CLUSTERING

What is Clustering

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Grouping a set of data objects into clusters
- Clustering is unsupervised classification:

  no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Examples of Clustering Applications

- **Marketing**: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- **Land use:** Identification of areas of similar land use in an earth observation database
- **Insurance**: Identifying groups of motor insurance policy holders with a high average claim cost
- **Urban planning**: Identifying groups of houses according to their house type, value, and geographical location
- **Seismology**: Observed earth quake epicenters should be clustered along continent faults

What Is a Good Clustering?

- A good clustering method will produce clusters with
  - High intra-class similarity
  - Low inter-class similarity
- Precise definition of clustering quality is difficult
  - Application-dependent
  - Ultimately subjective

Similarity and Dissimilarity Between Objects

- Euclidean distance:

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+\ldots+|x_{i_p}-x_{j_p}|^2)}$$

- Properties of a metric *d(i,j)*:
  - $d(i,j) \geq 0$
  - $d(i,i) = 0$
  - $d(i,j) = d(j,i)$
  - $d(i,j) \leq d(i,k) + d(k,j)$

Major Clustering Approaches

- **Partitioning:** Construct various partitions and then evaluate them by some criterion
- **Hierarchical:** Create a hierarchical decomposition of the set of objects using some criterion
- **Model-based:** Hypothesize a model for each cluster and find best fit of models to data
- **Density-based**: Guided by connectivity and density functions

Partitioning Algorithms

- **Partitioning method:** Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- 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* (MacQueen, 1967): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw, 1987): Each cluster is represented by one of the objects in the cluster

K-Means Clustering

- Given *k*, the *k-means* algorithm consists of four steps:
  - Select initial centroids at random.
  - Assign each object to the cluster with the nearest centroid.
  - Compute each centroid as the mean of the objects assigned to it.
  - Repeat previous 2 steps until no change.

Algorithm Definition

- The K-Means algorithm is an method to cluster objects based on their attributes into k partitions.
- It assumes that the *k* clusters exhibit Gaussian distributions.
- It assumes that the object attributes form a vector space.
- The objective it tries to achieve is to minimize total intra-cluster variance.

Algorithm Fitness Function

- The K-Means algorithm attempts to minimize the squared error for all elements in all clusters.

- The error equation is:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \left| p - m_i \right|^2$$

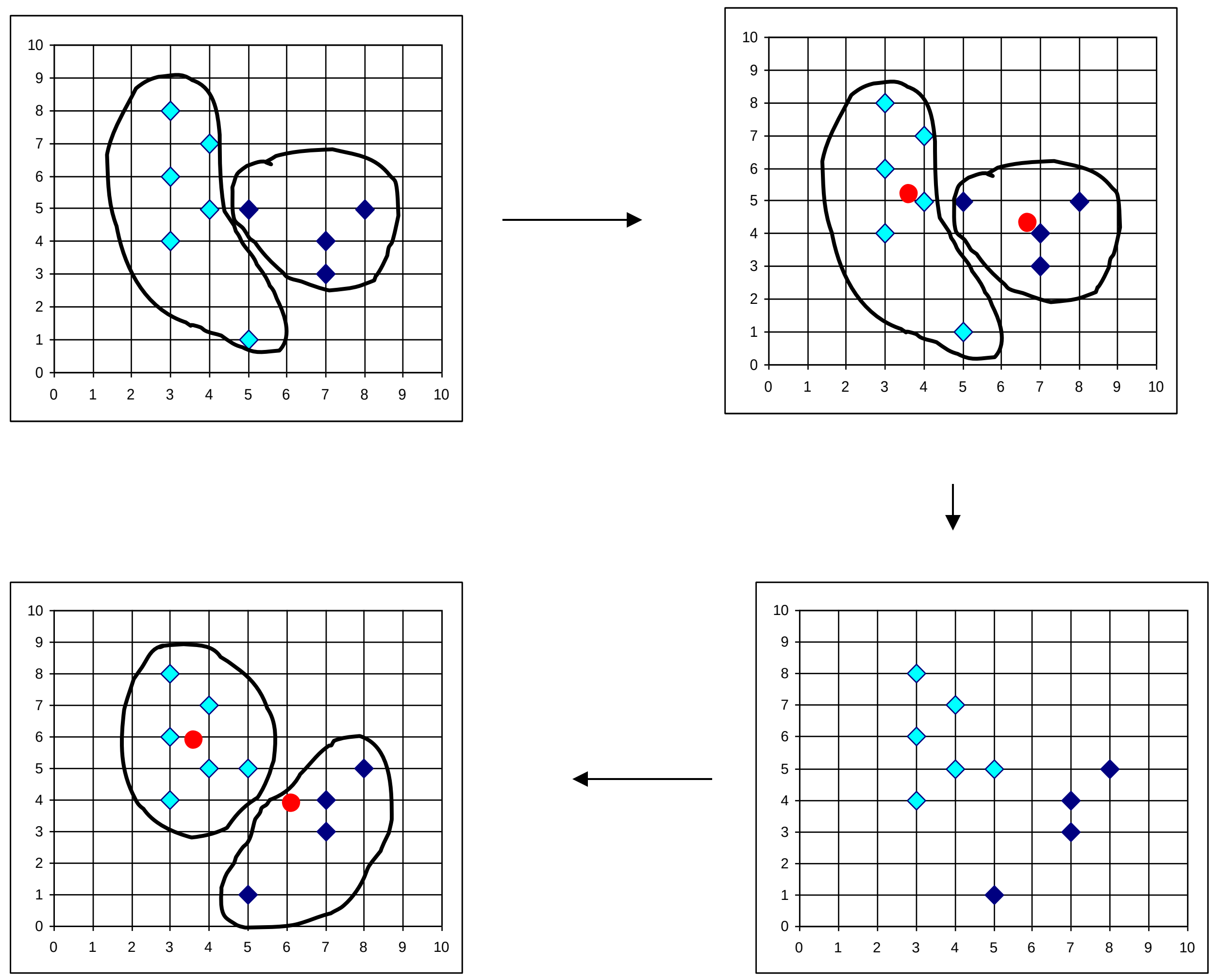
- Where **E** is the sum of the square error for all elements in the data set; **p** is a given element; and $\mathbf{m_i}$ is the mean of cluster $\mathbf{C_i}$

K-Means Algorithm

In case 2D data; E.g.
Cluster ci = {(1, 4), (9, 6), (2, 5)}
Cluster mean mi = (12/3, 15/3)
= (4, 5)

- **Input**
  - k: the number of clusters
  - D: a dataset containing *n* elements
- **Output**: a set of *k* clusters
- **Method**
  - (1) arbitrarily choose *k* elements from *D* as the initial cluster mean values
  - (2) **repeat**
  - (3) assign each element to the cluster whose mean the element is *closest* to
  - (4) once all of the elements are assigned to clusters, calculate the *actual* cluster means
  - (5) **until** there is no change between the new and old cluster means

K-Means Clustering (Example)



Comments on the K-Means Method

- **Strengths**
  - *Relatively efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, k, $t << n$.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as *simulated annealing* and *genetic algorithms*

- **Weaknesses**
  - Applicable only when *mean* is defined (what about categorical data?)
  - Need to specify k, the *number* of clusters, in advance
  - Trouble with noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

K-medoids **Clustering**

- ***K-means*** is appropriate when we can work with Euclidean distances
- Thus, *K*-means can work only with numerical, quantitative variable types
- Euclidean distances do not work well in at least two situations
  - Some variables are categorical
  - Outliers can be potential threats
- A general version of **K-means** algorithm called ***K-medoids*** can work with any distance measure
- ***K-medoids*** clustering is computationally more intensive

K-medoids Algorithm

- **Step 1:** For a given cluster assignment *C*, find the observation in the cluster minimizing the total distance to other points in that cluster:

$$i_k^* = \arg\min_{\{i:C(i)=k\}} \sum_{C(j)=k} d(x_i, x_j).$$

- **Step 2:** Assign $m_k = x_{i_k^*},\ k = 1,2,\ldots,K$

- **Step 3:** Given a set of cluster centers $\{m_1, \ldots, m_K\}$, minimize the total error by assigning each observation to the closest (current) cluster center:

$$C(i) = \arg\min_{1 \le k \le K} d(x_i, m_k),\ i = 1,\ldots,N$$

- Iterate steps 1 to 3

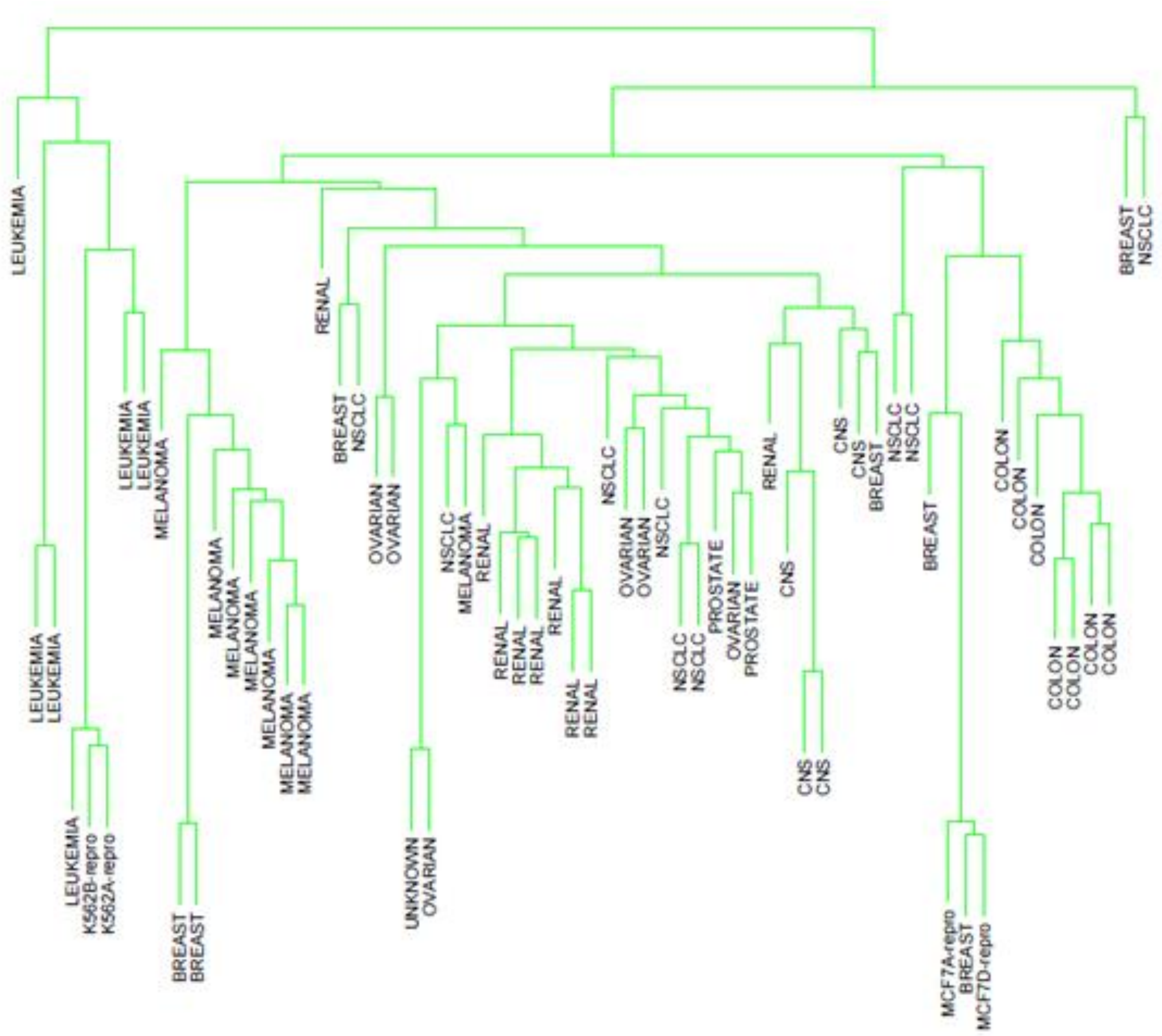
K-medoids **Summary**

- Generalized *K*-means
- Computationally much costlier that *K*-means
- Apply when dealing with categorical data
- Apply when data points are not available, but only pair-wise distances are available
- Converges to local minimum

Hierarchical Clustering

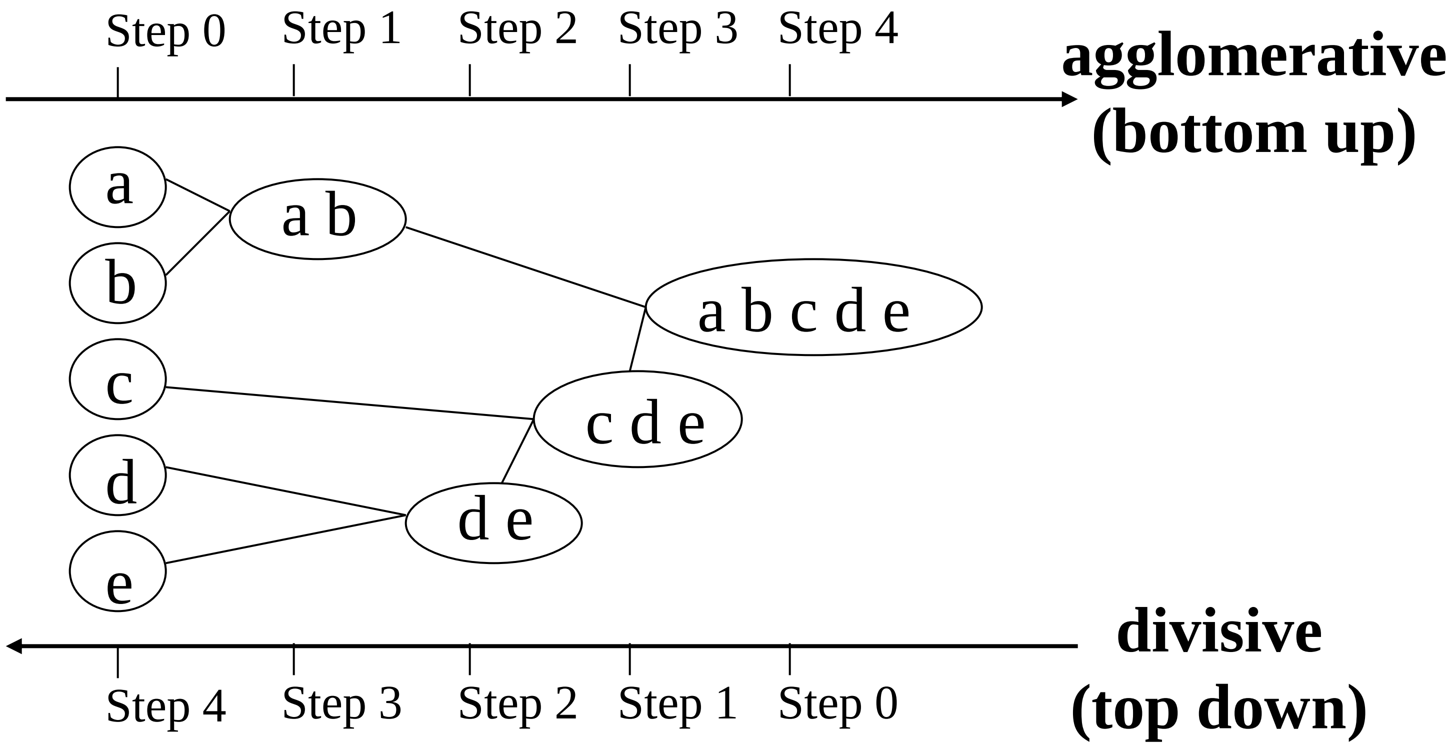
- Two types: (1) agglomerative (bottom up), (2) divisive (top down)
- **Agglomerative**: two groups are merged if distance between them is less than a threshold
- **Divisive**: one group is split into two if intergroup distance more than a threshold
- Can be expressed by an excellent graphical representation called "**dendogram**", when the process is monotonic: dissimilarity between merged clusters is increasing. Agglomerative clustering possesses this property. Not all divisive methods possess this monotonicity.
- Heights of nodes in a **dendogram** are proportional to the threshold value that produced them.

An Example Hierarchical Clustering



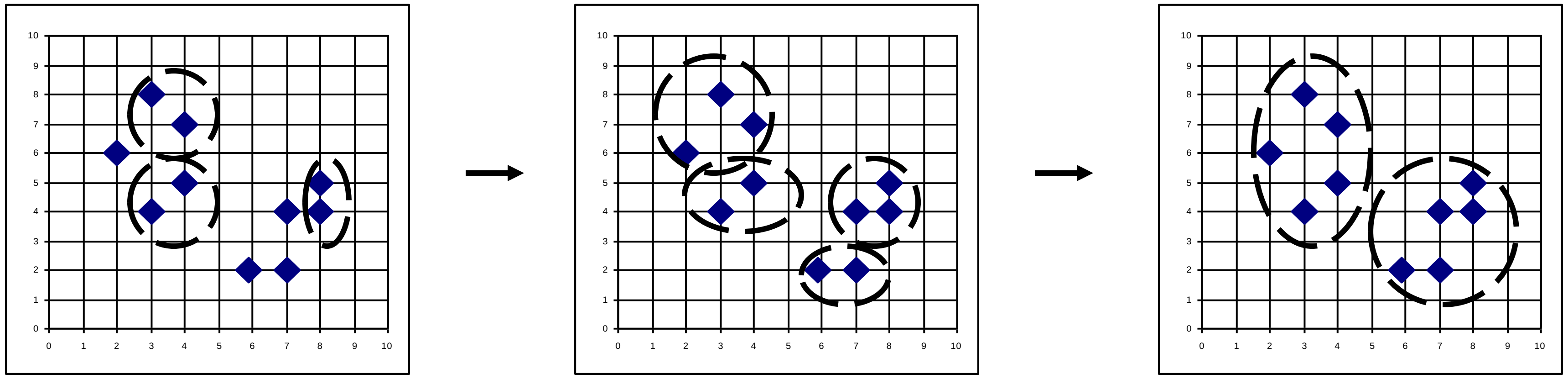
Hierarchical Clustering

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



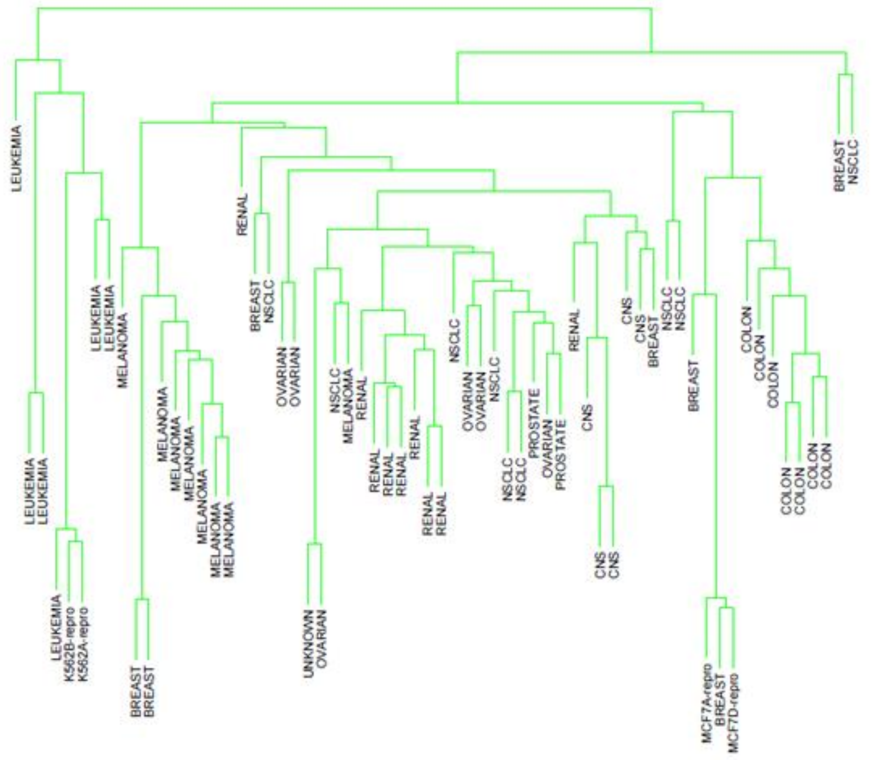
Agglomerative Nesting (Bottom Up)

- Produces tree of clusters (nodes)
- Initially: each object is a cluster (leaf)
- Recursively merges nodes that have the least dissimilarity
- Criteria: min distance, max distance, avg distance, center distance
- Eventually all nodes belong to the same cluster (root)



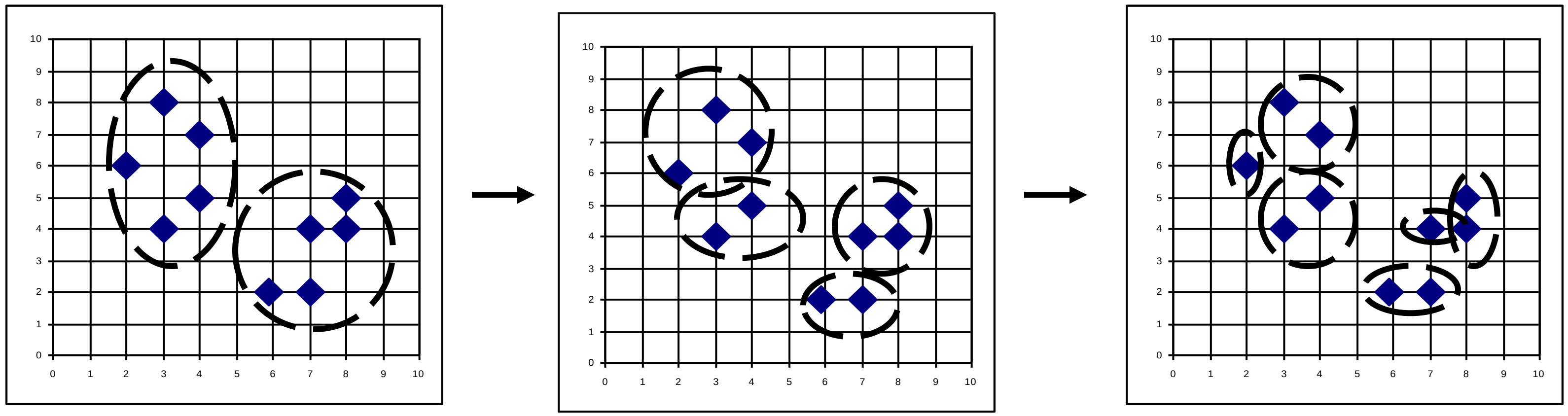
A *Dendrogram* Shows How the Clusters are Merged Hierarchically

- 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.



Divisive Analysis (Top Down)

- Inverse order of **Agglomerative**
- Start with root cluster containing all objects
- Recursively divide into subclusters
- Eventually each cluster contains a single object



Linkage Functions

- We know how to measure the distance between two objects, but defining the distance between an object and a cluster, or defining the distance between two clusters is non obvious.
  - **Single linkage (nearest neighbor):** In this method the distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters. $$d_{SL}(G,H) = \min_{\substack{i \in G \\ j \in H}} d_{ij}$$
  - **Complete linkage (furthest neighbor):** In this method, the distances between clusters are determined by the greatest distance between any two objects in the different clusters (i.e., by the "furthest neighbors"). $$d_{CL}(G,H) = \max_{\substack{i \in G \\ j \in H}} d_{ij}$$
  - **Group average linkage:** In this method, the distance between two clusters is calculated as the average distance between all pairs of objects in the two different clusters. $$d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{j \in H} d_{ij}$$

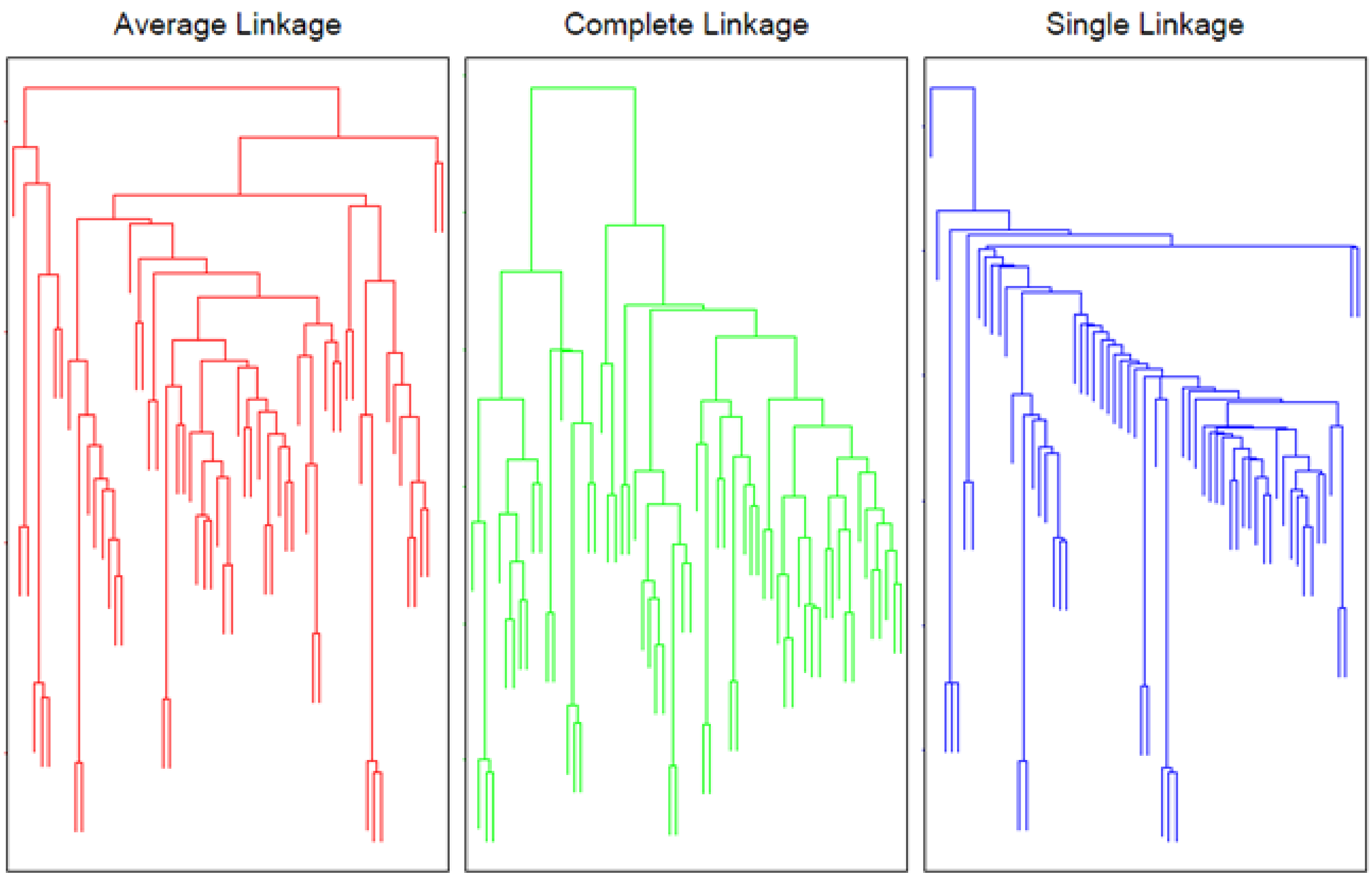
Linkage Functions

- SL considers only a single pair of data points; if this pair is close enough then action is taken. So, SL can form a “chain” by combining relatively far apart data points.
- SL often violates the compactness property of a cluster. SL can produce clusters with large diameters (D_G).

$$D_G = \max_{i \in G,\, j \in G} d_{ij}$$

- CL is just the opposite of SL; it produces many clusters with small diameters.
- CL can violate “closeness” property- two close data points may be assigned to different clusters.
- GA is a compromise between SL and CL

Linkage Functions



Density-Based Clustering Methods

- 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

Model based clustering

- Assume data generated from K probability distributions
- Typically Gaussian distribution Soft or probabilistic version of K-means clustering
- Need to find distribution parameters.
- EM Algorithm