

Cluster Analysis: Basic Concepts and Methods

Prepared by Dr. Amjad Qtaish

DS601
Data Mining

Lecture 09



Lecture: #09

Cluster Analysis: Basic Concepts and Methods



Dr. Amjad Qtaish

*Faculty of Informatics
Midocean University*





Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Evaluation of Clustering
- Summary





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



- **Biology:** taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- **Information retrieval:** document clustering
- **Land use:** Identification of areas of similar land use in an earth observation database
- **Marketing:** Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs



- **Summarization:**
 - Preprocessing for regression, PCA, classification, and association analysis
- **Compression:**
 - Image processing: vector quantization
- **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



- 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



- 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



- **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)



- **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





Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods 
- Hierarchical Methods
- Density-Based Methods
- Evaluation of Clustering
- Summary





- **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)

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

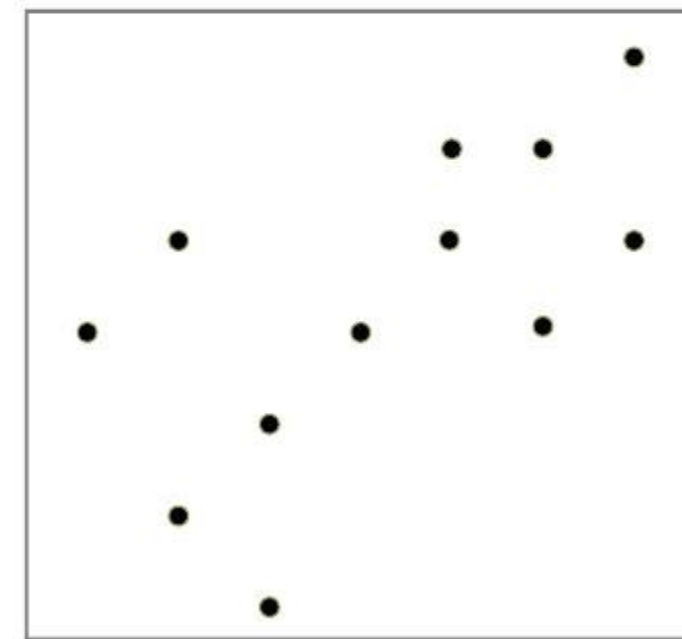
- Given k , find a partition of k clusters that optimizes the chosen partitioning criterion
- Global optimal: exhaustively enumerate all partitions
- Heuristic methods: *k-means* and *k-medoids* algorithms
- *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
- *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster



- 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



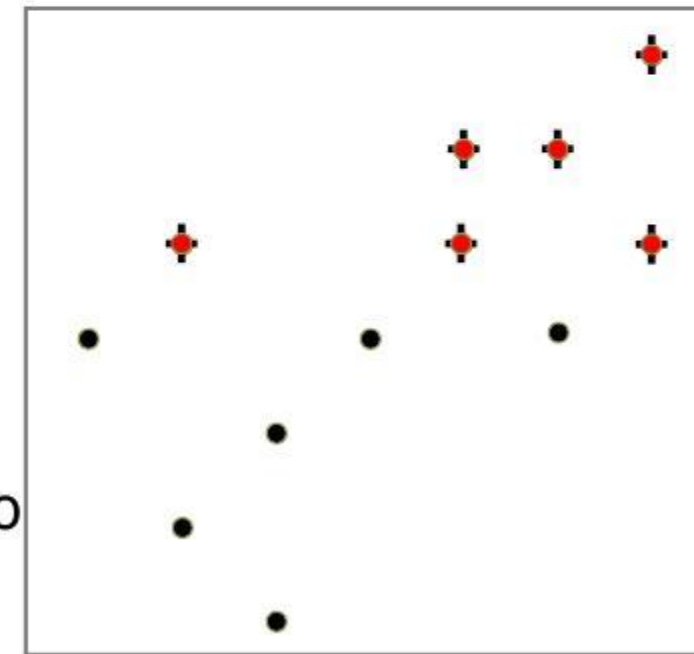
An Example of K-Means Clustering



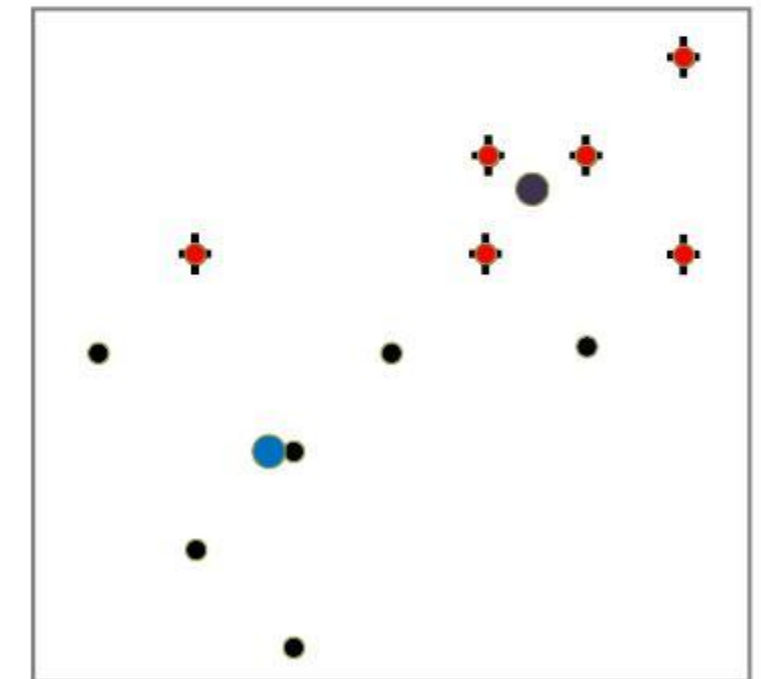
The initial data set

$K=2$

Arbitrarily
partition
objects into
 k groups

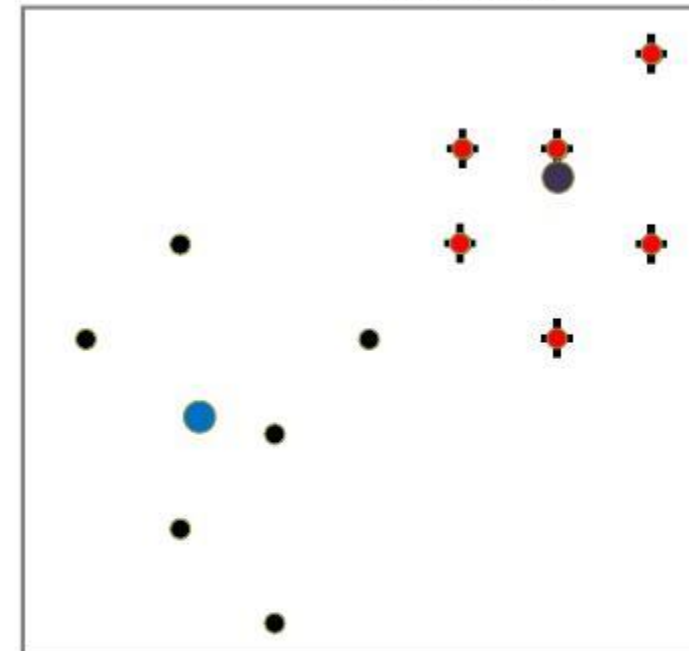


Update the
cluster
centroids

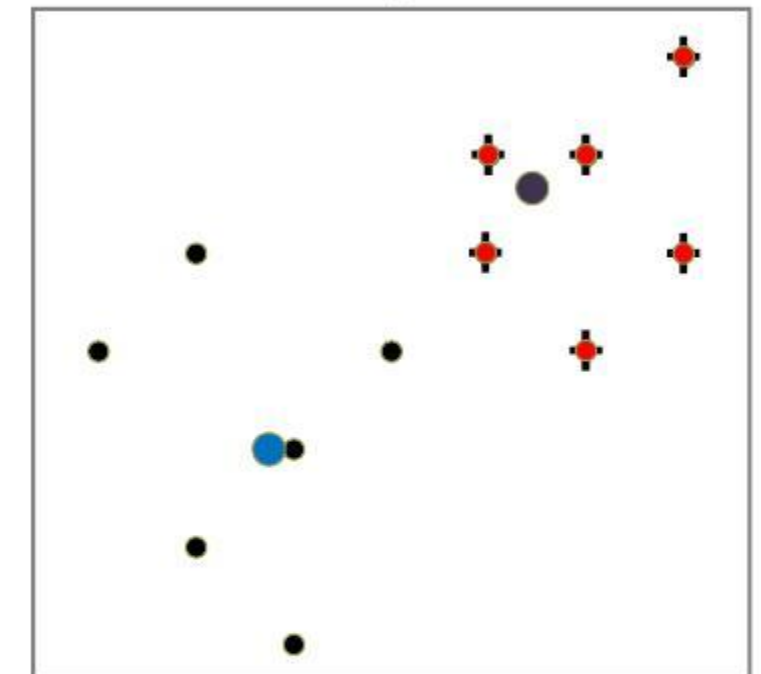


Reassign objects

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



Comments on the K-Means Method

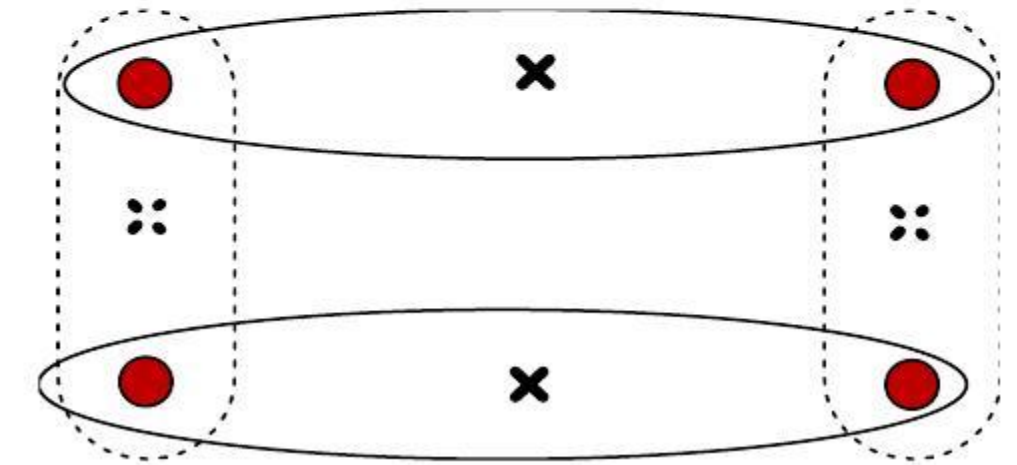


- **Strength**: *Efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t \ll n$.
 - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- **Comment**: Often terminates at a *local optimal*.
- **Weakness**
 - Applicable only to objects in a continuous n -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*



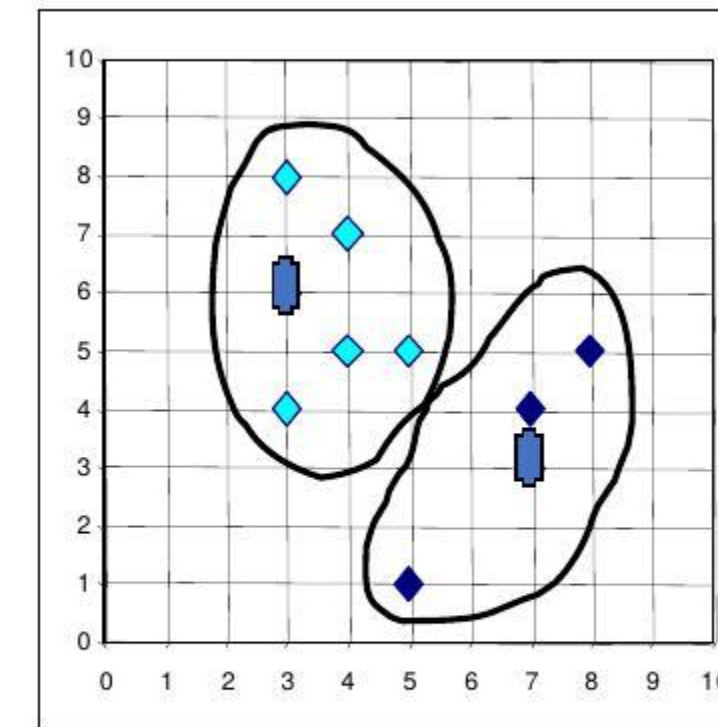
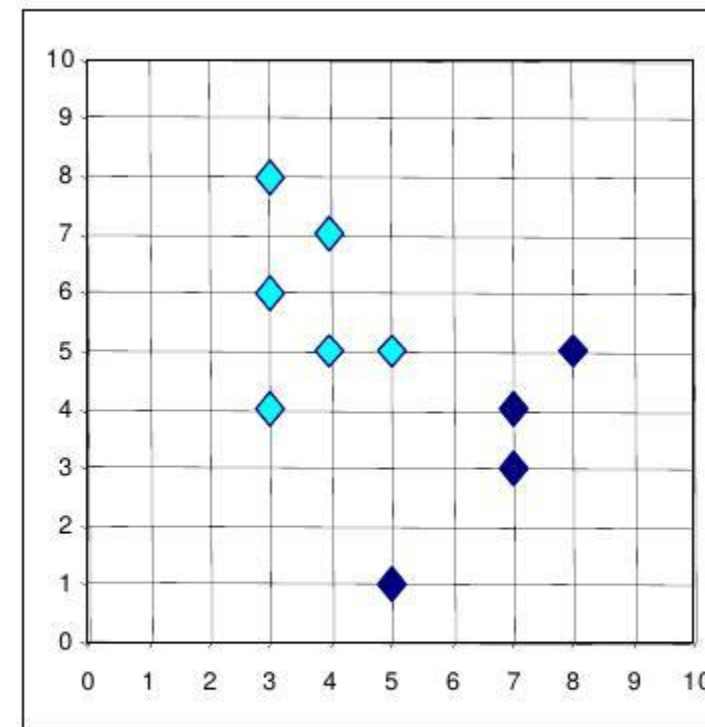


- Most of the variants of the *k-means* which differ in
 - Selection of the initial *k* means
 - Dissimilarity calculations
 - Strategies to calculate cluster means
- Handling categorical data: *k-modes*
 - Replacing means of clusters with modes
 - Using new dissimilarity measures to deal with categorical objects
 - Using a frequency-based method to update modes of clusters
 - A mixture of categorical and numerical data: *k-prototype* 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



K=2

**Do loop
Until no
change**

Swapping O
and O_{random}
If quality is
improved.

Arbitrary
choose k
object as
initial
medoids

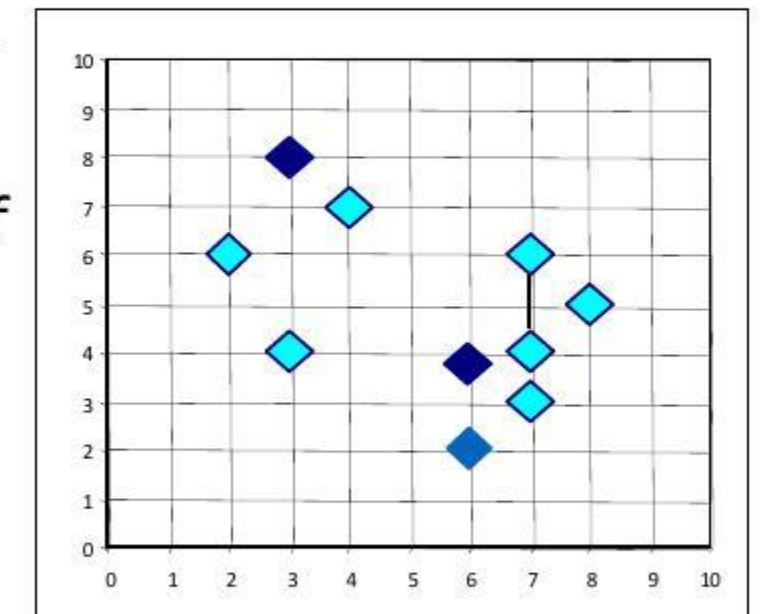
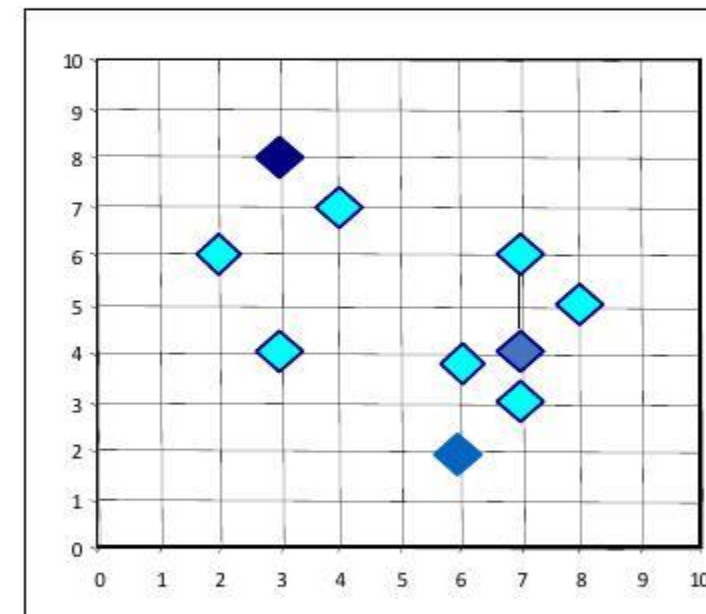
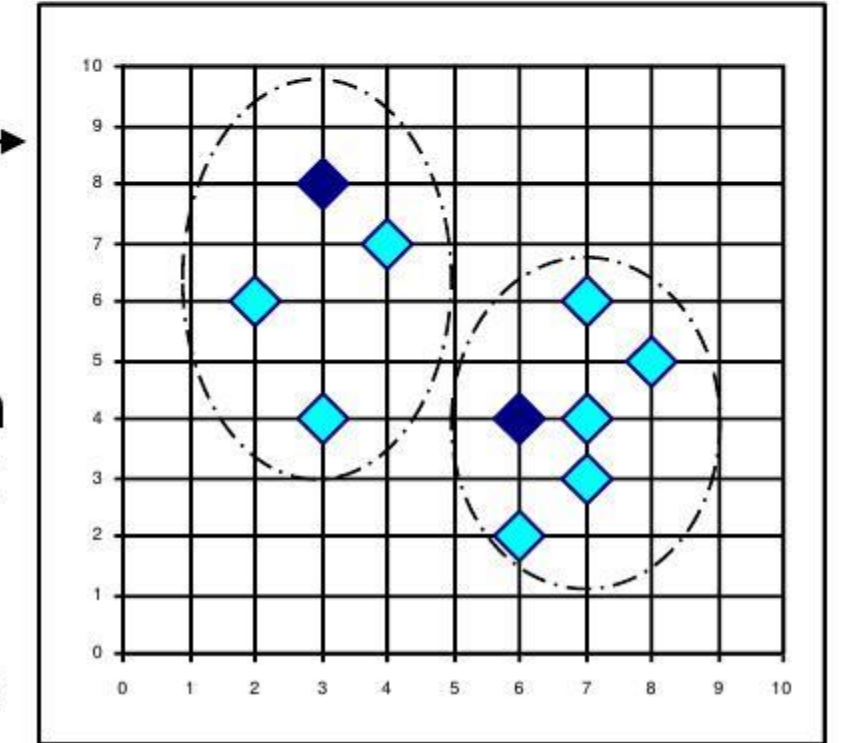
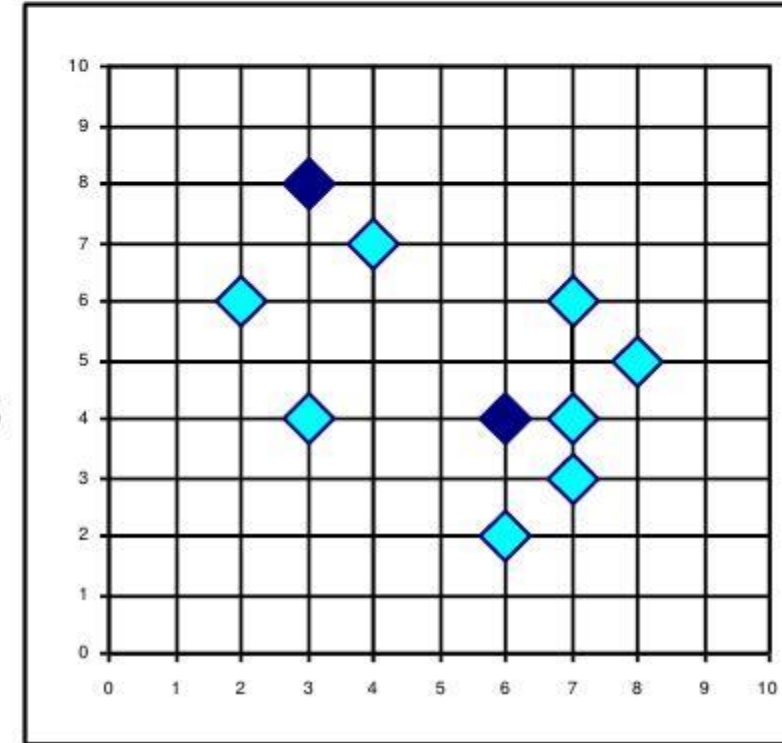
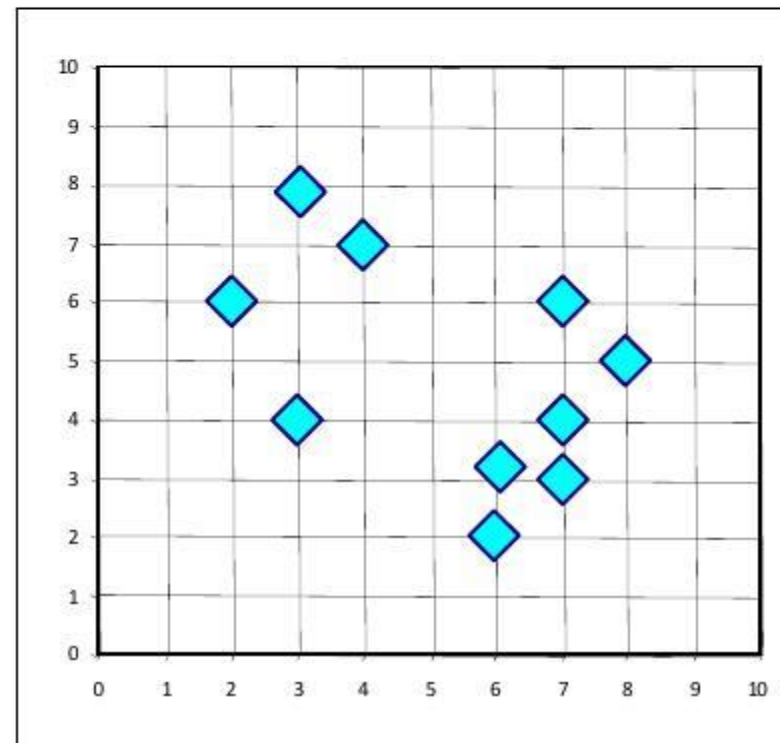
Assign
each remainin
g object to
nearest
medoids

Randomly select a
nonmedoid object, O_{random}

Compute
total cost of
swapping

Total Cost = 20

Total Cost = 26





Cluster Analysis: Basic Concepts and Methods

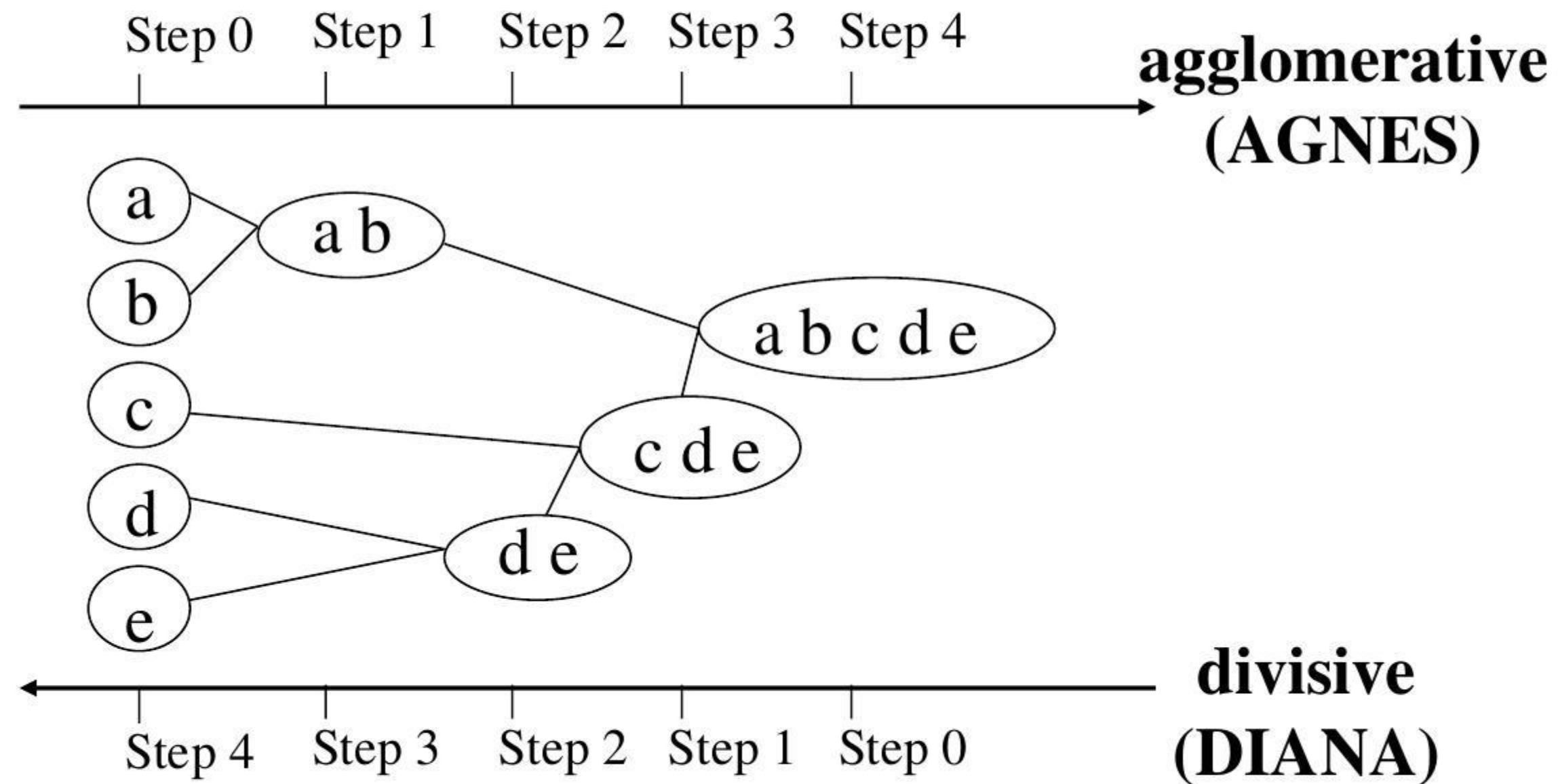
- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods 
- Density-Based Methods
- Evaluation of Clustering
- Summary





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

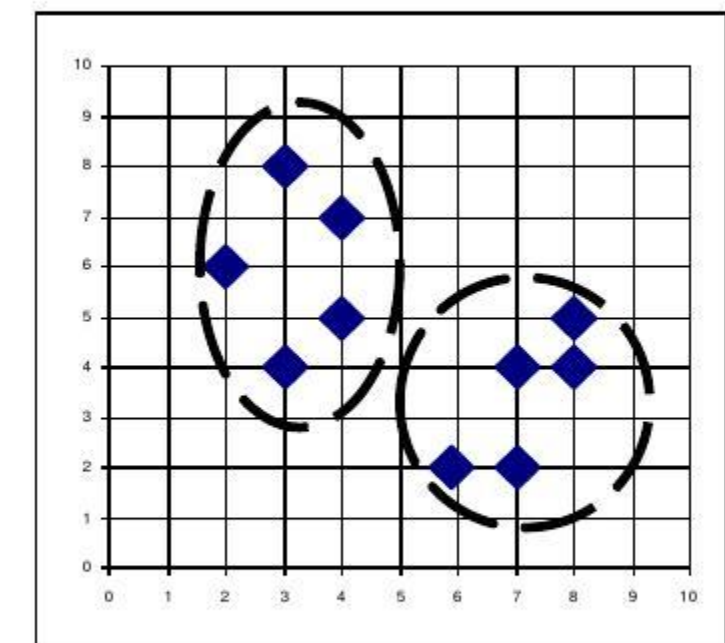
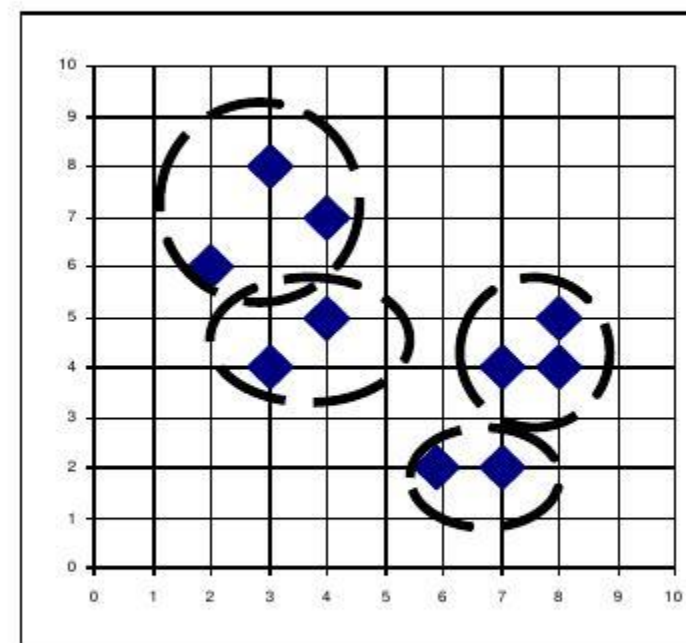
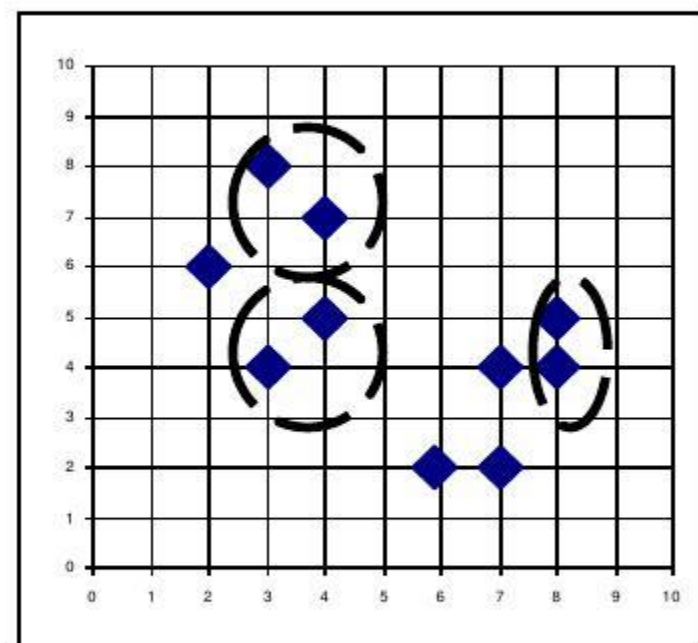




AGNES (Agglomerative Nesting)



- 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



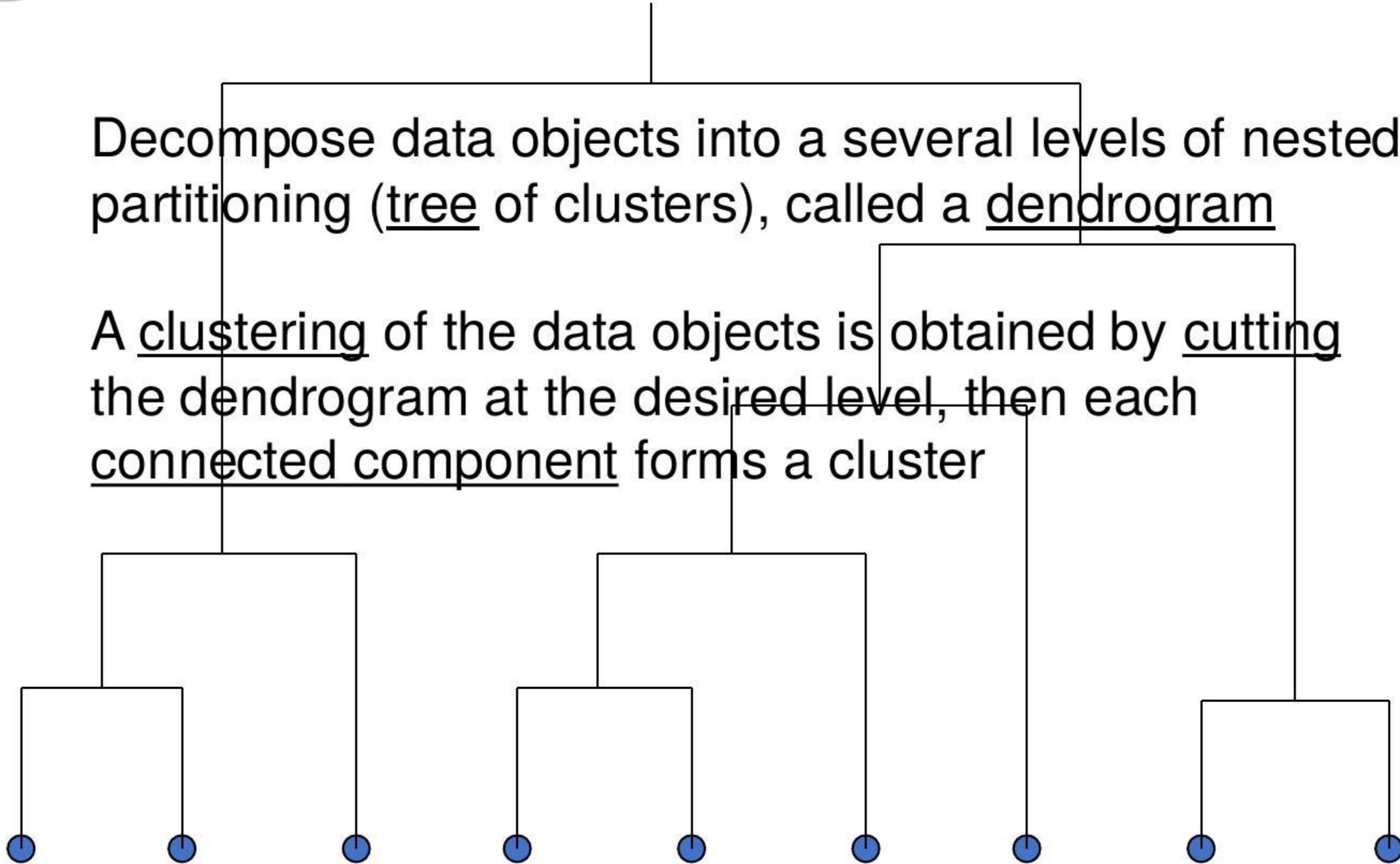


Dendrogram: Shows How Clusters are Merged



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

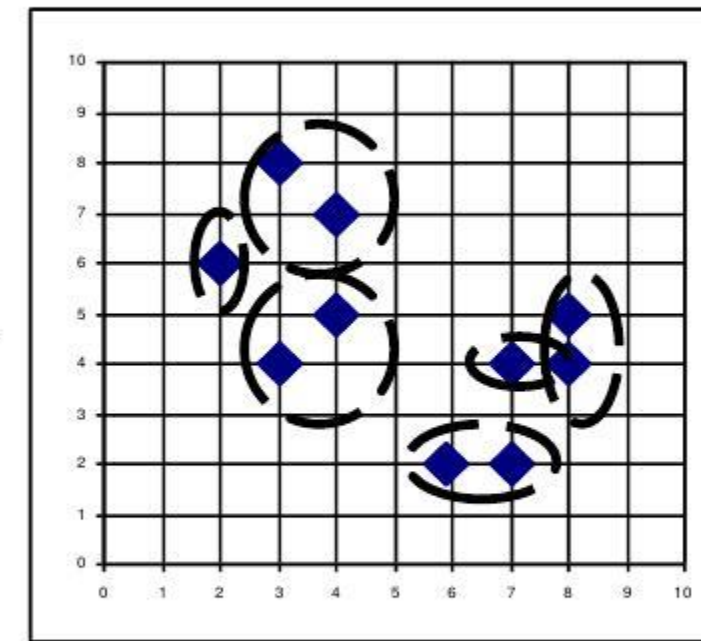
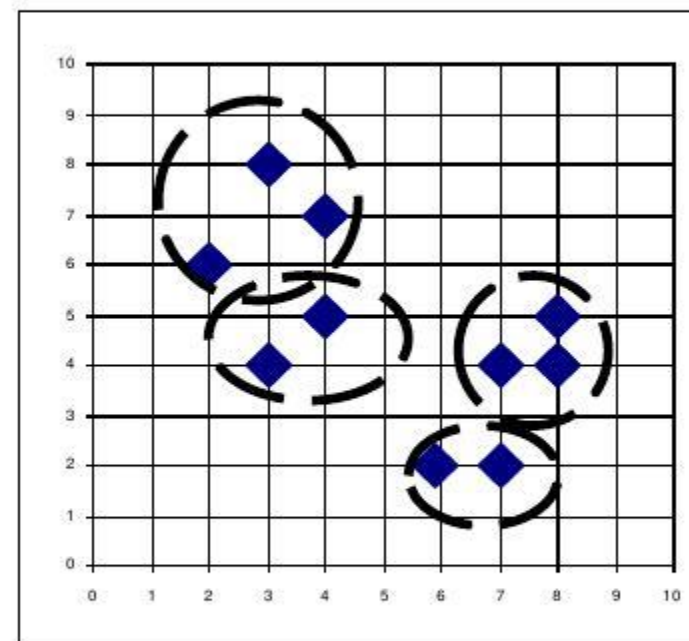
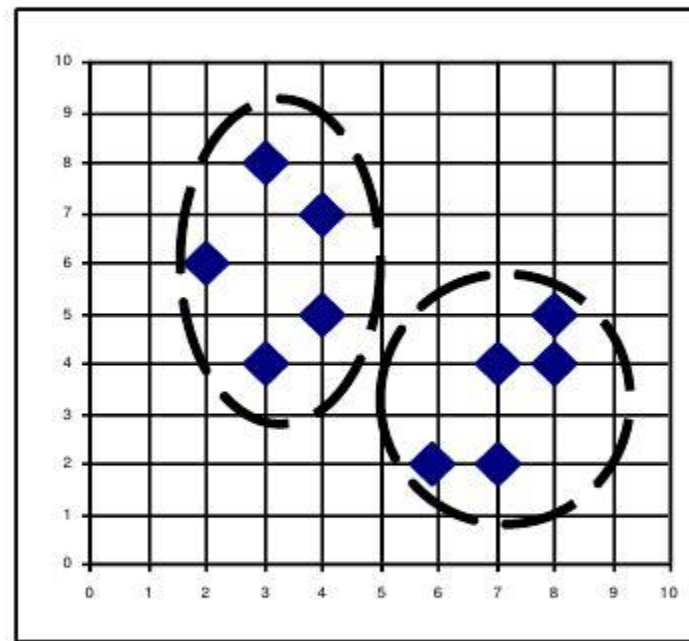




DIANA (Divisive Analysis)

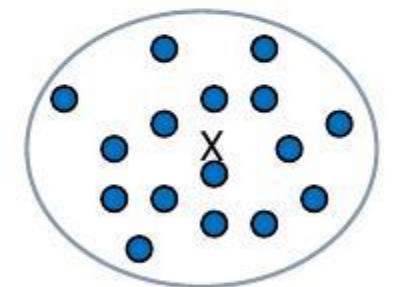
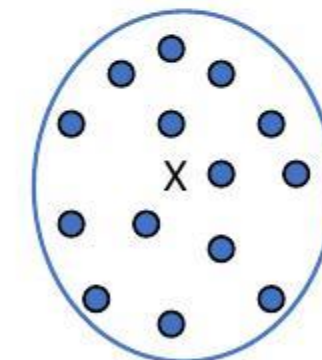


- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own





- Single link: smallest distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \min(t_{ip}, t_{jq})$
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \max(t_{ip}, t_{jq})$
- Average: avg distance between an element in one cluster and an element in the other, i.e., $\text{dist}(K_i, K_j) = \text{avg}(t_{ip}, t_{jq})$
- Centroid: distance between the centroids of two clusters, i.e., $\text{dist}(K_i, K_j) = \text{dist}(C_i, C_j)$
- Medoid: distance between the medoids of two clusters, i.e., $\text{dist}(K_i, K_j) = \text{dist}(M_i, M_j)$
 - Medoid: a chosen, centrally located object in the cluster





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

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





- 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



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



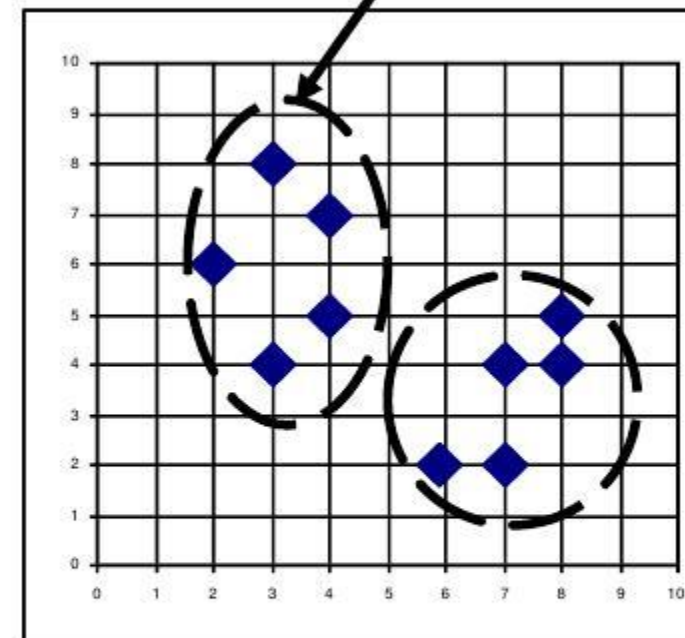
Clustering Feature (CF): $CF = (N, LS, SS)$

N : Number of data points

LS : linear sum of N points: $\sum_{i=1}^N X_i$

SS : square sum of N points

$$\sum_{i=1}^N X_i^2$$



$CF = (5, (16, 30), (54, 190))$

(3,4)

(2,6)

(4,5)

(4,7)

(3,8)





- Clustering feature:
 - Summary of the statistics for a given subcluster: the 0-th, 1st, and 2nd moments of the subcluster from the statistical point of view
 - Registers crucial measurements for computing cluster and utilizes storage efficiently
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
 - A nonleaf node in a tree has descendants or “children”
 - The nonleaf nodes store sums of the CFs of their children
- A CF tree has two parameters
 - Branching factor: max # of children
 - Threshold: max diameter of sub-clusters stored at the leaf nodes



The CF Tree Structure



$B = 7$

$L = 6$

Root

CF_1	CF_2	CF_3		CF_6
child ₁	child ₂	child ₃		child ₆

Non-leaf node

CF_1	CF_2	CF_3		CF_5
child ₁	child ₂	child ₃		child ₅

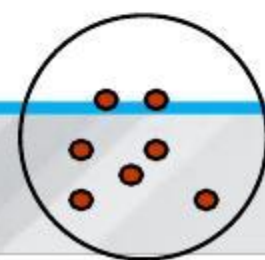
Leaf node

prev	CF_1	CF_2		CF_6	next
------	--------	--------	-------	--------	------

Leaf node

prev	CF_1	CF_2		CF_4	next
------	--------	--------	-------	--------	------

||





- Cluster Diameter

$$\sqrt{\frac{1}{n(n-1)} \sum (x_i - x_j)^2}$$



- For each point in the input
 - Find closest leaf entry
 - Add point to leaf entry and update CF
 - If entry diameter > max_diameter, then split leaf, and possibly parents
- Algorithm is O(n)
- Concerns
 - Sensitive to insertion order of data points
 - Since we fix the size of leaf nodes, so clusters may not be so natural
 - Clusters tend to be spherical given the radius and diameter measures





Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Evaluation of Clustering
- Summary



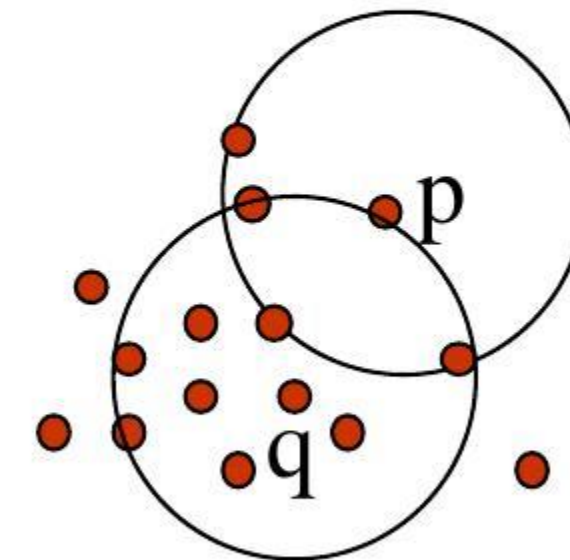


- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
 - Discover clusters of arbitrary shape
 - Handle noise
 - One scan
 - Need density parameters as termination condition
- Several interesting studies:
 - DBSCAN: Ester, et al. (KDD'96)
 - OPTICS: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



- Two parameters:
 - *Eps*: Maximum radius of the neighbourhood
 - *MinPts*: Minimum number of points in an *Eps*-neighbourhood of that point
- $N_{Eps}(p)$: $\{q \text{ belongs to } D \mid \text{dist}(p, q) \leq Eps\}$
- **Directly density-reachable**: A point p is directly density-reachable from a point q w.r.t. *Eps*, *MinPts* if
 - p belongs to $N_{Eps}(q)$
 - core point condition:

$$|N_{Eps}(q)| \geq MinPts$$

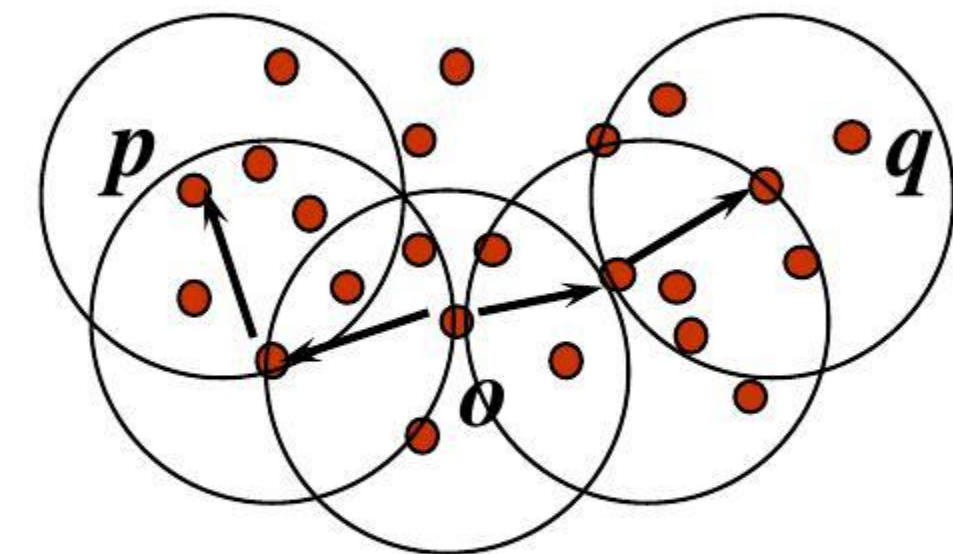
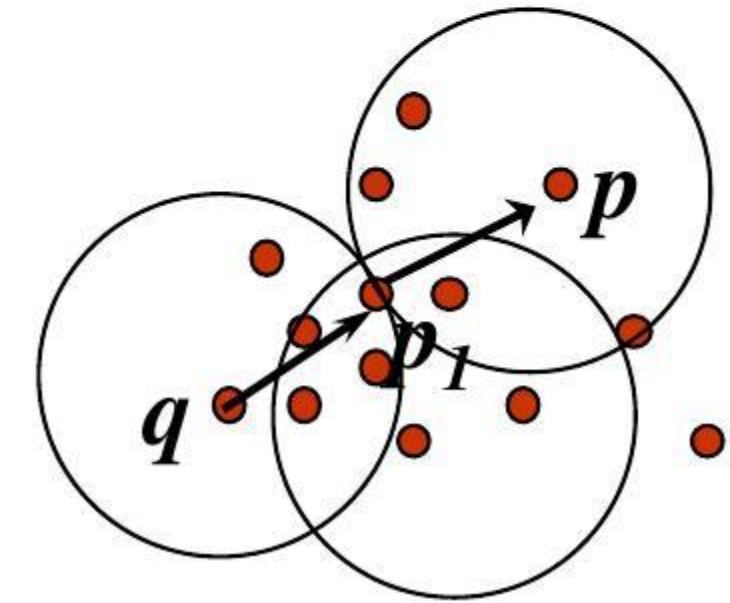


MinPts = 5

Eps = 1 cm

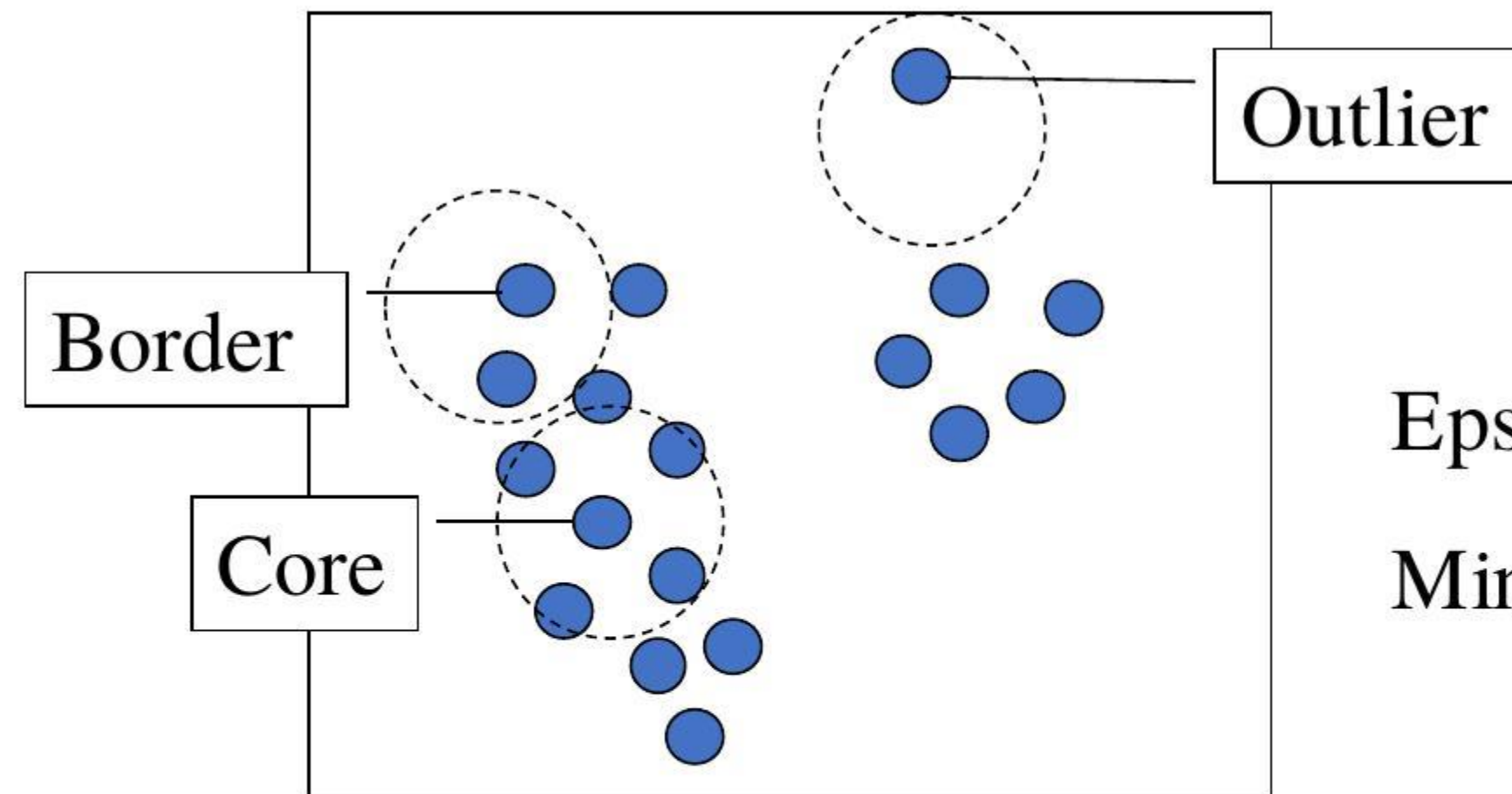


- Density-reachable:
 - A point p is **density-reachable** from a point q w.r.t. Eps , $MinPts$ if there is a chain of points $p_1, \dots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i
- Density-connected
 - A point p is **density-connected** to a point q w.r.t. Eps , $MinPts$ if there is a point o such that both, p and q are density-reachable from o w.r.t. Eps and $MinPts$





- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise



Eps = 1cm

MinPts = 5



- Arbitrary select a point p
- Retrieve all points density-reachable from p w.r.t. Eps and $MinPts$
- If p is a core point, a cluster is formed
- If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database
- Continue the process until all of the points have been processed



DBSCAN: Sensitive to Parameters

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

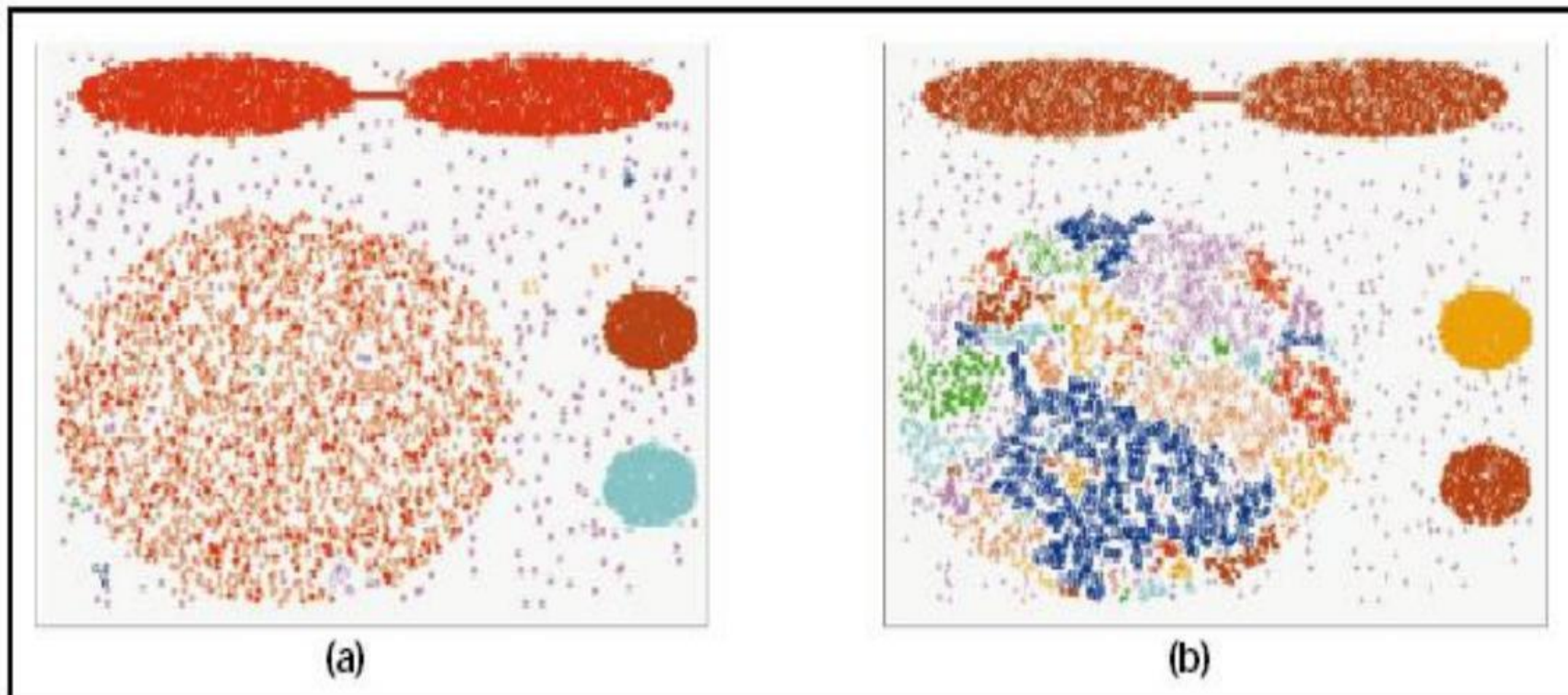
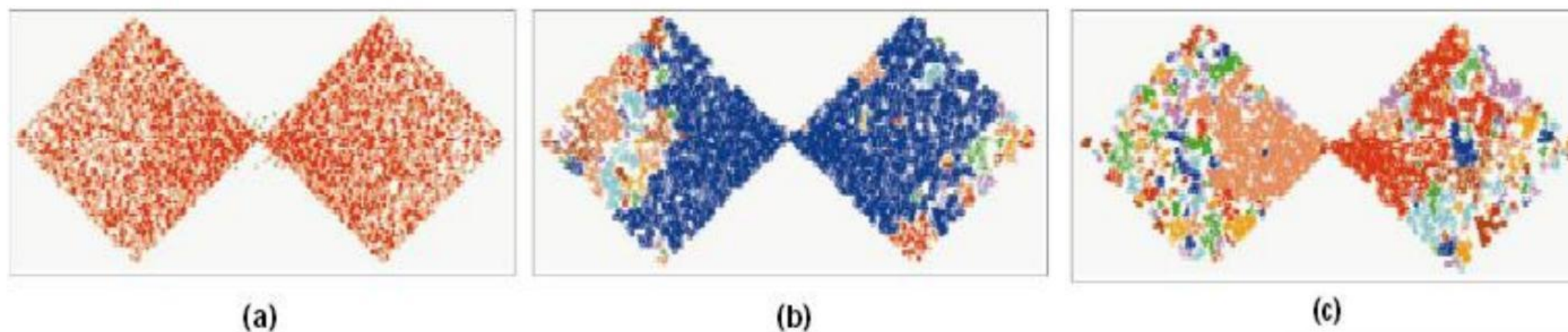


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.





Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Evaluation of Clustering 
- Summary





- Assess if non-random structure exists in the data by measuring the probability that the data is generated by a uniform data distribution
- Test spatial randomness by statistic test: Hopkins Static
 - Given a dataset D regarded as a sample of a random variable o , determine how far away o is from being uniformly distributed in the data space
 - Sample n points, $p_1, \dots, p_n$, uniformly from D . For each p_i , find its nearest neighbor in D : $x_i = \min\{\text{dist}(p_i, v)\}$ where v in D
 - Sample n points, $q_1, \dots, q_n$, uniformly from D . For each q_i , find its nearest neighbor in $D - \{q_i\}$: $y_i = \min\{\text{dist}(q_i, v)\}$ where v in D and $v \neq q_i$
 - Calculate the Hopkins Statistic:
$$H = \frac{\sum_{i=1}^n y_i}{\sum_{i=1}^n x_i + \sum_{i=1}^n y_i}$$
 - If D is uniformly distributed, $\sum x_i$ and $\sum y_i$ will be close to each other and H is close to 0.5. If D is highly skewed, H is close to 0



Determine the Number of Clusters



- Empirical method
 - # of clusters $\approx \sqrt{n}/2$ for a dataset of n points
- Elbow method
 - Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters
- Cross validation method
 - Divide a given data set into m parts
 - Use $m - 1$ parts to obtain a clustering model
 - Use the remaining part to test the quality of the clustering
 - E.g., 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 the 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



- Two methods: extrinsic vs. intrinsic
- Extrinsic: supervised, i.e., the ground truth is available
 - Compare a clustering against the ground truth using certain clustering quality measure
 - Ex. BCubed precision and recall metrics
- Intrinsic: unsupervised, i.e., the ground truth is unavailable
 - Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
 - Ex. Silhouette coefficient



Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: Basic Concepts
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Evaluation of Clustering
- 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
- **DBSCAN**, **OPTICS**, and **DENCLU** are interesting density-based algorithms
- **STING** and **CLIQUE** are grid-based methods, where CLIQUE is also a subspace clustering algorithm
- Quality of clustering results can be evaluated in various ways

Thank You

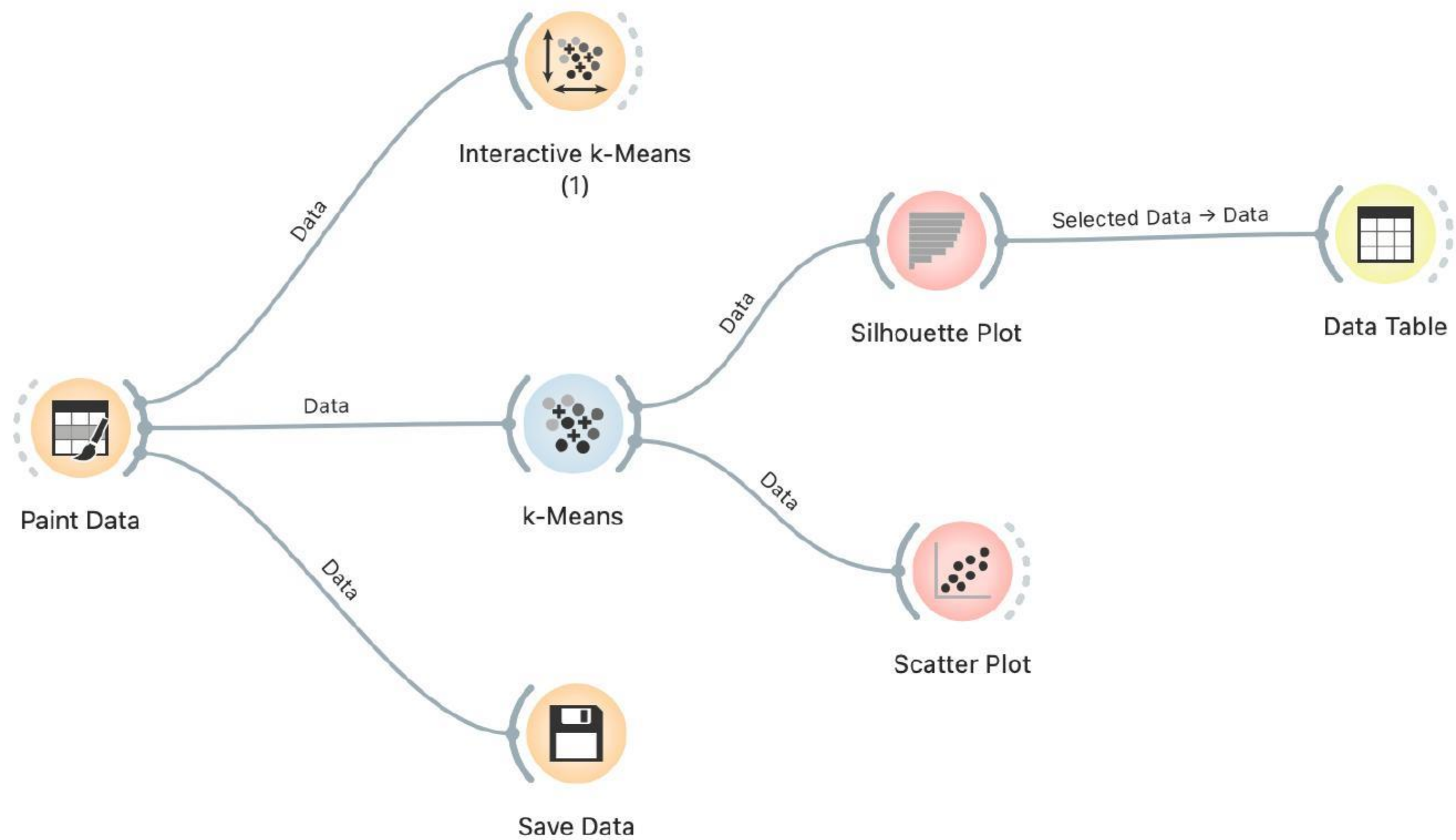
Clustering



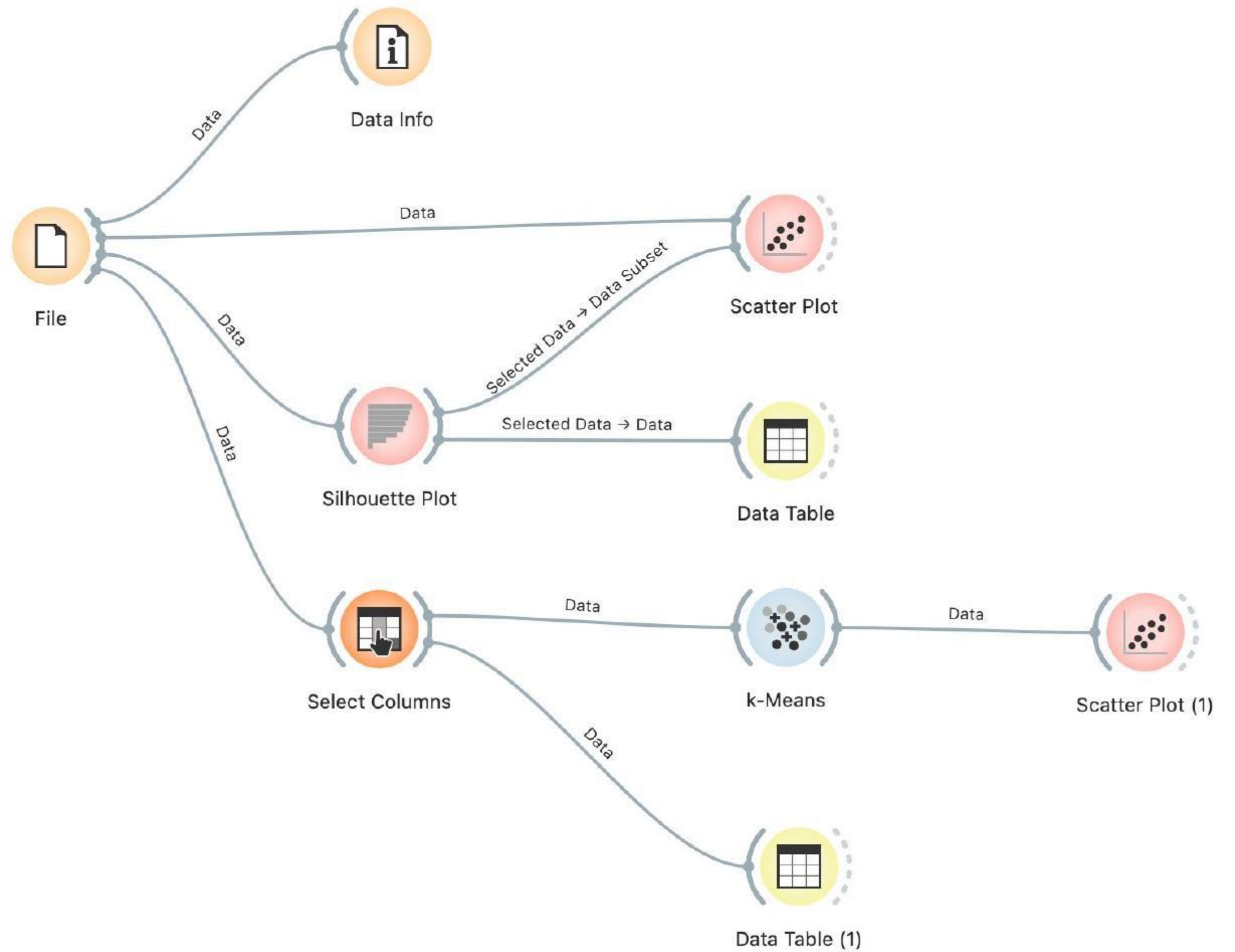
Orange Data Mining

Data Mining Software

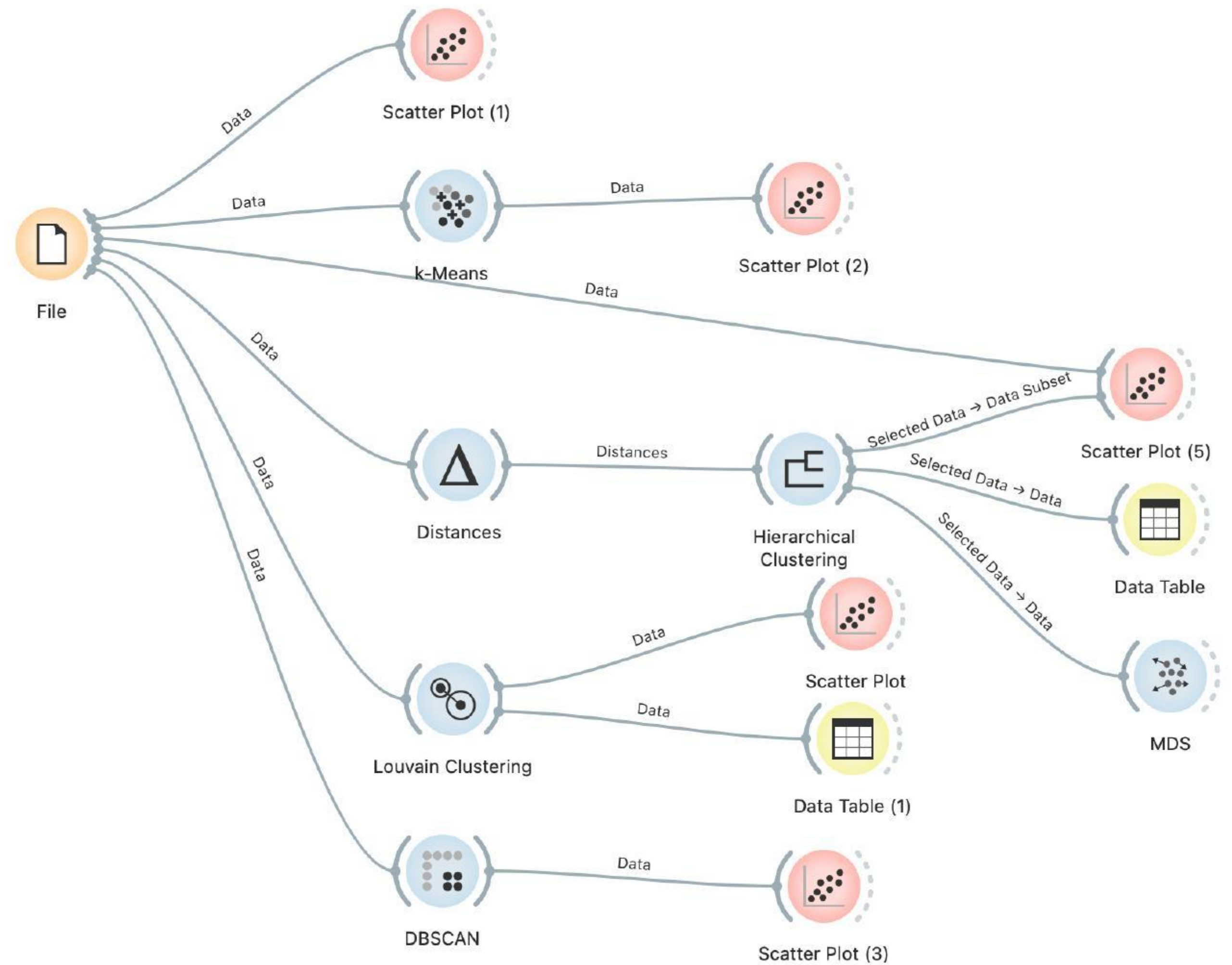
Clustering – Practice 1



Clustering – Practice 2



Clustering – Practice 3



Clustering – Practice 4

