ARTIFICIAL INTELLIGENCE NEEDS REAL INTELLIGENCE

Clustering II

Professor: Tianyuan ZHANG
tianyuan.zhang@kedgebs.com

KEDGE BUSINESS SCHOOL

Recap of Previous Session

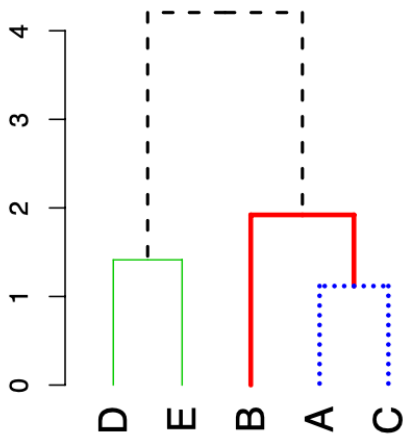
- Unsupervised Learning
  - Learn hidden patterns or structures exclusively from unlabeled data
  - Three types of common unsupervised learning algorithms
    - Clustering:
      - Natural groups → Clusters
    - Association rule mining:
      - Frequent co-occurrence → Association rules
    - Dimensionality reduction:
      - Simplifications → Compressed data

Recap of Previous Session

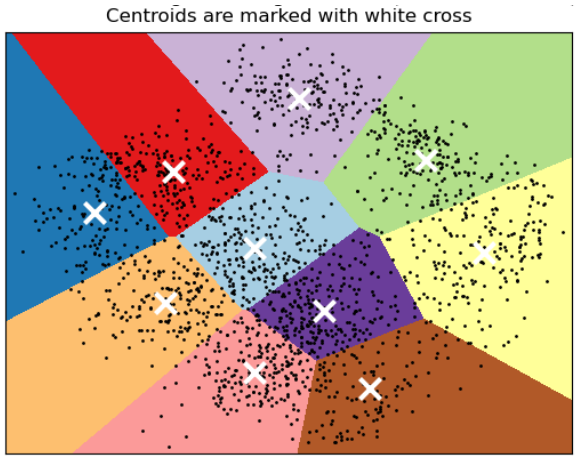
- Clustering
  - Grouping unlabeled data into different natural clusters
  - Data points within the same cluster are similar.
  - Data points in different clusters are different.
  - The clusters are not pre-defined by humans before clustering
  - We need to interpret the clustering results after clustering

Recap of Previous Session

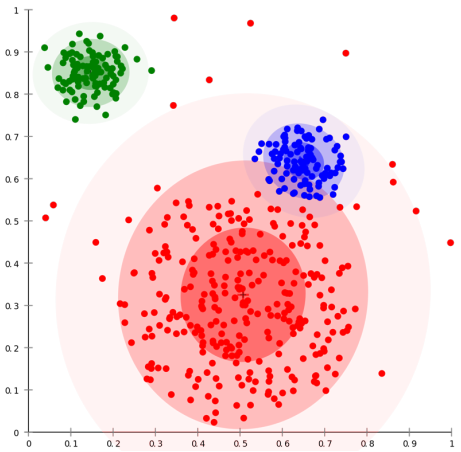
- Different types of clustering algorithms



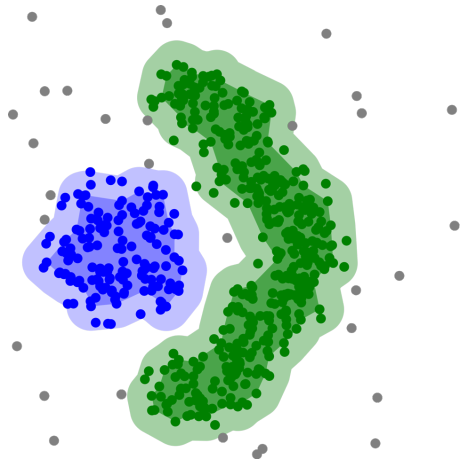
Connectivity-based
Hierarchical clustering



Centroid-based clustering
Partitioning method



Distribution-based
clustering



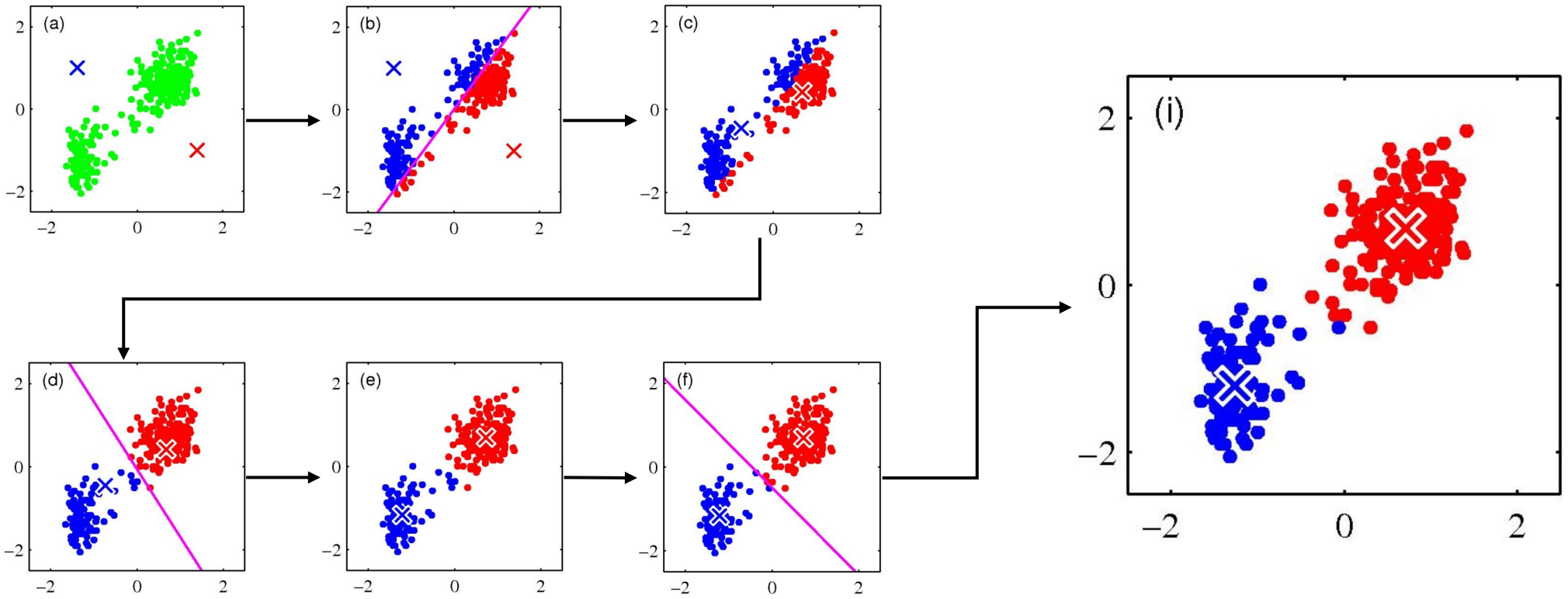
Density-based
clustering

Recap of Previous Session

- K-Means Clustering
  - An iterative centroid-based clustering algorithm
  - A partitioning method that divides the feature space into K distinct, non-overlapping subsets, so-called clusters
  - Each cluster is represented by a centroid
    - Which cluster a point belongs to depends on which centroid it is closest to
  - Steps
    1. Initialize K centroids randomly
    2. Assign each data point to the nearest centroid, forming K clusters
    3. Recalculate the centroids as the mean of all points in each cluster
    4. Repeat steps 2 and 3 until convergence

Recap of Previous Session

- K-Means Clustering



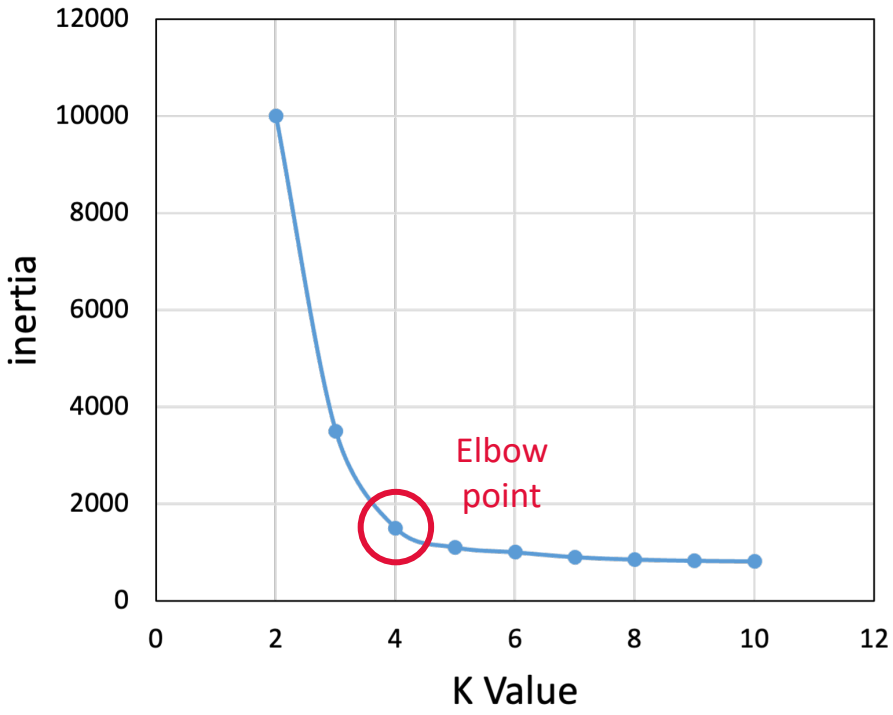
Recap of Previous Session

- K-Means Clustering
  - Requirements
    - The clusters need to be convex and isotropic.
    - The clusters should have equal variance.
    - The clusters should have similar size.
    - Perform feature scaling before using K-Means algorithm.
    - Select a proper K value

Recap of Previous Session

- K-Means Clustering
  - Select optimal K value
    - Elbow methods
    - inertia = $\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$
    - If there is no clear elbow point, use other metrics
      - Silhouette Coefficient
      - Calinski-Harabasz Index
      - Davies-Bouldin Index

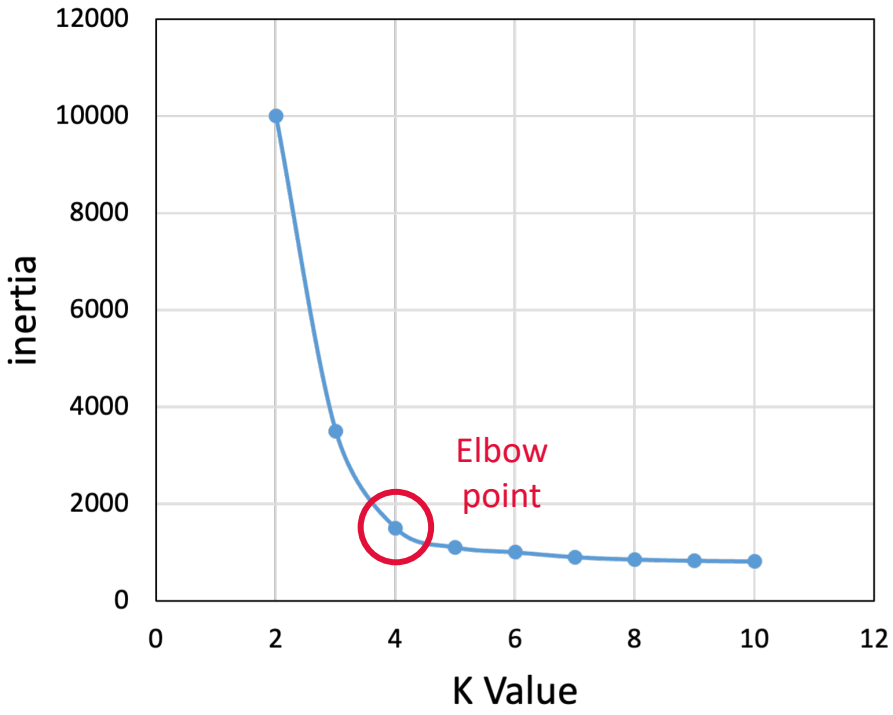
Higher value is better



Recap of Previous Session

- K-Means Clustering
  - Select optimal K value
    - Elbow methods
    - inertia = $\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$
    - If there is no clear elbow point, use other metrics
      - Silhouette Coefficient
      - Calinski-Harabasz Index
      - Davies-Bouldin Index

Lower value is better

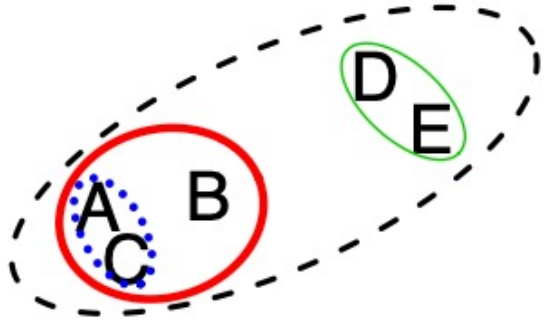


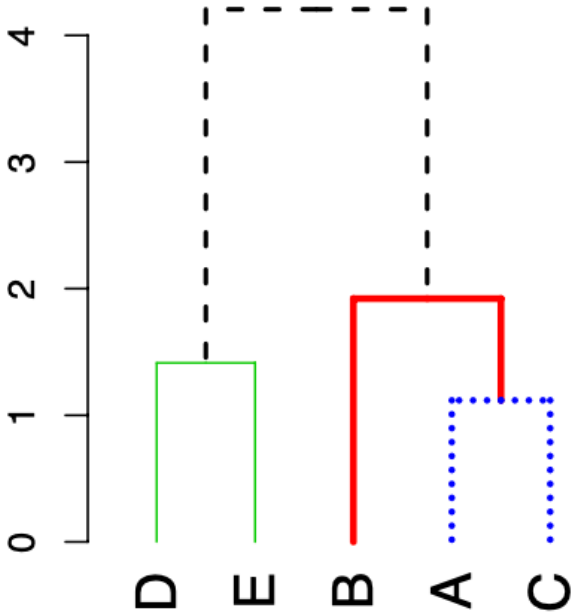
Outline

- Hierarchical Clustering

Hierarchical Clustering

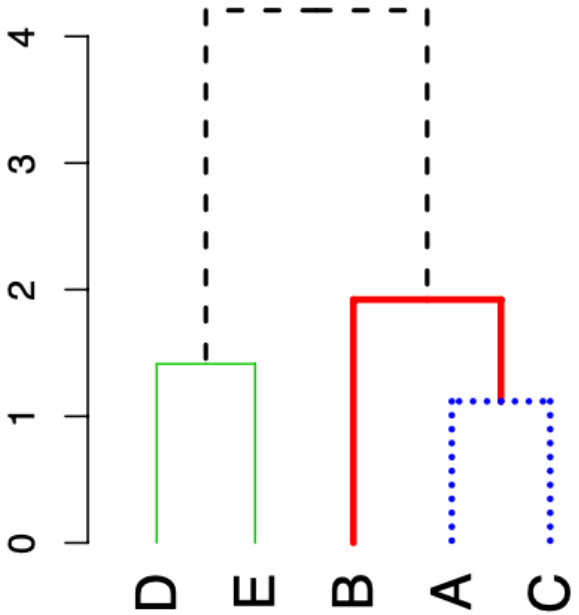
- Connectivity-based clustering algorithm
- Cluster is defined as sets of points that are closely connected to each other
- Build nested clusters by merging or splitting them successively
- Result is a hierarchy of clusters, represented by a dendrogram or a tree





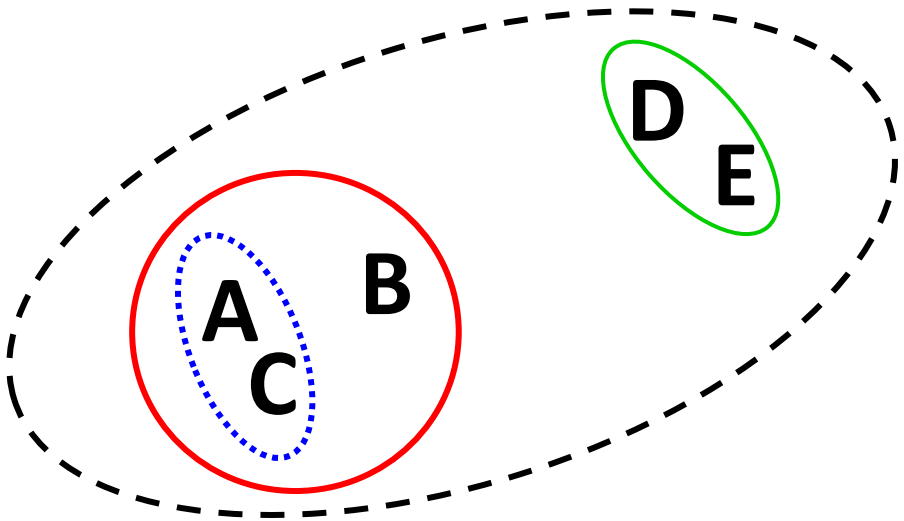
Hierarchical Clustering

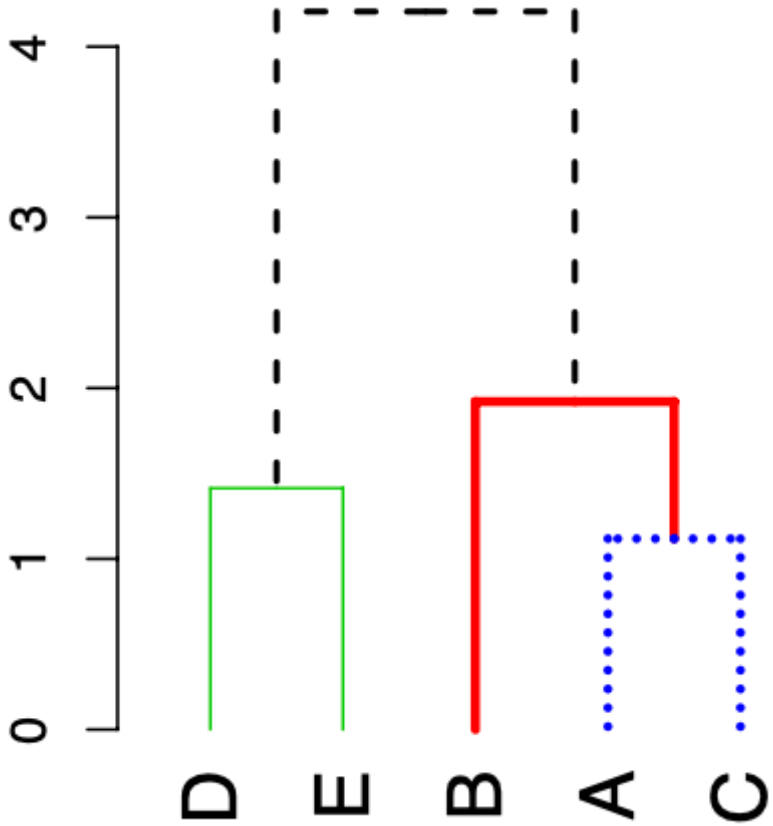
- Two types of hierarchical clustering algorithm
  - Agglomerative clustering
    - Bottom-up approach
    - Every point starts in its own cluster
    - Pairs of clusters are merged as one moves up the hierarchy
  - Divisive clustering
    - Top-down approach
    - All point start in one cluster
    - Splits are performed recursively as one moves down the hierarchy
  - We will focus on agglomerative clustering only



Hierarchical Clustering

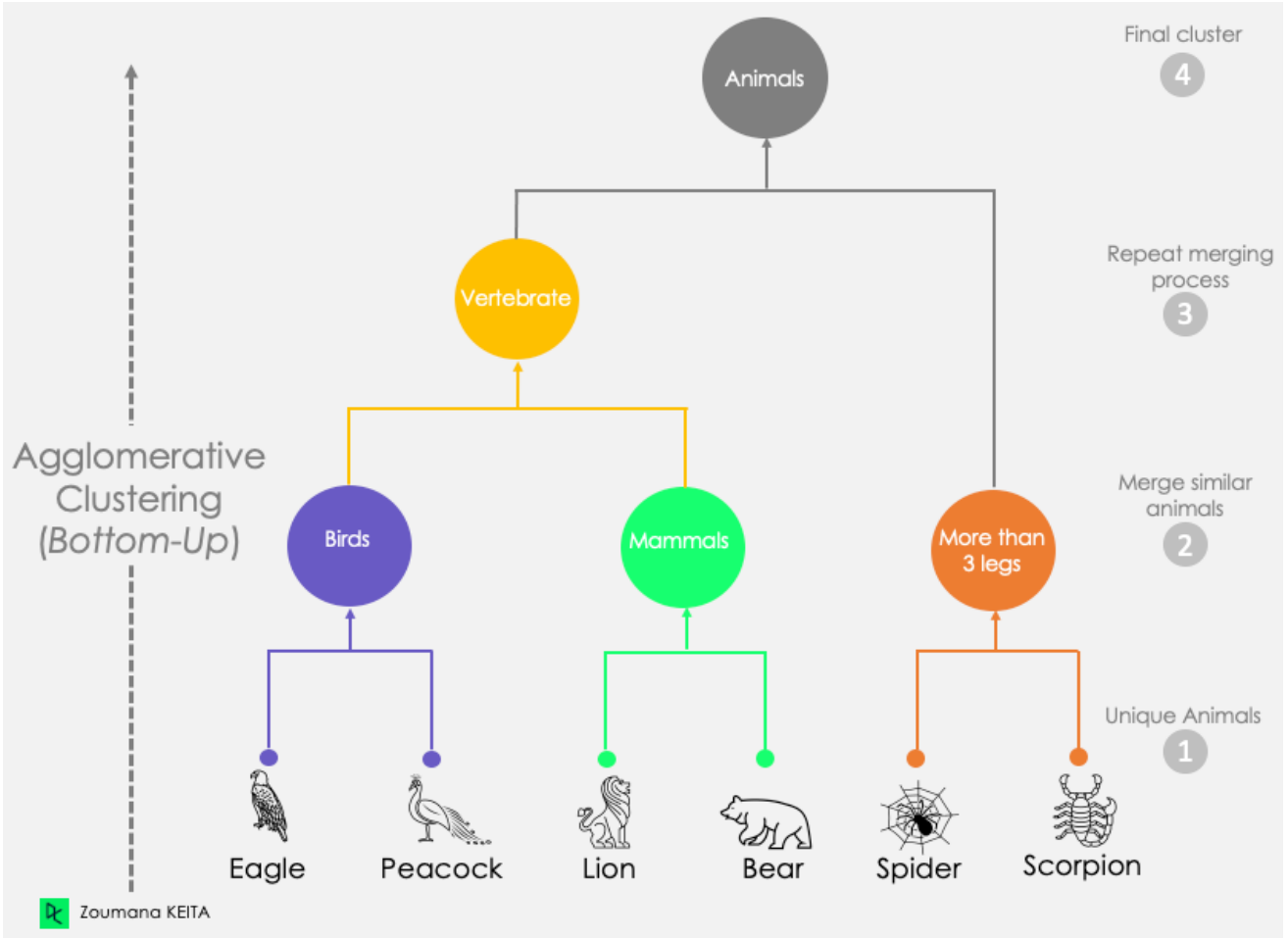
- Agglomerative clustering
  1. Start with each point in its own cluster
  2. Identify the two closest clusters. Merge them.
  3. Repeat until all points are in a single cluster.





Hierarchical Clustering

- Agglomerative clustering



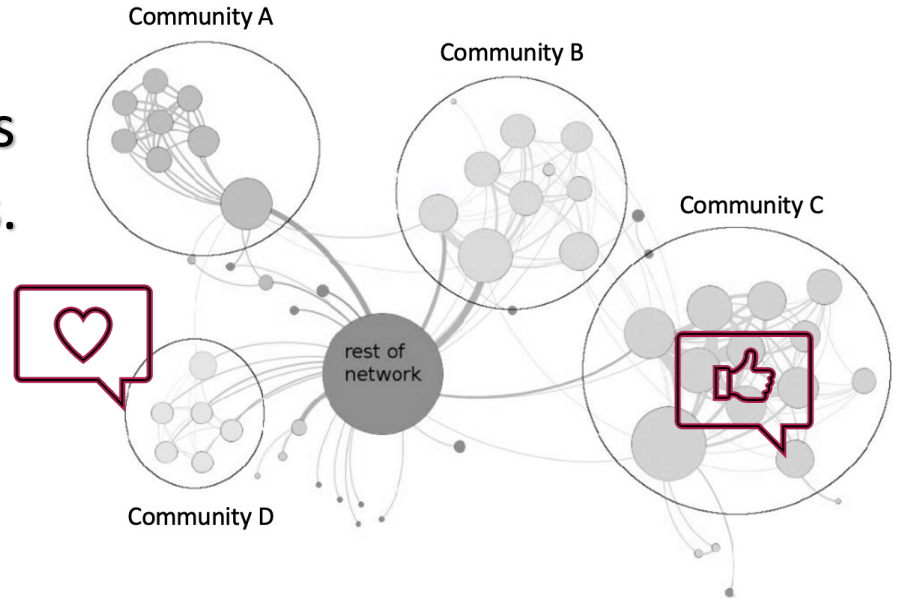
Hierarchical Clustering

- Agglomerative clustering is based on distance
  - Close points are similar to each other → Small distance
  - Distance represents the dissimilarity between points
  - Distance metrics
    - Euclidean distance (default in sklearn)
    - Other options

metric	Function
'cityblock'	metrics.pairwise.manhattan_distances
'cosine'	metrics.pairwise.cosine_distances
'euclidean'	metrics.pairwise.euclidean_distances
'haversine'	metrics.pairwise.haversine_distances
'l1'	metrics.pairwise.manhattan_distances
'l2'	metrics.pairwise.euclidean_distances
'manhattan'	metrics.pairwise.manhattan_distances
'nan_euclidean'	metrics.pairwise.nan_euclidean_distances

Hierarchical Clustering

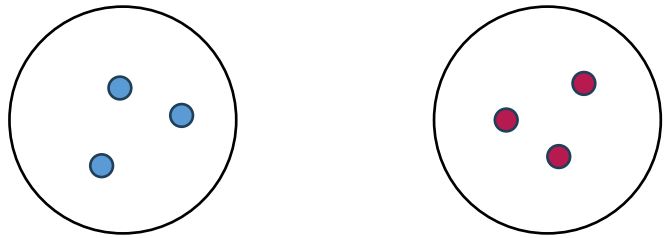
- Agglomerative clustering is based on distance
  - Distance represents the dissimilarity between points
  - Some special dataset needs special distance metrics.
    - For example, if we have a social media network dataset records the interactions between different users.
    - The more frequently two users interact, the closer we assume them to be.



$$distance = \frac{1}{Frequency\ of\ interaction}$$

Hierarchical Clustering

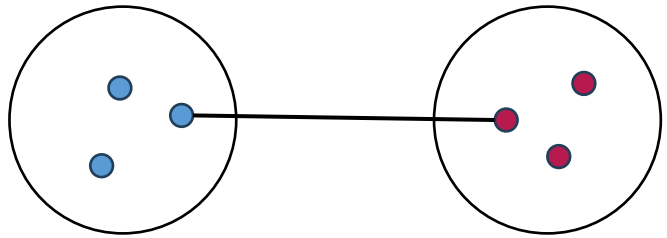
- How to calculate the distance between two clusters?
- The distance (or dissimilarity) between two clusters is called the linkage.
- Linkage
  - A dissimilarity measure
  - Given two two clusters, a linkage tells us how different the points in these clusters are



Cluster S ***Cluster T***

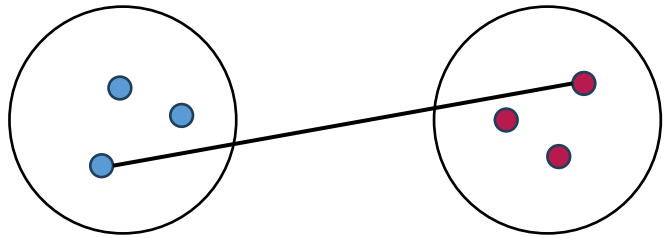
Hierarchical Clustering

- Common linkage types
  - Single linkage
    - Minimal distance between the closest points of the pair of clusters
    - $d(S,T) = \min\{d(x,y): x \in S, y \in T\}$
    - $d(x,y)$ is the distance between elements x and y



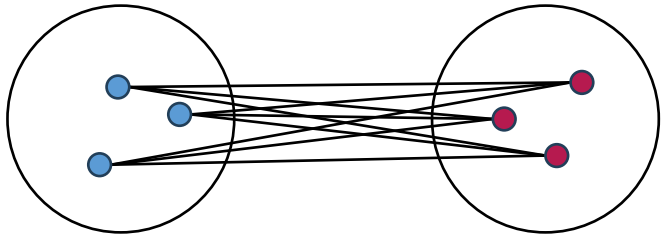
Hierarchical Clustering

- Common linkage types
  - Single linkage
  - Complete linkage
    - Longest distance between any two points in the clusters
    - $d(S,T) = \max\{d(x,y): x \in S, y \in T\}$
    - $d(x,y)$ is the distance between elements x and y



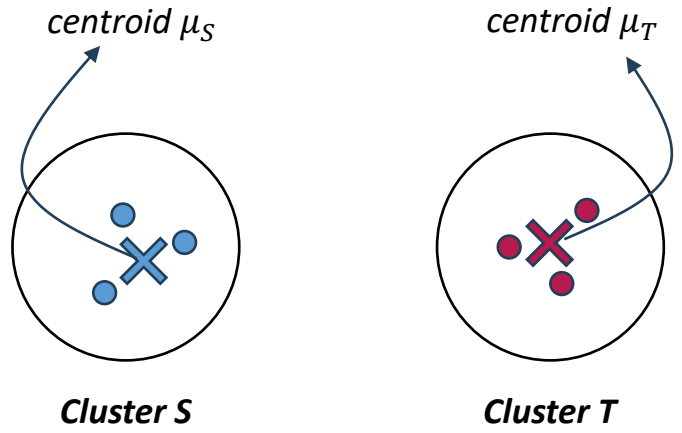
Hierarchical Clustering

- Common linkage types
  - Single linkage
  - Complete linkage
  - Average linkage
    - Average distance between all pairs of points in the two clusters
    - $d(S,T) = \frac{1}{|S|\times|T|}\sum_{x\in S}\sum_{y\in T} d(x,y)$
    - $d(x,y)$ is the distance between elements x and y
    - $|S|$ and $|T|$ are the sizes of clusters S and T



Hierarchical Clustering

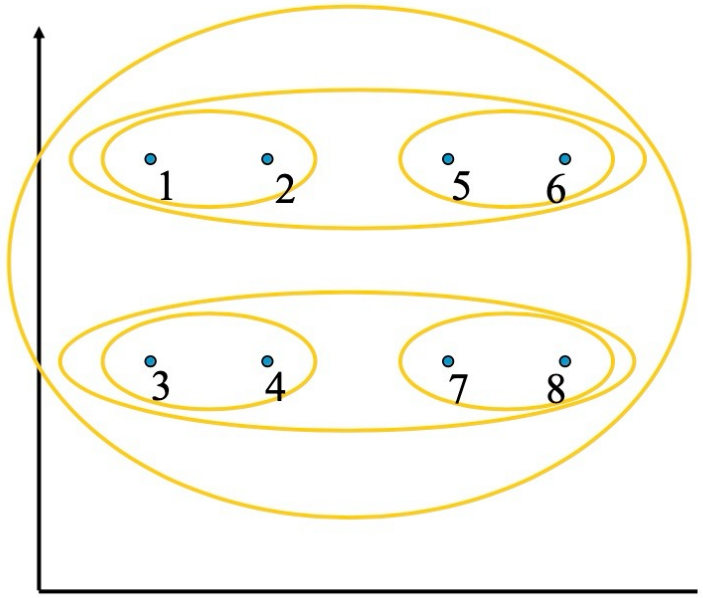
- Common linkage types
  - Single linkage
  - Complete linkage
  - Average linkage
  - Ward linkage
    - The closest pair of clusters should have minimal increase in total within-cluster variance after merging the two



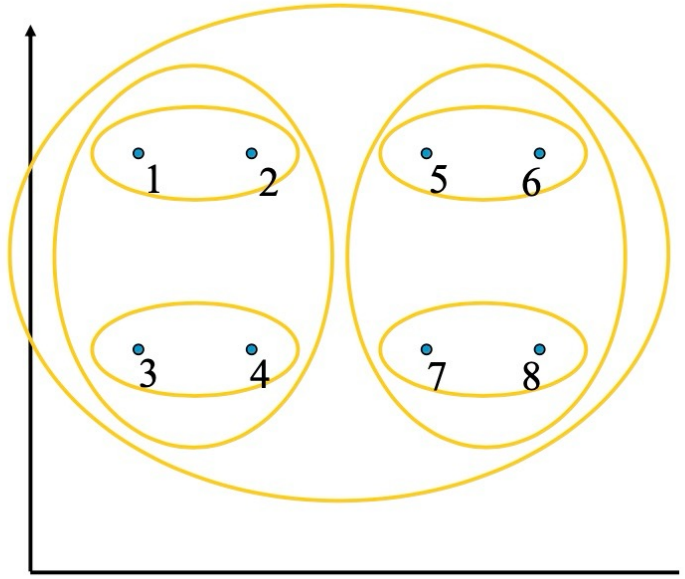
$$d(S,T) = \underbrace{\sum_{x \in S \cup T} \|x - \mu_{S \cup T}\|^2}_{\text{within-cluster variance after merging cluster S and T}} - \underbrace{\sum_{x \in S} \|x - \mu_S\|^2}_{\text{within-cluster variance of cluster S}} - \underbrace{\sum_{x \in T} \|x - \mu_T\|^2}_{\text{within-cluster variance of cluster T}}$$

Hierarchical Clustering

- Different types of linkage will lead to different results.



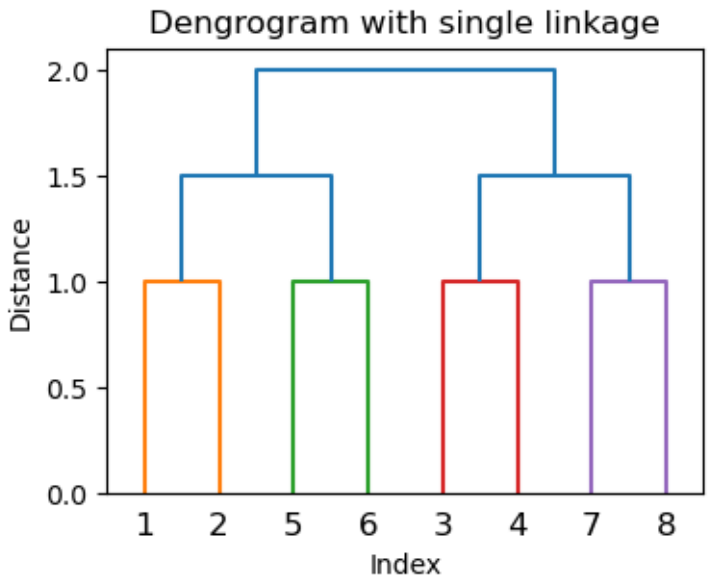
Single linkage

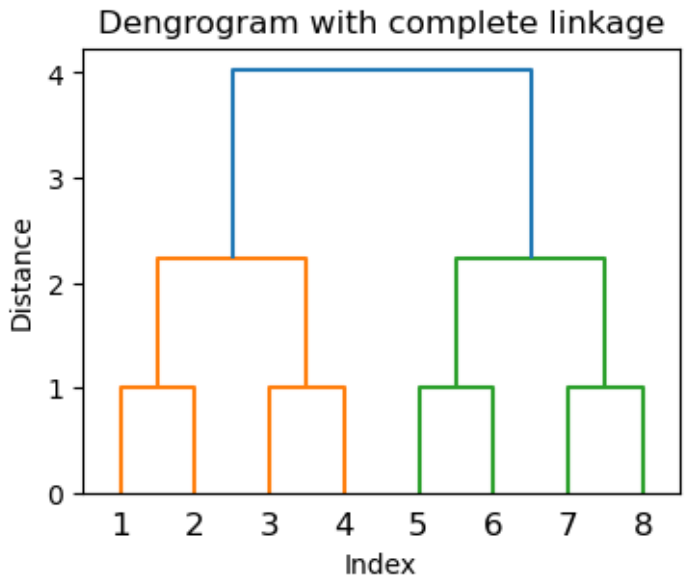


Complete linkage

Hierarchical Clustering

- Different types of linkage will lead to different results.





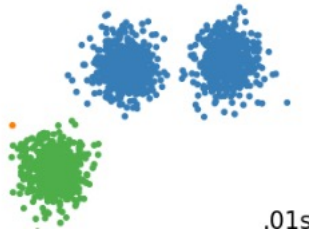
Hierarchical Clustering

- Different types of linkage will lead to different results.
  - Single linkage
    - Tends to produce long clusters as chains.
    - Can handle non-elliptical shapes.
    - Have problems when clusters are close to each other

Single Linkage

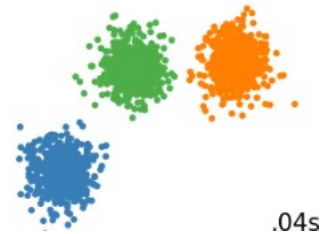
vs

Complete Linkage



Single Linkage

vs



Complete Linkage

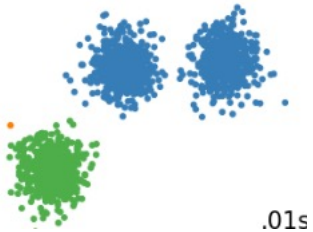
Hierarchical Clustering

- Different types of linkage will lead to different results.
  - Complete linkage
    - Tends to produce compact, tightly bound clusters.
    - Struggle with elongated shapes.
    - Can handle compact clusters that are close to each other.

Single Linkage

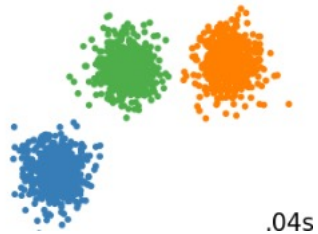
vs

Complete Linkage



Single Linkage

vs

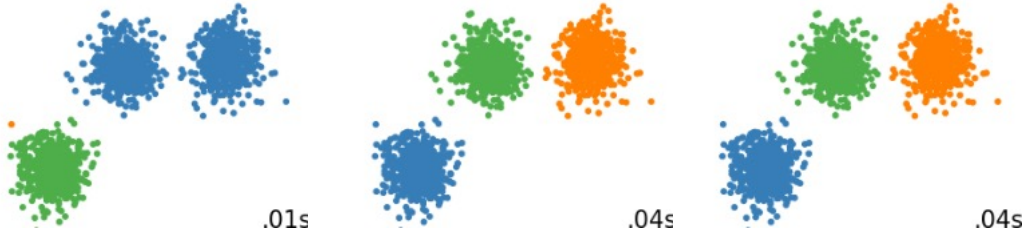


Complete Linkage

Hierarchical Clustering

- Different types of linkage will lead to different results.
  - Average linkage
    - Strikes a balance between single and complete linkage
    - Sometime can produce more natural clusters, but not always
    - Good to use when there is no prior knowledge about the distribution of the data

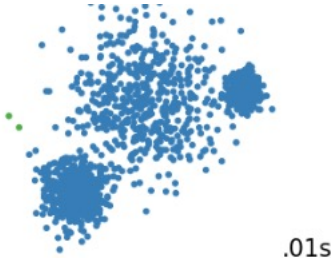
Single Linkage *Average Linkage* *Complete Linkage*



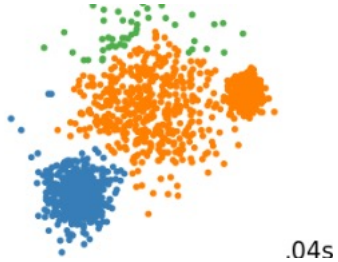
Single Linkage *Average Linkage* *Complete Linkage*

Hierarchical Clustering

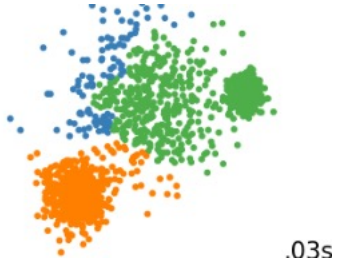
- Different types of linkage will lead to different results.
  - Ward linkage
    - Minimize the within-cluster variance
    - Tend to produce spherical clusters.
    - Can be more computationally intensive.



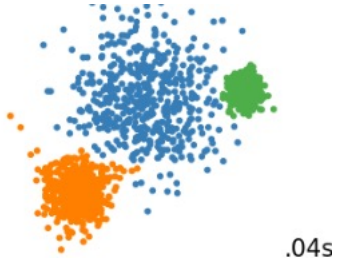
Single Linkage



Average Linkage



Complete Linkage



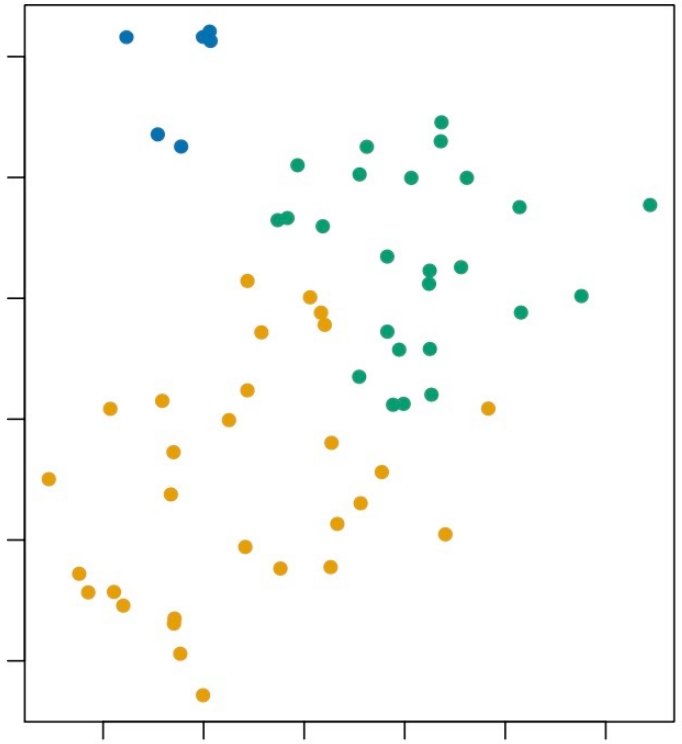
Ward Linkage

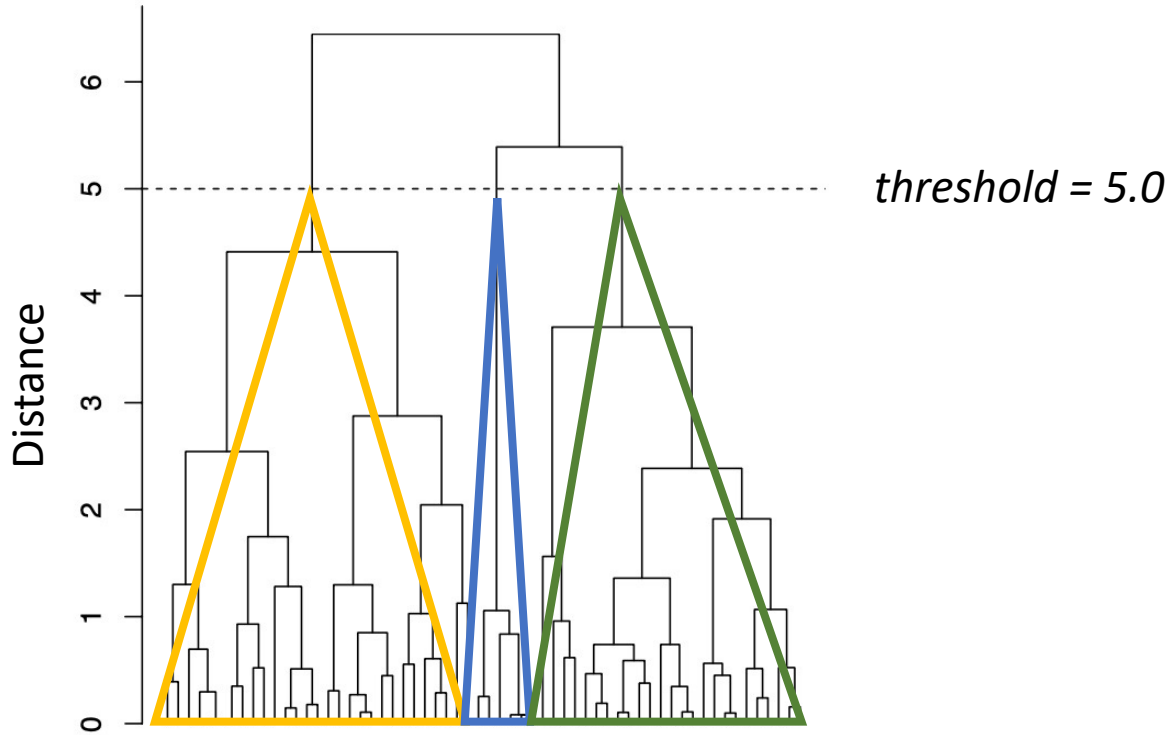
Hierarchical Clustering

- How to get the clustering assignments from the hierarchy of clusters?
- We need to set a distance threshold for cutting the tree
  - Clusters with distance smaller than the threshold will be merged.
  - Clusters with distance larger than the threshold won't be merged.

Hierarchical Clustering

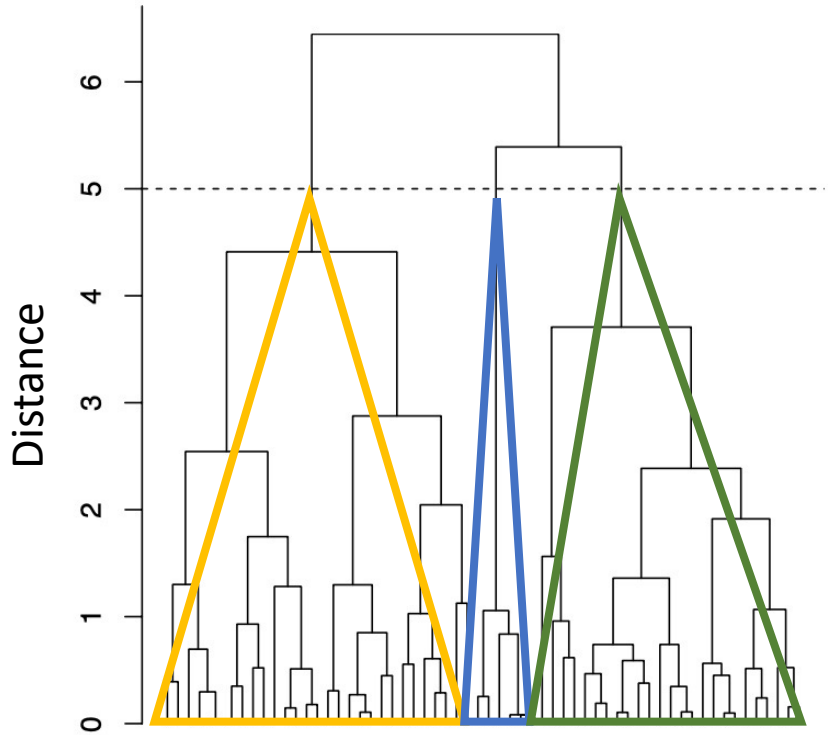
- Set a distance threshold for cutting the tree



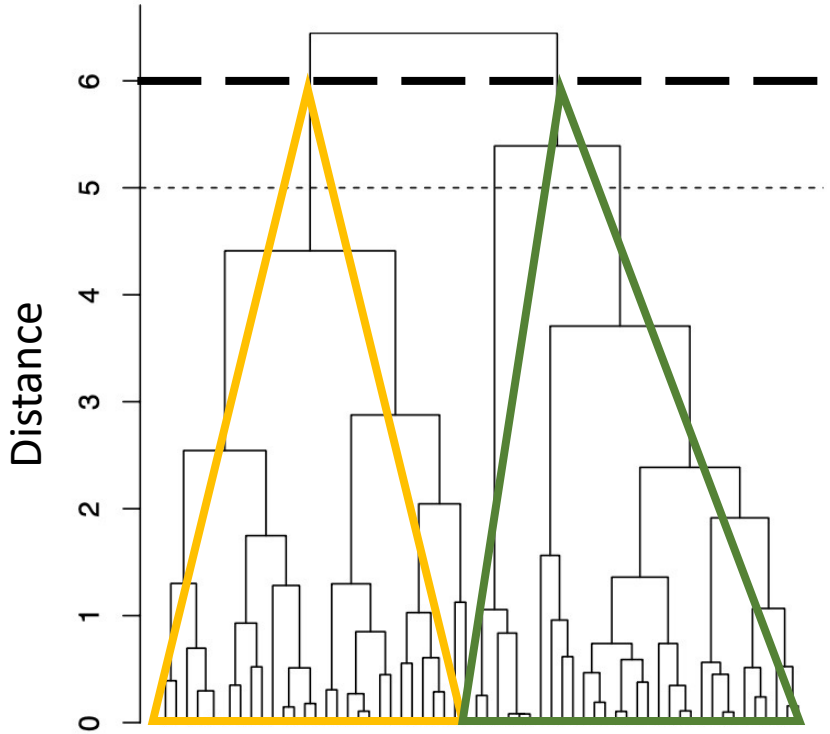


Hierarchical Clustering

- Set a distance threshold for cutting the tree



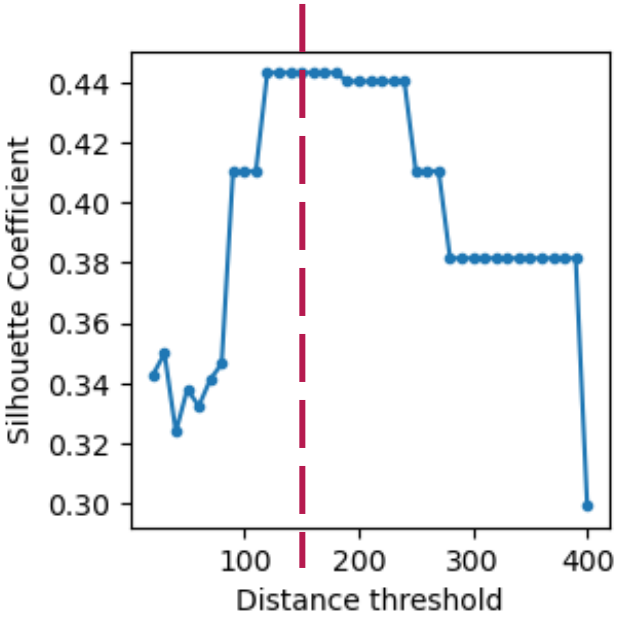
threshold = 5.0
3 clusters

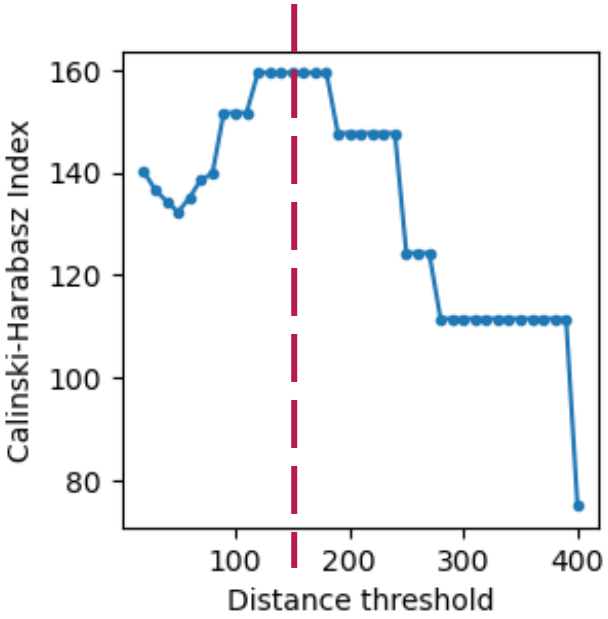


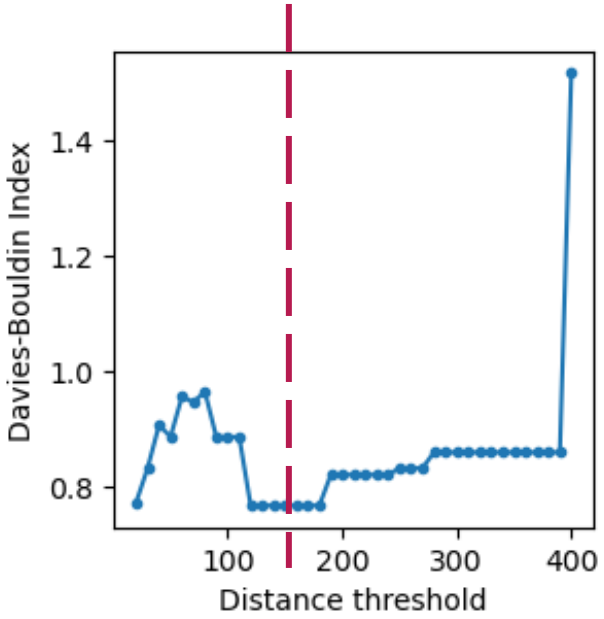
threshold = 6.0
2 clusters

Hierarchical Clustering

- How to select the distance threshold?
  - Use clustering quality evaluation metrics as reference







Hierarchical Clustering

- How to select the distance threshold?
  - Use clustering quality evaluation metrics as reference
  - Good metric values can't guarantee the meaningful results

Better clusters

.01s .03s

Silhouette Coefficient	0.33	0.48
Calinski-Harabasz Index	661.8	1326.8
Davies-Bouldin Index	1.16	0.81

Better metric values

Hierarchical Clustering

- Instead of selecting a distance threshold, we can directly specify the number of clusters we want

sklearn.cluster.AgglomerativeClustering

class sklearn.cluster.**AgglomerativeClustering**(*n_clusters=2*, ***, *affinity='deprecated'*, *metric=None*, *memory=None*, *connectivity=None*, *compute_full_tree='auto'*, *linkage='ward'*, *distance_threshold=None*, *compute_distances=False*) ¶ [source]

Parameters:

n_clusters : *int or None, default=2*

The number of clusters to find. It must be `None` if `distance_threshold` is not `None`.

Hands-on Exercise

- Exercise 08 Clustering II