  - Continuously merge nodes that have the least dissimilarity
  - Eventually all nodes belong to the same cluster
- Agglomerative clustering varies on different similarity measures among clusters
  - Single link (nearest neighbor)
  - Complete link (diameter)
  - Average link (group average)
  - Centroid link (centroid similarity)

Agglomerative vs. Divisive Clustering

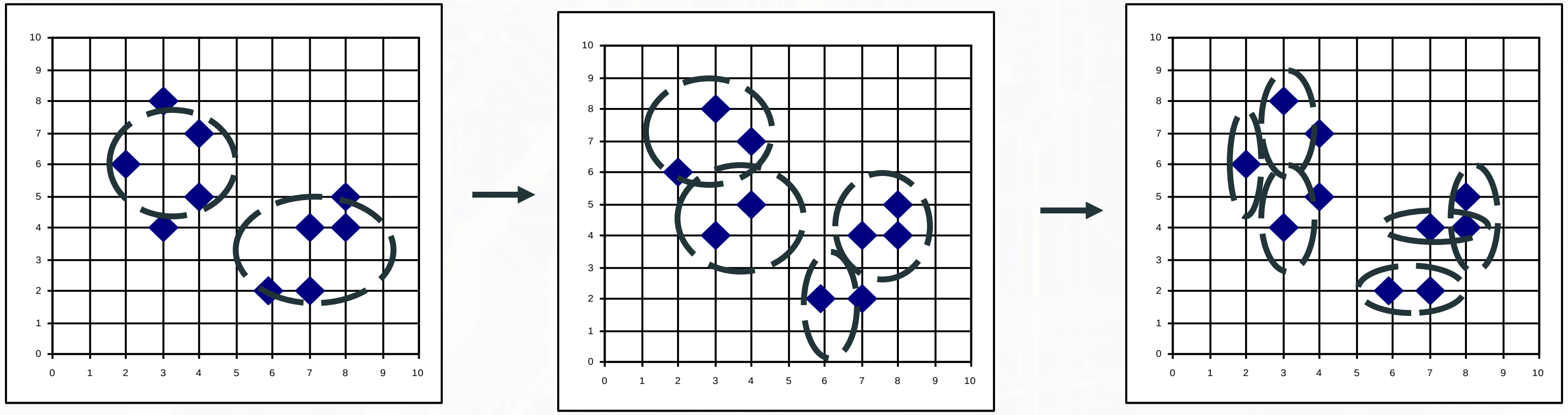


Agglomerative Clustering Algorithm



Divisive Clustering

- DIANA (Divisive Analysis) (Kaufmann and Rousseeuw,1990)
  - Implemented in some statistical analysis packages, e.g., Splus
- Inverse order of AGNES: Eventually each node forms a cluster on its own



Probabilistic Hierarchical Clustering

- Algorithmic hierarchical clustering
  - Nontrivial to choose a good distance measure
  - Hard to handle missing attribute values
  - Optimization goal not clear: heuristic, local search
- Probabilistic hierarchical clustering
  - Use probabilistic models to measure distances between clusters
  - Generative model: Regard the set of data objects to be clustered as a sample of the underlying data generation mechanism to be analyzed
  - Easy to understand, same efficiency as algorithmic agglomerative clustering method, can handle partially observed data
- In practice, assume the generative models adopt common distribution functions, e.g., Gaussian distribution or Bernoulli distribution, governed by parameters

Outline

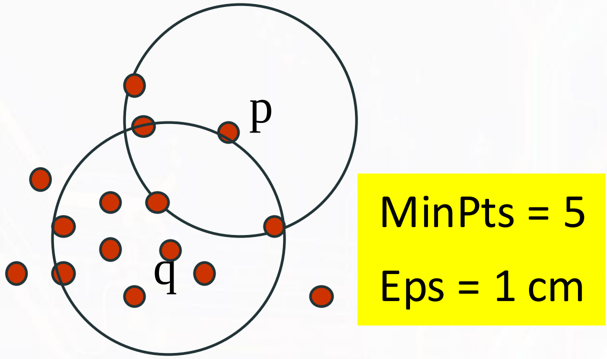
- Basic Concepts of Cluster Analysis
- Partitioning Methods
- Hierarchical Methods
- Density methods
- Density-based and grid-based methods

Density-Based Clustering Methods

- Clustering based on density (a local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan (only examine the local region to justify density)
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99)
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (also, grid-based)

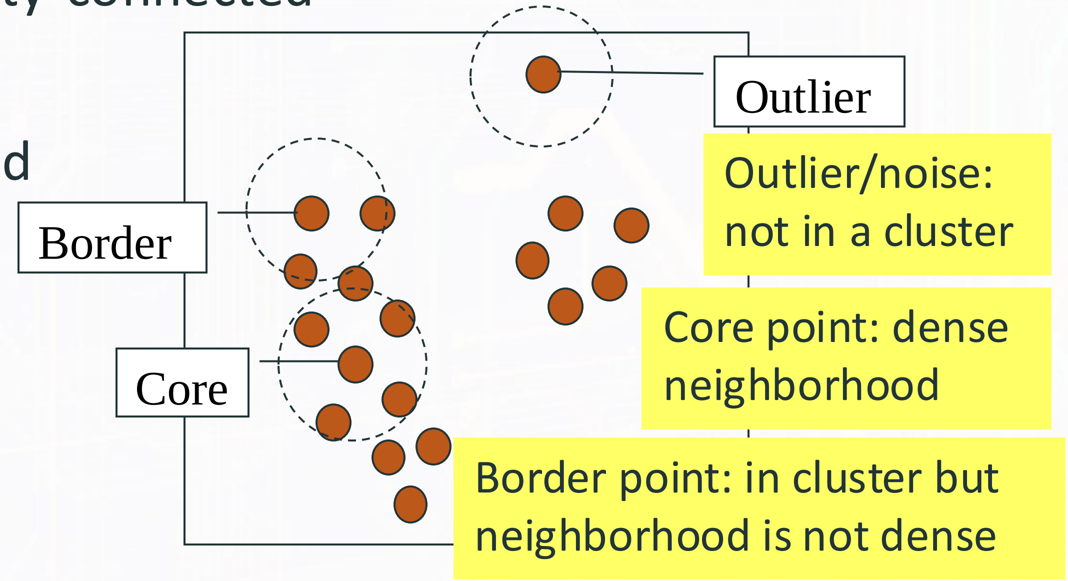
DBSCAN: A Density-Based Spatial Clustering Algorithm

- DBSCAN (M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, KDD’96)
  - Discovers clusters of arbitrary shape: Density-Based Spatial Clustering of Applications with Noise
- A density-based notion of cluster
  - A cluster is defined as a maximal set of density-connected points
  - Two parameters:
  - Eps (ε): Maximum radius of the neighborhood
  - MinPts: Minimum number of points in the
    - Eps-neighborhood of a point
- The Eps(ε)-neighborhood of a point q:
  - NEps(q): {p belongs to D | dist(p, q) ≤ Eps}



MinPts = 5

Eps = 1 cm



Outlier

Border

Core

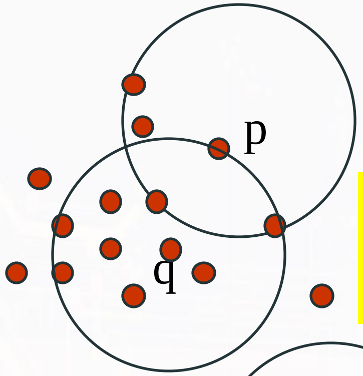
Outlier/noise: not in a cluster

Core point: dense neighborhood

Border point: in cluster but neighborhood is not dense

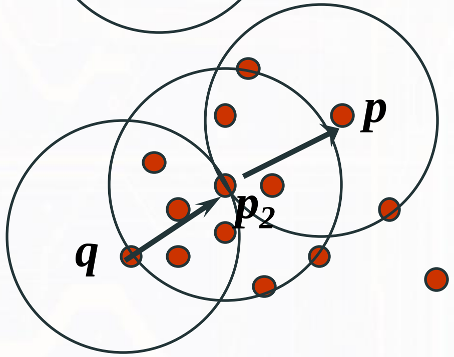
DBSCAN: Density-Reachable and Density-Connected

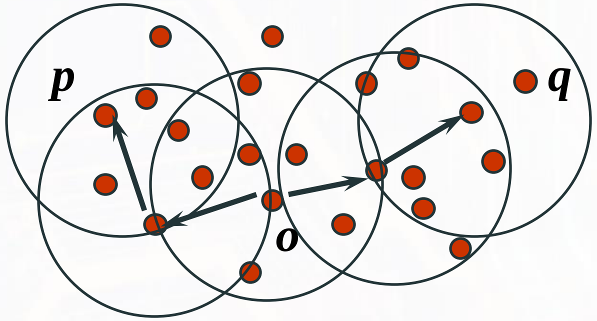
- Directly density-reachable:
  - A point p is directly density-reachable from a point q w.r.t. Eps (ε), MinPts if
    - p belongs to NEps(q)
    - core point condition: |NEps (q)| ≥ MinPts
- Density-reachable:
  - A point p is density-reachable from a point q w.r.t. Eps, MinPts if there is a chain of points $p_1, \dots, p_n, p_1 = q, p_n = p$ such that $p_i + 1$ is directly density-reachable from p_i
- Density-connected:
  - A point p is density-connected to a point q w.r.t. Eps, MinPts if there is a point o such that both p and q are density-reachable from o w.r.t. Eps and MinPts



MinPts = 5

Eps = 1 cm





DBSCAN Is Sensitive to the Setting of Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

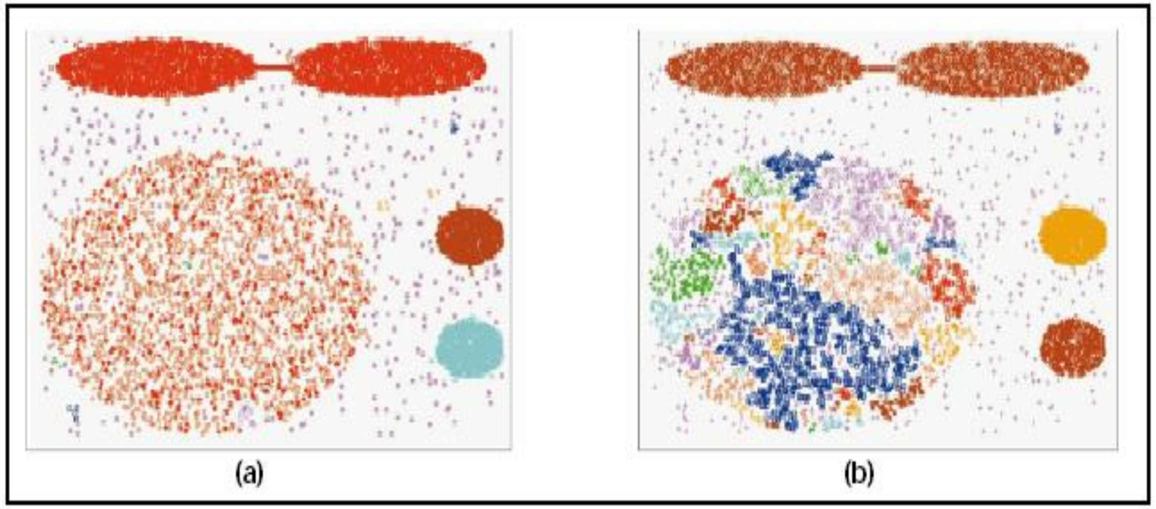
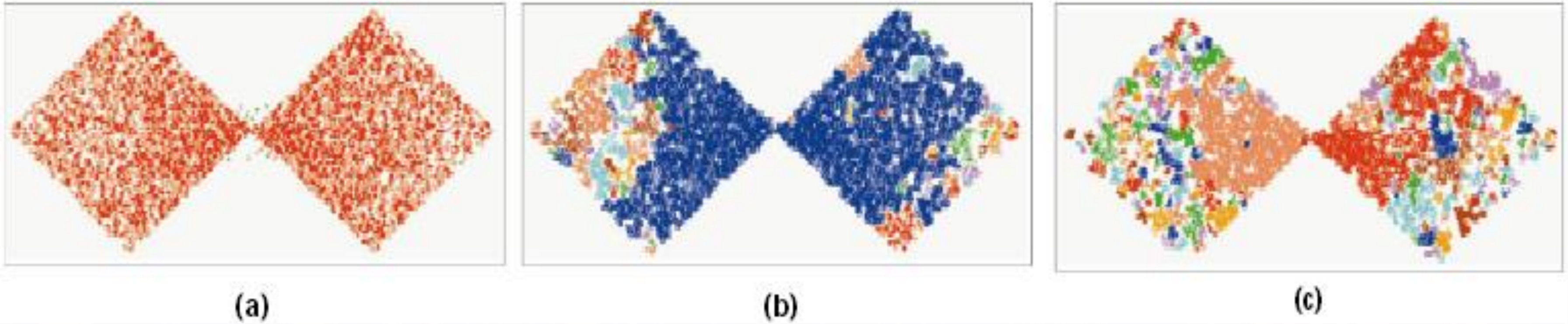


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



Ack. Figures from G. Karypis, E.-H. Han, and V. Kumar, *COMPUTER*, 32(8), 1999

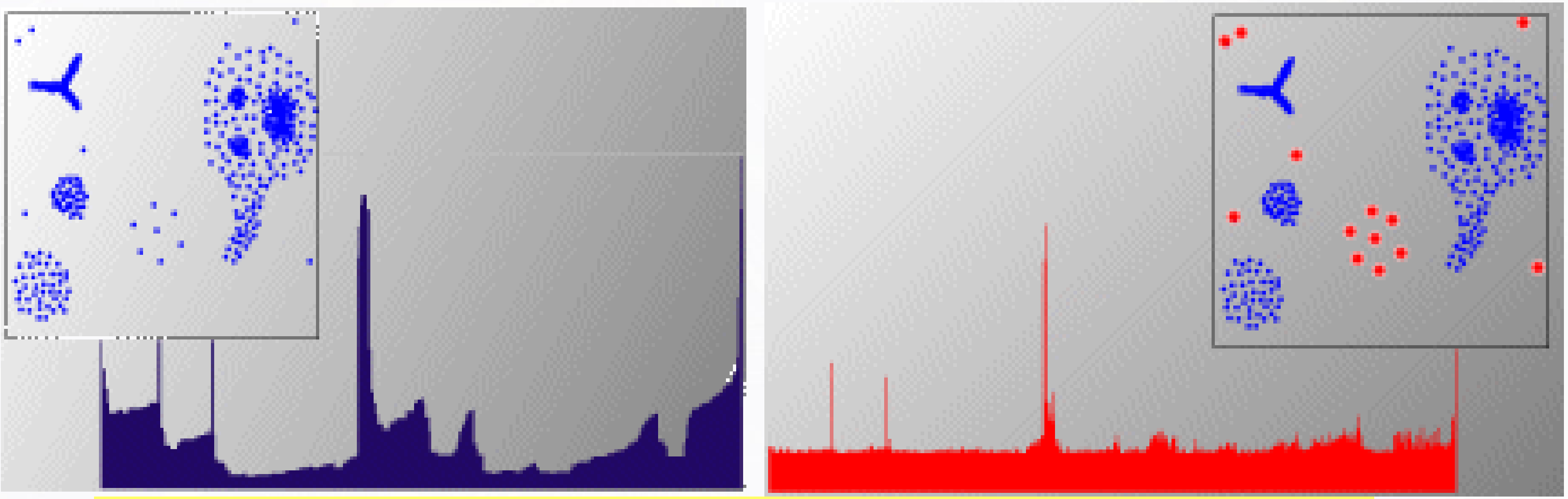
OPTICS: Ordering Points To Identify Clustering Structure

- OPTICS (Ankerst, Breunig, Kriegel, and Sander, SIGMOD’99)
  - DBSCAN is sensitive to parameter setting
  - An extension: finding clustering structure
- Observation: Given a MinPts, density-based clusters w.r.t. a higher density are completely contained in clusters w.r.t. to a lower density
- Idea: Higher density points should be processed first—find high-density clusters first
- OPTICS stores such a clustering order using two pieces of information:
  - Core distance and reachability distance

OPTICS: Finding Hierarchically Nested Clustering Structures

- OPTICS produces a special cluster-ordering of the data points with respect to its density-based clustering structure
- The cluster-ordering contains information equivalent to the density-based clusterings corresponding to a broad range of parameter settings
- Good for both automatic and interactive cluster analysis—finding intrinsic, even hierarchically nested clustering structures

OPTICS: Finding Hierarchically Nested Clustering Structures



Finding nested clustering structures with different parameter settings

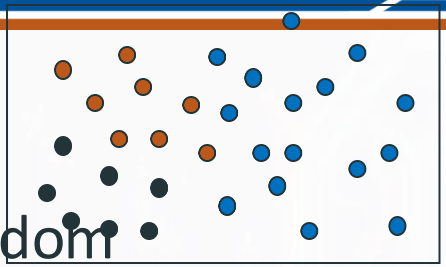
Outline

- Basic Concepts of Cluster Analysis
- Partitioning Methods
- Hierarchical Methods
- Density methods
- Density-based and grid-based methods

Evaluation of Clustering: Basic Concepts

- Evaluation of clustering
  - Assess the feasibility of clustering analysis on a data set
  - Evaluate the quality of the results generated by a clustering method
- Major issues on clustering assessment and validation
  - Clustering tendency: assessing the suitability of clustering: whether the data has any inherent grouping structure
  - Determining the Number of Clusters: determining for a dataset the right number of clusters that may lead to a good quality clustering
  - Clustering quality evaluation: evaluating the quality of the clustering results

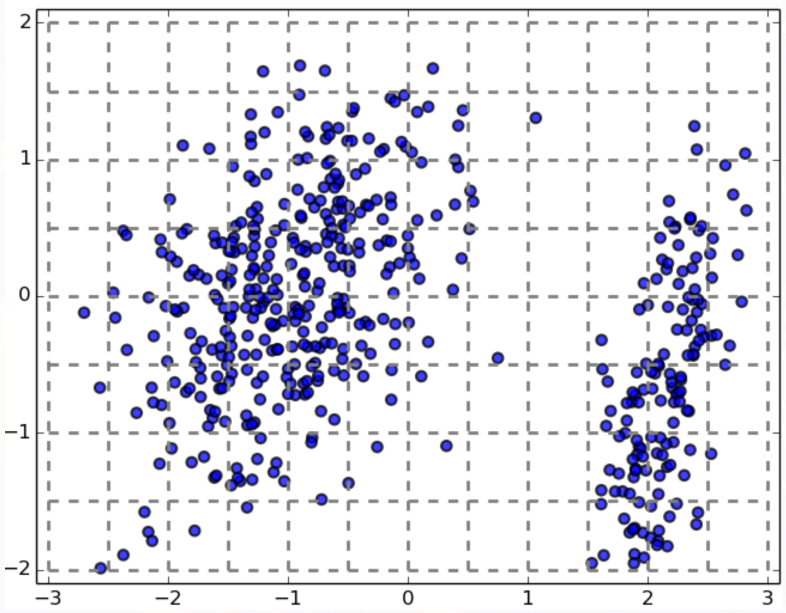
Clustering Tendency: Whether the Data Contains Inherent Grouping Structure



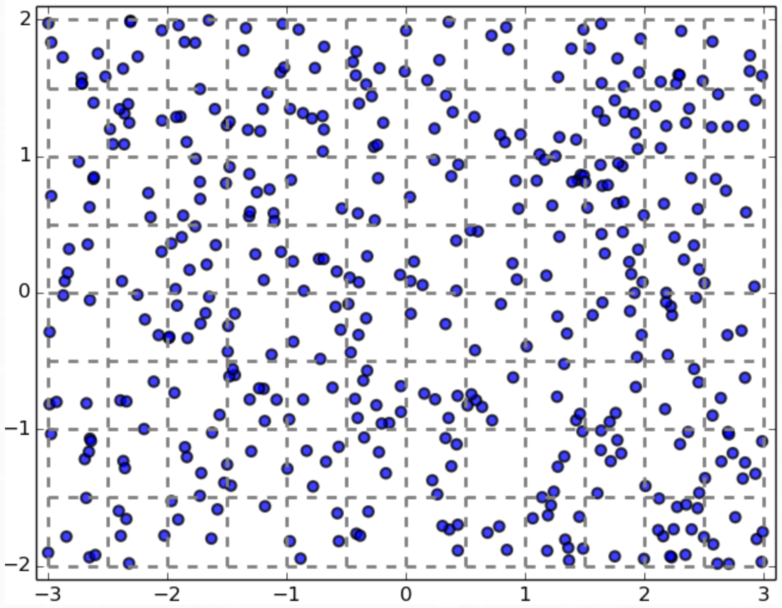
- Assess the suitability of clustering
  - Whether the data has any "inherent grouping structure" — non-random structure that may lead to meaningful clusters
- Determine clustering tendency or clusterability
  - A hard task because there are so many different definitions of clusters
    - Different definitions: Partitioning, hierarchical, density-based and graph-based
  - Even fixing a type, still hard to define an appropriate null model for a data set
- There are some clusterability assessment methods, such as
  - Spatial histogram: Contrast the histogram of the data with that generated from random samples
  - Distance distribution: Compare the pairwise point distance from the data with those from the randomly generated samples
  - Hopkins Statistic: A sparse sampling test for spatial randomness

Testing Clustering Tendency: A Spatial Histogram Approach

- Spatial Histogram Approach: Contrast the d-dimensional histogram of the input dataset D with the histogram generated from random samples
  - Dataset D is clusterable if the distributions of two histograms are rather different



(a) Input dataset



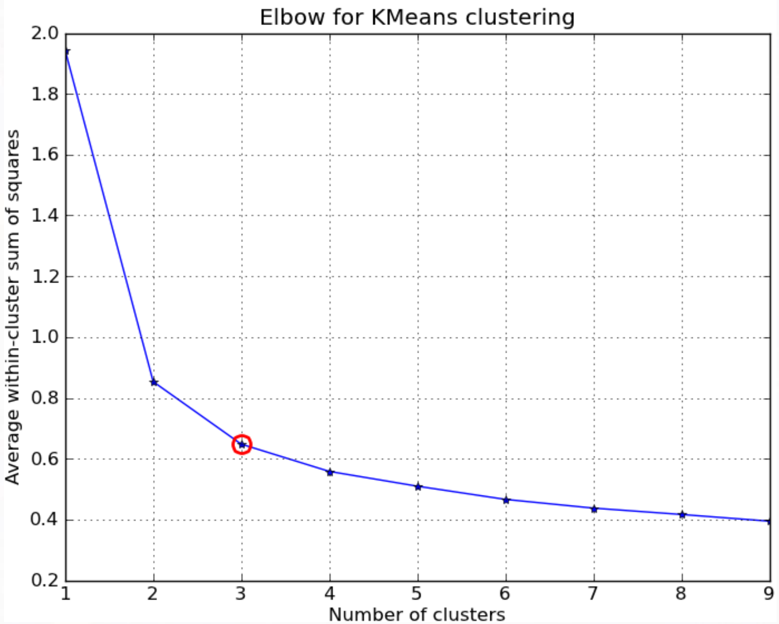
(b) Data generated from random samples

Determining the Number of Clusters

- The right number of clusters often depends on the distribution's shape and scale in the data set, as well as the clustering resolution required by the user
- Methods for determining the number of clusters
- An empirical method
  - # of clusters: $k \approx \sqrt{\frac{n}{2}}$ for a dataset of n points (e.g., n = 200, k = 10)
  - Each cluster is expected to have about $\sqrt{2n}$ points

Finding the Number of Clusters: the Elbow Method

- Use the turning point in the curve of the sum of within cluster variance with respect to the # of clusters
  - Increasing the # of clusters can help reduce the sum of within-cluster variance of each cluster
  - But splitting a cohesive cluster gives only a small reduction

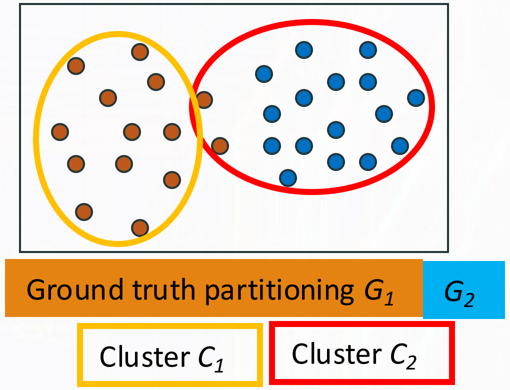


Finding K, the Number of Clusters: A Cross Validation Method

- Divide a given data set into m parts, and use m – 1 parts to obtain a clustering model
- Use the remaining part to test the quality of the clustering
    - For example, for each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and their closest centroids to measure how well the model fits the test set
- For any k > 0, repeat it m times, compare the overall quality measure w.r.t. different k’s, and find # of clusters that fits the data the best

Commonly Used Extrinsic Methods

- Matching-based methods
  - Examine how well the clustering results match the ground truth in partitioning the objects in the data set
- Information theory-based methods
  - Compare the distribution of the clustering results and that of the ground truth
  - Information theory (e.g., entropy) used to quantify the comparison
  - Ex. Conditional entropy, normalized mutual information (NMI)
- Pairwise comparison-based methods
  - Treat each group in the ground truth as a class, and then check the pairwise consistency of the objects in the clustering results
  - Ex. Four possibilities: TP, FN, FP, TN; Jaccard coefficient



Information Theory-Based Methods (II)
Normalized Mutual Information (NMI)

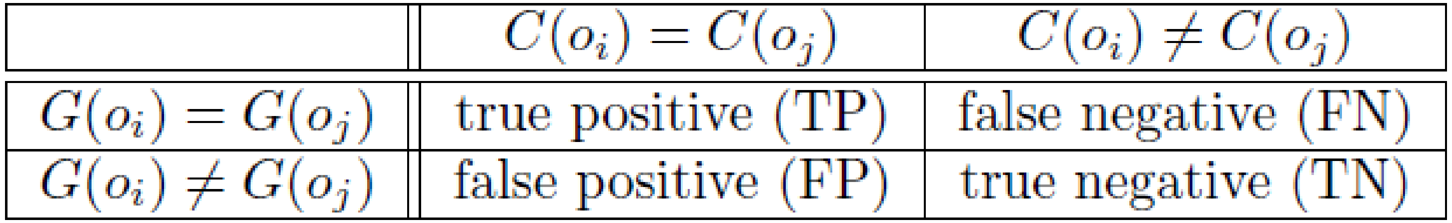
- Mutual information $I(C,G) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij} \log \frac{p_{ij}}{p_{C_i} p_{G_j}}$
  - Quantify the amount of shared info between the clustering C and the ground-truth partitioning G
  - Measure the dependency between the observed joint probability p_{ij} of C and G, and the expected joint probability $p_{C_i} p_{G_j}$ under the independence assumption
  - When C and G are independent, $p_{ij} = p_{C_i} p_{G_j}$, I(C, G) = 0
  - However, there is no upper bound on the mutual information
- Normalized mutual information $NMI(C,G) = \sqrt{\frac{I(C,G)}{H(C)} \frac{I(C,G)}{H(G)}} = \frac{I(C,G)}{\sqrt{H(C)H(G)}}$
  - Value range of NMI: [0,1]
  - Value close to 1 indicates a good clustering

Pairwise Comparison-Based Methods: Jaccard Coefficient

- Pairwise comparison: treat each group in the ground truth as a class
- For each pair of objects (o_i, o_j) in D, if they are assigned to the same cluster/group, the assignment is regarded as positive; otherwise, negative
  - Depending on assignments, we have four possible cases:

Note: Total # of pairs of points

$$N = \binom{n}{2}$$



	$C(o_i) = C(o_j)$	$C(o_i) \neq C(o_j)$
$G(o_i) = G(o_j)$	true positive (TP)	false negative (FN)
$G(o_i) \neq G(o_j)$	false positive (FP)	true negative (TN)

- **Jaccard coefficient:** Ignoring the true negatives (thus asymmetric)
  - *Jaccard* = *TP*/(*TP* + *FN* + *FP*) [i.e., denominator ignores *TN*]
    - *Jaccard* = 1 if perfect clustering
- Many other measures are based on the pairwise comparison statistics:
  - Rand statistic
  - Fowlkes-Mallows measure

Intrinsic Methods (I): Dunn Index

$$DI = \frac{\delta}{\Delta}$$

- Intrinsic methods (i.e., no ground truth) examine how compact clusters are and how well clusters are separated, based on similarity/distance measure between objects
- Dunn Index:
  - The compactness of clusters: the maximum distance between two points that belong to the same cluster: $\Delta = \max_{C(o_i)=C(o_j)} \{d(o_i, o_j)\}$
  - The degree of separation among different clusters: the minimum distance between two points that belong to different clusters: $\delta = \min_{C(o_i)\neq C(o_j)} \{d(o_i, o_j)\}$
  - The Dunn index is simply the ratio: $DI = \frac{\delta}{\Delta}$, the larger the ratio, the farther away the clusters are separated comparing to the compactness of the clusters
- Dunn index uses the extreme distances to measure the cluster compactness and inter-cluster separation and it can be affected by outliers

Intrinsic Methods (II): Silhouette Coefficient

- Suppose D is partitioned into k clusters: $C_1, \ldots, C_k$
- For each object o in D, we calculate
  - $a(o) = \frac{\sum_{o' \in C_i, o \neq o'} dist(o,o')}{|C_i|-1}$: avg distance between o and all other objects in the cluster to which o belongs, reflects the compactness of the cluster to which o belongs
  - $b(o) = \min_{C_j: 1 \le j \le k, j \neq i} \left\{ \frac{\sum_{o' \in C_j} dist(o,o')}{|C_j|} \right\}$: minimum avg distance from o to all clusters to which o does not belong, captures the degree to which o is separated from other clusters
- Silhouette Coefficient: $s(o) = \frac{b(o)-a(o)}{\max\{a(o),b(o)\}}$, value range (-1, 1)
  - When the value of o approaches 1, the cluster containing o is compact and o is far away from other clusters, which is the preferable case
  - When the value is negative (i.e., b(o) < a(o)), o is closer to the objects in another cluster than to the objects in the same cluster as o: a bad situation to be avoided

Attributions

Material presented in this lecture was adapted from

Han, J., Kamber, M., & Pei, J. (2012). *Data mining concepts and techniques* (3rd ed.). Waltham, MA: Elsevier. Retrieved from https://hanj.cs.illinois.edu/bk3/bk3_slidesindex.htm

and

Canny, J., Franklin, M., Bruckner, Sparks, E., & Venkataraman, S. (2014). CIS194 introduction to data science [PowerPoint]. Retrieved from https://bcourses.berkeley.edu/courses/1267848/files/folder/lectures