Data Science and Visualization (DSV, F23)

7. Clustering (I)

Hua Lu
https://luhua.ruc.dk; luhua@ruc.dk
PLIS, IMT, RUC

Supervised vs. Unsupervised Learning

- Supervised learning generalizes from *known examples* to automate decision-making processes.
  - Classification: Predict a discrete value from a *pre-defined* set of class labels
  - Regression: Predict a continuous value from a continuous range
- Unsupervised learning does *not* need any known examples. It works on input data directly.
  - Clustering
  - Association rules
  - Dimensionality reduction

Agenda

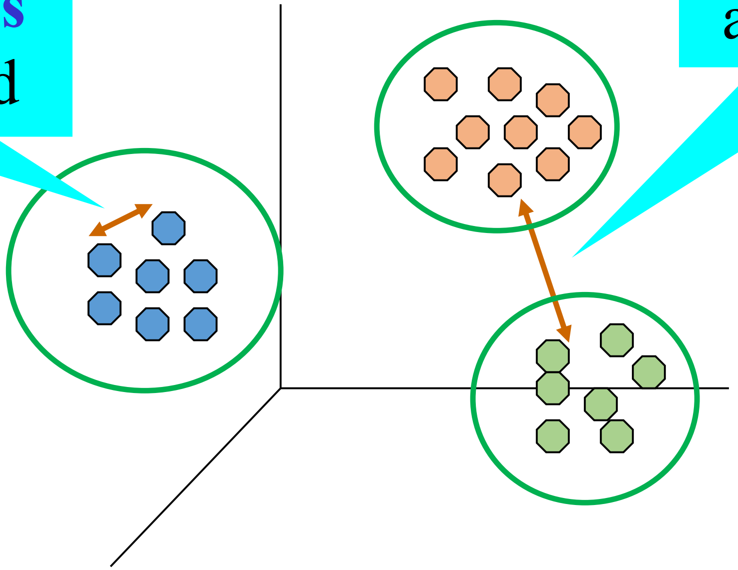
- Clustering in general
- k-Means
- Hierarchical clustering

What is Clustering?

- Grouping of objects, *s.t.* the objects in a group (*cluster*) are similar (or related) to each other and different from (or unrelated to) objects in other groups

Intra-cluster **dissimilarities** are minimized

Inter-cluster **dissimilarities** are maximized



Dissimilarity is often measure as **distance**.

A clustering is a set of clusters. Sometimes we use clustering (analysis) to refer to the process of deciding all clusters.

A More Formal Definition of Clustering

- **Input**: A collection *C* of data objects
- **Output**: A set of *disjoint* clusters whose union is *C*.
  - Objects in the same clusters are *similar* to each other.
  - Objects in one cluster are *dissimilar* to those in other clusters.
- **Process**: Finding similarities between data objects according to the characteristics in the data, and grouping similar data objects into clusters.

- Typical use of clustering
  - As a **stand-alone tool** to get insight into data distribution
  - As a **preprocessing step** for other algorithms
- **Unsupervised learning**: clusters are *not* pre-defined
  - Classification is *supervised learning*: we have training data with known class labels

Classification vs. Clustering

Classification

- Predefined classes
  - Number of classes
  - Meaning of classes
- Training
  - Supervised learning
- Work for any number of objects
  - Given an object, a classifier (trained model) assigns it to a class

Clustering

- No prior knowledge about
  - Number of clusters *
  - Meaning of clusters
- No training
  - Unsupervised learning
- There must be a sufficient number of objects
  - Meaningless to conduct clustering analysis on one or few objects

Basic Steps of Clustering

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

- What attributes should we consider?

- How to measure similarity?

- How close two points should be to get into the same cluster?

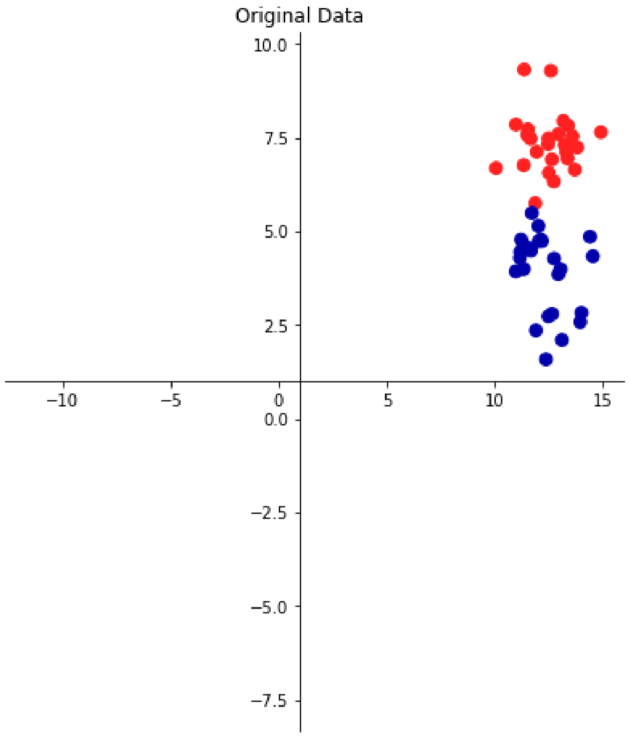
Domain expertise may be needed.

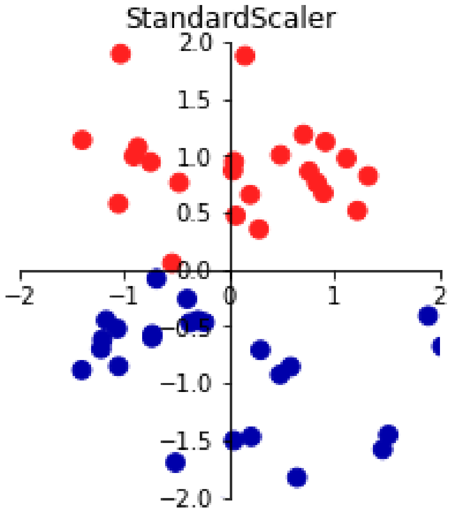
Similarity and Distance

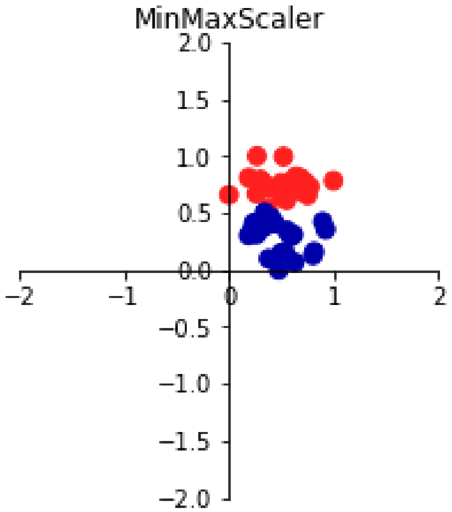
age	income
64	87083.24
33	76807.82
24	12043.60
33	61972.00
78	60120.32
62	40058.42

- If we calculate distance directly on this dataset, the distance will very likely be dominated by the income values.
  - Dimensions age and income are not measured in the same scale.
- Data (re)scaling is needed before reasonable distances can be calculated on the two dimensions.
  - This is part of preprocessing of the data before distance based ML algorithms, e.g., kNN for classification and those for clustering

Preprocessing and Scaling







Standard Scaling (aka standardization or Z-score normalization)

- Afterwards, for each feature has mean=0 and variance=1

Min-Max Scaling (aka Normalization)

- Shifts the data, *s.t.* each feature falls in [0..1]

Typical Clustering Algorithms

- Partitioning approach (centroid-based)
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical method: **K-means**
- Hierarchical approach (connectivity-based)
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical method: **Bottom-up** or **top-down**
- Density-based approach
  - Based on connectivity and density functions
  - Typical method: **DBSCAN** (next week)

Agenda

- Clustering in general
- k-Means
- Hierarchical clustering

The K-Means Clustering Method

- Given K, the K-means algorithm works in four steps

Initialization

1. Partition all objects *randomly* into K nonempty subsets

Iterations

2. Compute *seed points* as the centroids of the clusters of the current partitioning
    - The centroid is the center, i.e., mean, of all data objects in a cluster

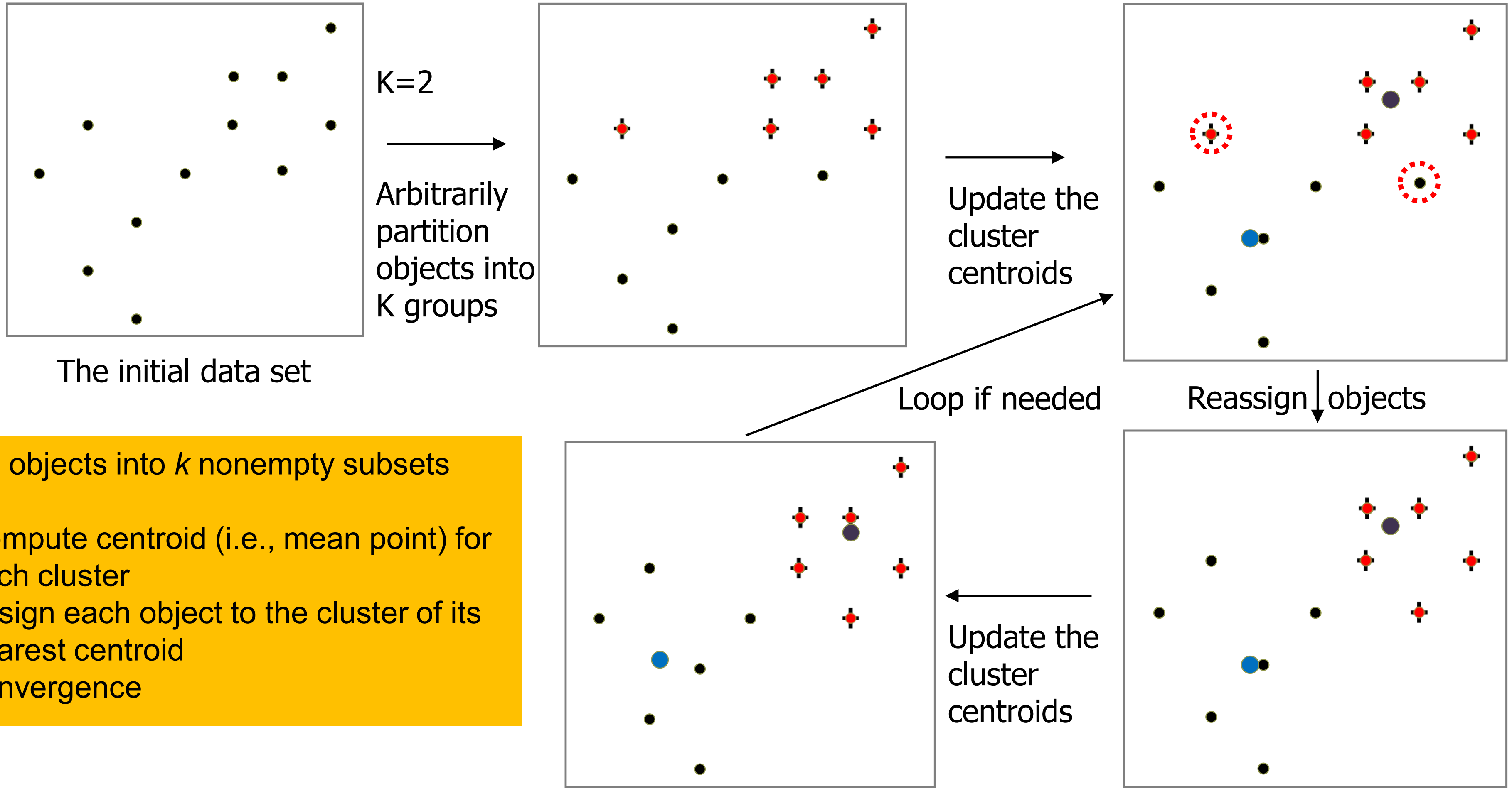
$$centroid = C_m = \frac{\sum_{i=1}^{N}(t_{mi})}{N}$$

3. Assign each object to the cluster with the *nearest* seed point

Convergence

4. Go back to Step 2, repeat and stop when the assignment does not change or the change is sufficiently small

An Example of K-Means Clustering



Partition objects into *k* nonempty subsets

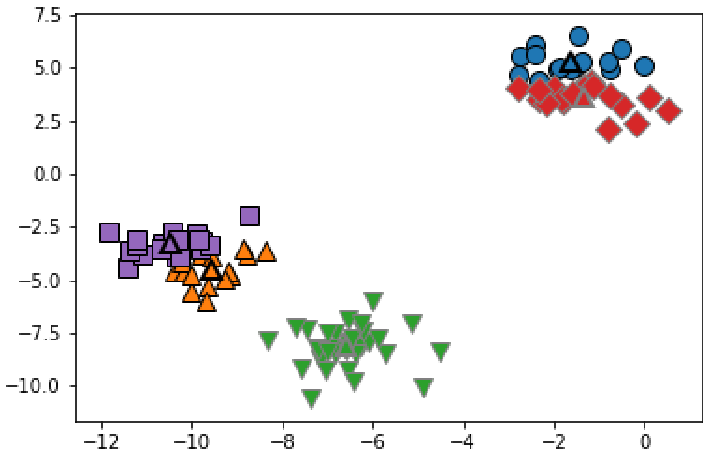
Repeat

- Compute centroid (i.e., mean point) for each cluster
- Assign each object to the cluster of its nearest centroid

Until convergence

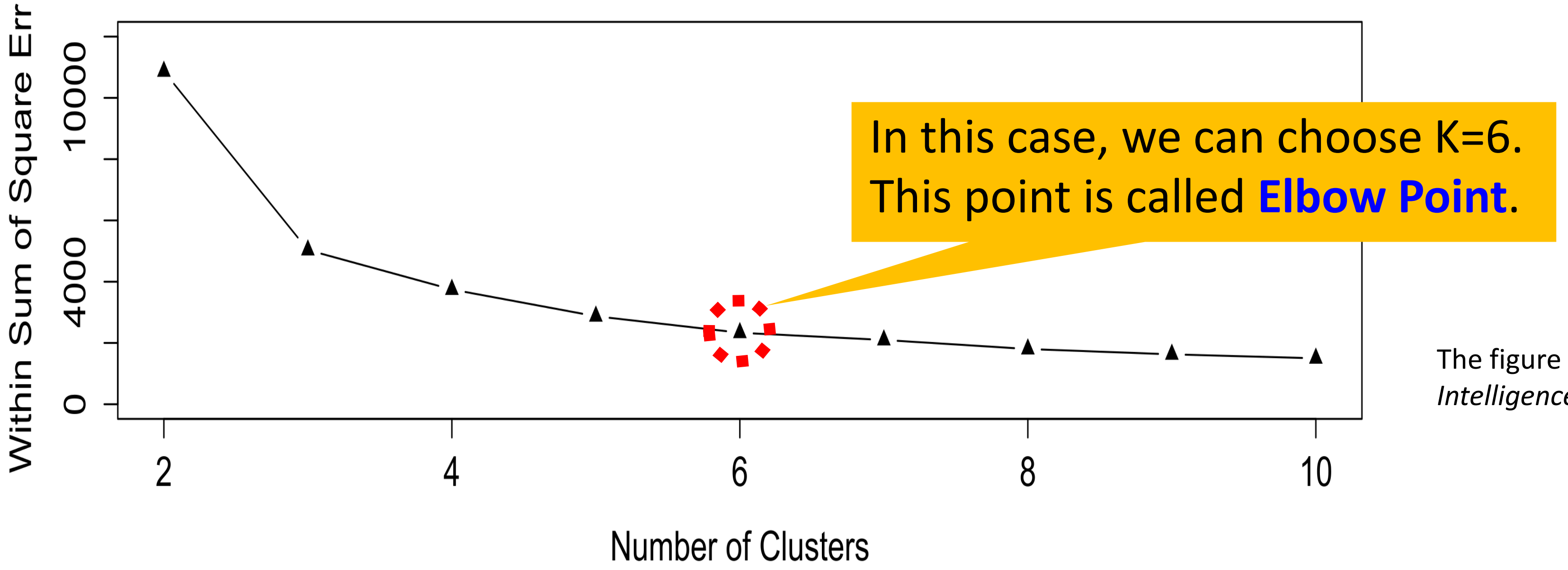
Note on K

- The time complexity of K-means depends on K
- A larger K:
  - More clusters to maintain, more mean points to calculate, and more distance calculations and comparisons in the reassignment step.
- A smaller K:
  - Less clusters to maintain, less mean points to calculate, and less distance calculations and comparisons in the reassignment step.
- K may also affect the clustering quality
- We may use EDA and visualization to decide K.



Elbow Method: To decide the best K

- Let c_i be the *centroid/mean* of cluster C_i in a given clustering result.
- We check the Sum of Squared Distance (aka sum of squared error SSE) for all points *ps* in all clusters: $E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$
- Vary K from 1 to a max (e.g., 10), plot a graph for (K, SSE), and find the K value *after which* the performance gain is *insignificant*.



The figure is from *Introduction to R for Business Intelligence* by Jay Gendron

Example in Jupyter Notebook

- Age-Income dataset
  - 8105 data objects, 3 columns
  - Available in Moodle
  - From Jay Gendron's *Introduction to R for Business Intelligence*, Packt Publishing Ltd., 2016
- Question:
  - How are people segmented in terms of their age *and* income?

- Lecture7_KMeans_age-income.ipynb

	bin	age	income
0	60-69	64	87083.236510
1	30-39	33	76807.824635
2	20-29	24	12043.598766
3	30-39	33	61972.002432
4	70-79	78	60120.315192

Another K-Means Example

- Given: {2, 4, 10, 12, 3, 20, 30, 11, 25}, K=2
- Randomly assign means: m_1=3, m_2=4
- C_1={2, 3}, C_2={4, 10, 12, 20, 30, 11, 25}
  - Update means: m_1=2.5, m_2=16
  - Need to move 4 as 4 is closer to 2.5 than to 16
- C_1={2, 3, 4}, C_2={10, 12, 20, 30, 11, 25}
  - Update means: m_1=3, m_2=18
  - Need to move 10 as 10 is closer to 3 than to 18
- C_1={2, 3, 4, 10}, C_2={12, 20, 30, 11, 25}
  - Update means: m_1=4.75, m_2=19.6
  - Need to move 11 and 12 as they are closer to 4.75
- C_1={2, 3, 4, 10, 11, 12}, C_2={20, 30, 25}
  - Update means: m_1=7, m_2=25
  - Nothing to move, and the algorithm stops

Note

- Here we start with two randomly decided means, not K (=2) subsets.
- The overall effect is the same

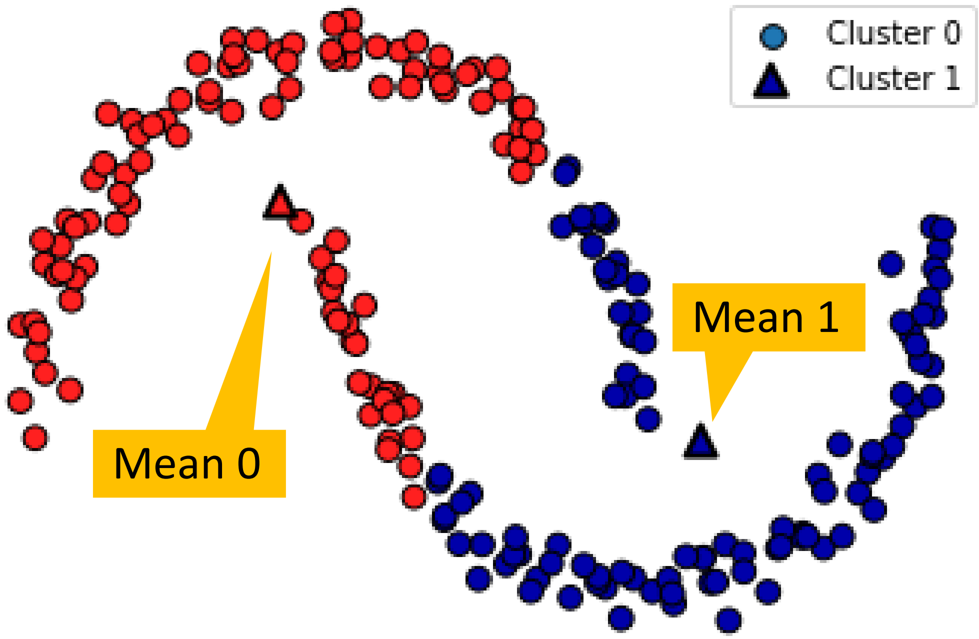
Exercises --- using the same set of numbers:

- Work out the clustering result using 2-means but starting with m_1=10, m_2=20
- Work out the clustering result using 3-means.
  - Start with 3 initial random means
  - Or with 3 initial random clusters

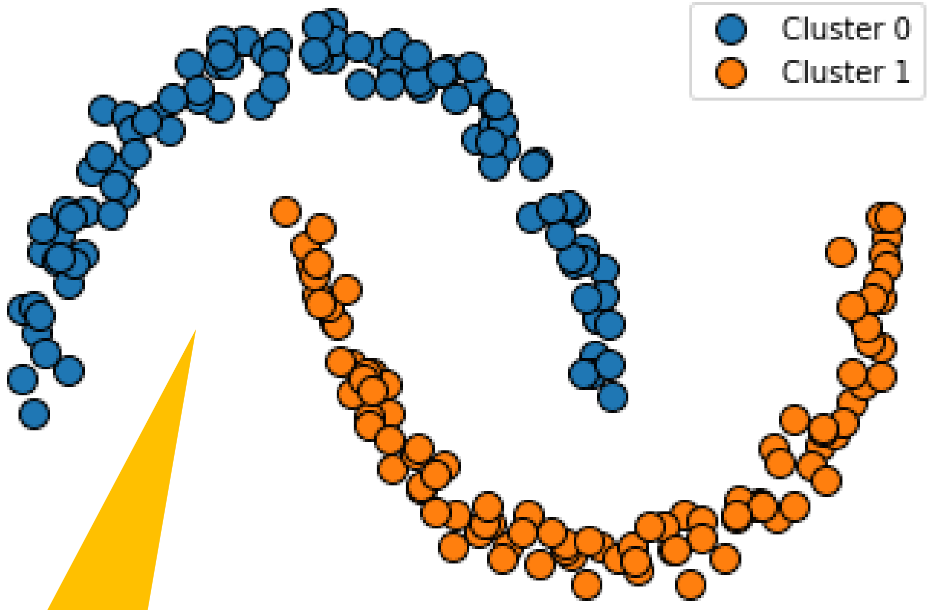
Weaknesses of K-Means

- Applicable only to objects in a *continuous* n-dimensional space
  - We cannot calculate means on categorical values, e.g., {CPH, RO, AAL}
- Initialization matters. Need to specify K, the number of clusters, in advance
  - In literature, there are ways to automatically determine the best k
- Convergence
  - Stop condition can be ‘Relatively few points change clusters’.
  - Often terminates at a *local* optimal.
- Sensitive to noisy data and outliers
- Not suitable to discover clusters with non-convex shapes

K-means on Non-convex Shapes



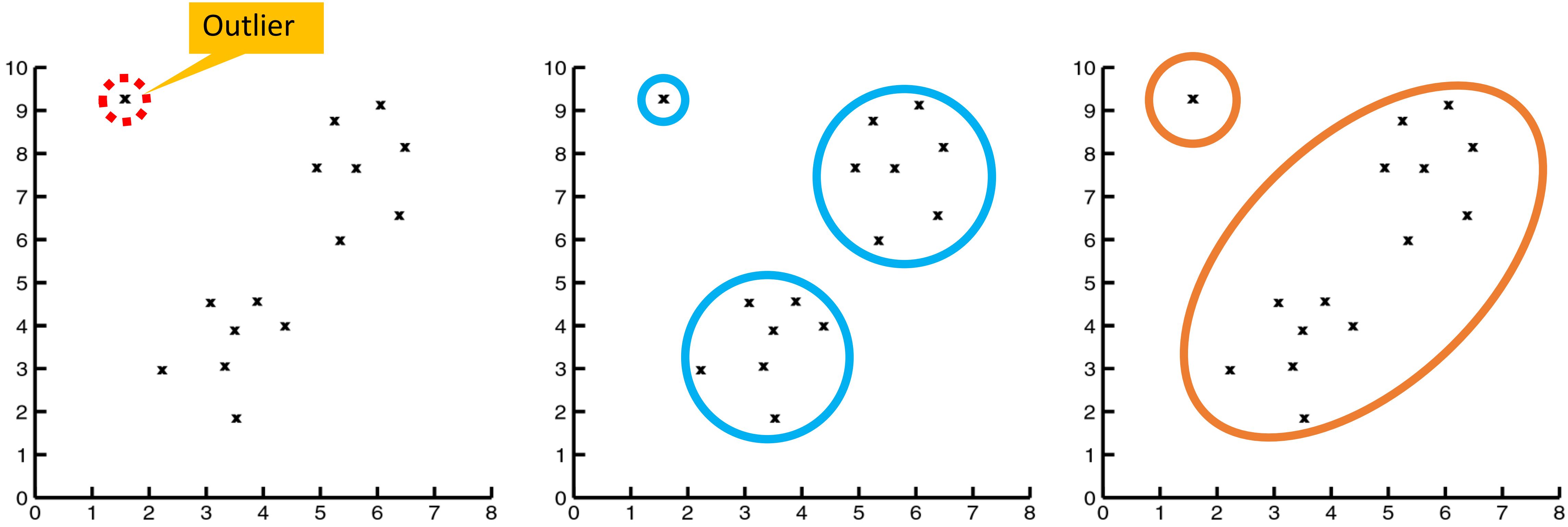
K-means clustering result (K=2)



Density Based Spatial Clustering of Applications with Noise (**DBSCAN**)

Desired clustering result

Impact of Outliers on k-Means



Dataset with outlier

K=3

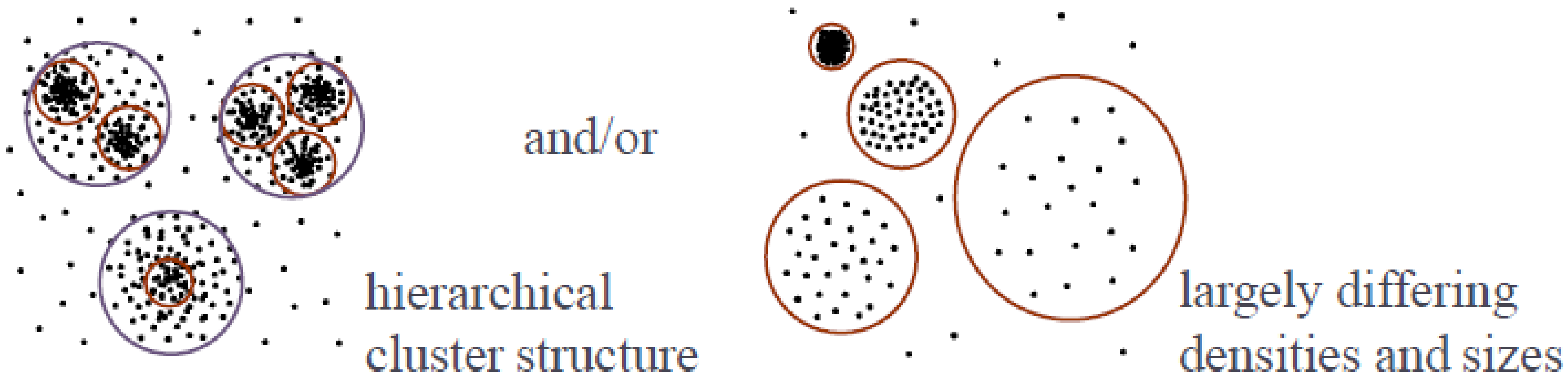
K=2

Agenda

- Clustering problem
- k-Means
- Hierarchical clustering

Why Hierarchical Clustering?

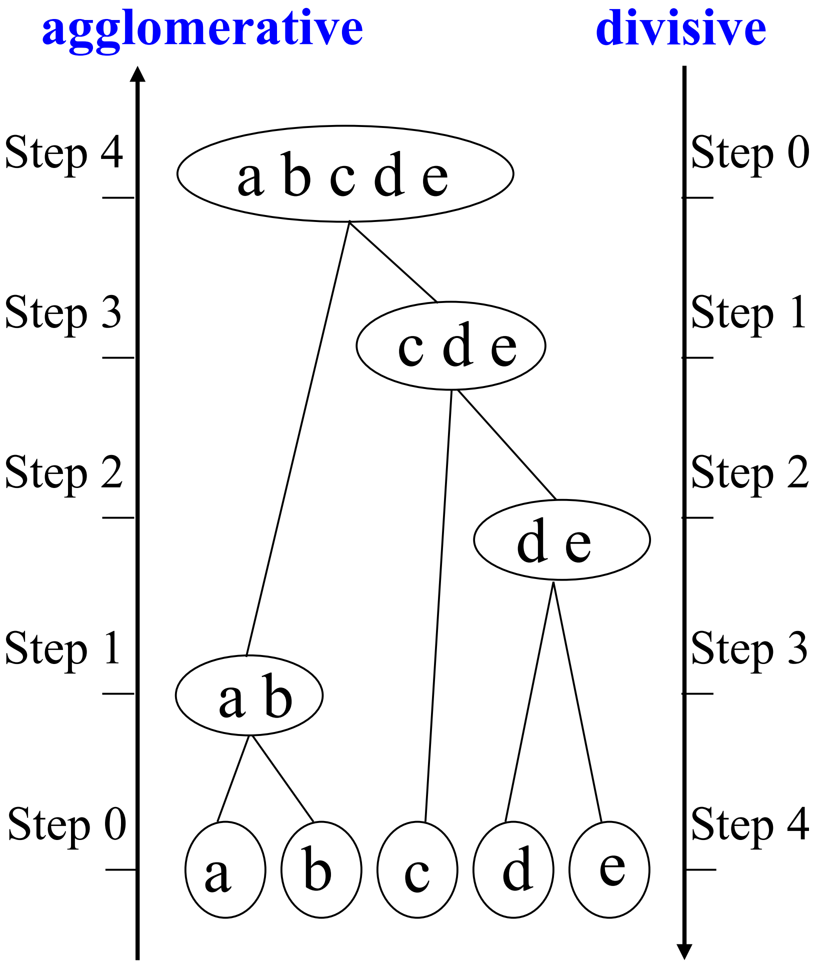
- Sometimes, global parameters to separate all clusters with a partitioning clustering method may *not* exist.



- Hierarchical clustering can handle such situations.
    - Clusters are created in *levels*, actually creating sets of clusters at each level.

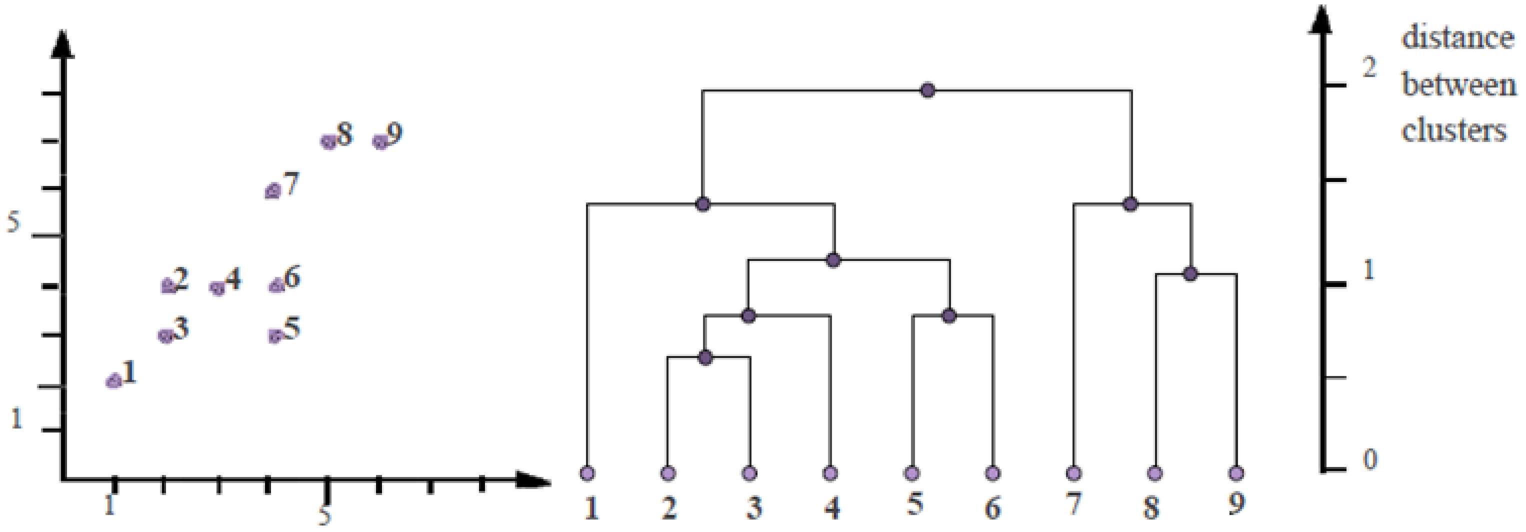
Hierarchical Clustering Approaches

- HC uses distance matrix as clustering criteria. It does not require the number of clusters as an input, but needs a termination condition.
- Agglomerative clustering algorithms
  - Initially each item in its own cluster
  - Iteratively clusters are merged together
  - Bottom Up
- Divisive clustering algorithms
  - Initially all items in one cluster
  - Large clusters are successively divided
  - Top Down

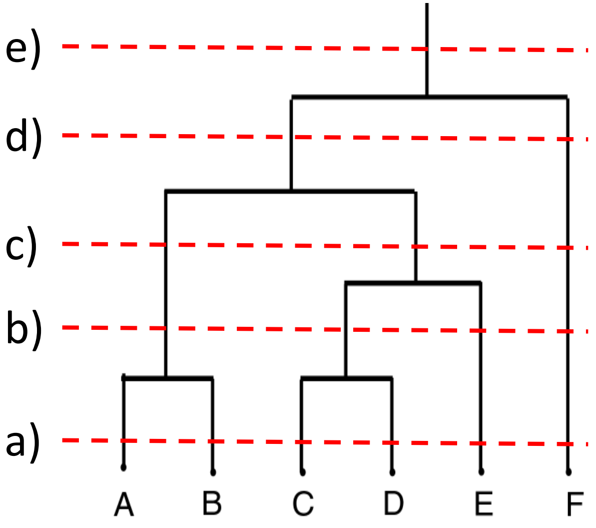


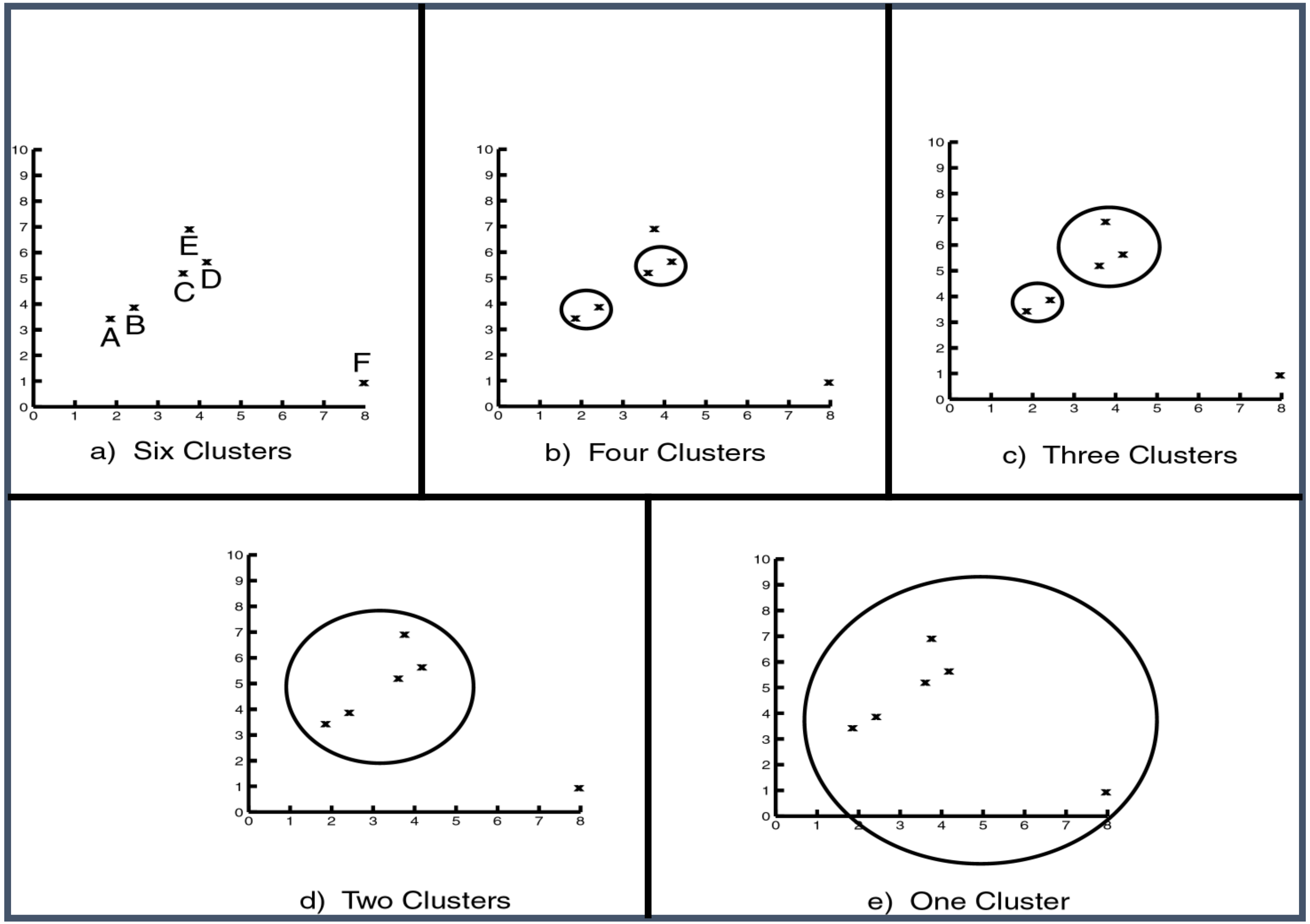
Dendrogram

- **Dendrogram**: a tree data structure that illustrates hierarchical clustering techniques.
- Each level shows clusters for that level.
  - Leaf: individual data points
  - Root: one cluster
  - A cluster at level i is the union of its child clusters at level i+1.
- The height of an internal node represents the distance between its two child nodes.



Levels of Clustering (Agglomerative)



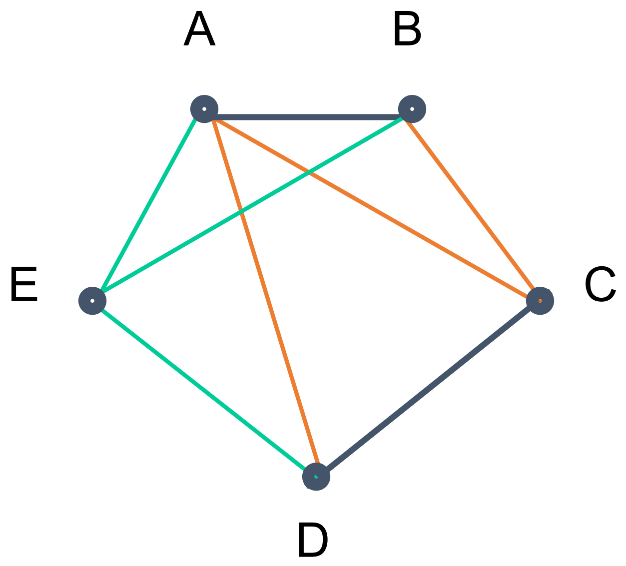


Agglomerative Clustering Algorithm

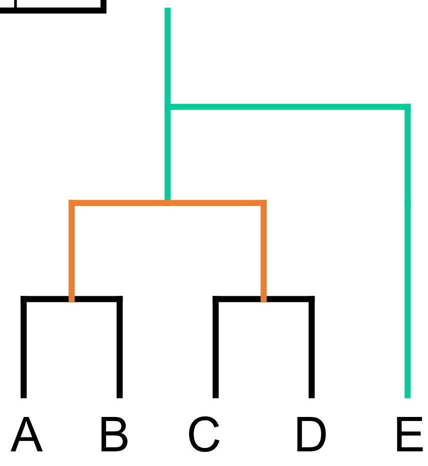
- Most popular hierarchical clustering technique
- Basic algorithm:
  1. Compute an adjacency matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge two clusters if the distance is small enough
  5. Update the adjacency matrix and distance threshold
  6. **Until** only a single cluster remains
- Key operation: computing similarity of two clusters
  - Different ways to define distance between clusters.
  - They produce different clustering results.

An Agglomerative Example

	A	B	C	D	E
A	0	1	2	2	3
B	1	0	2	4	3
C	2	2	0	1	5
D	2	4	1	0	3
E	3	3	5	3	0



For simplicity, we work on the original adjacency matrix and use **MIN** in this example.

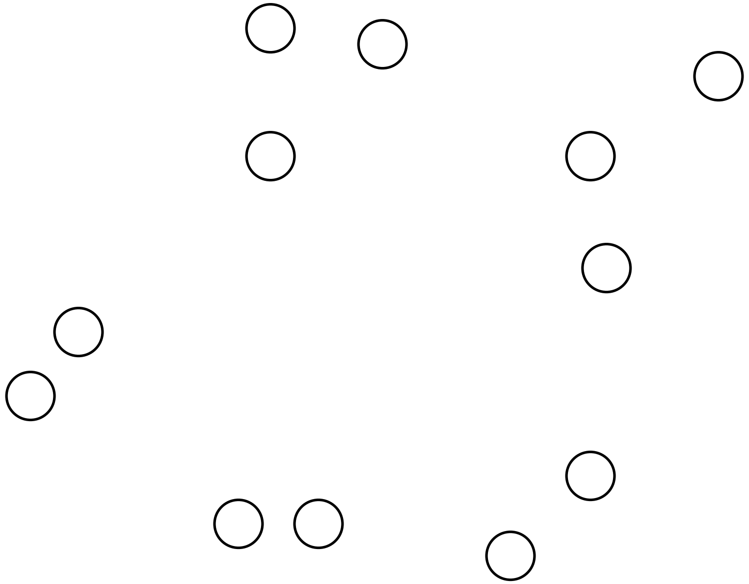


Distance threshold (for *similarity*)

1 2 3

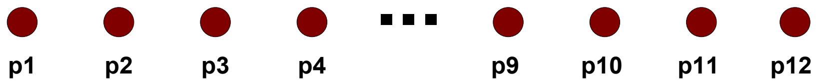
Starting Situation

- Start with clusters of individual points and an adjacency matrix



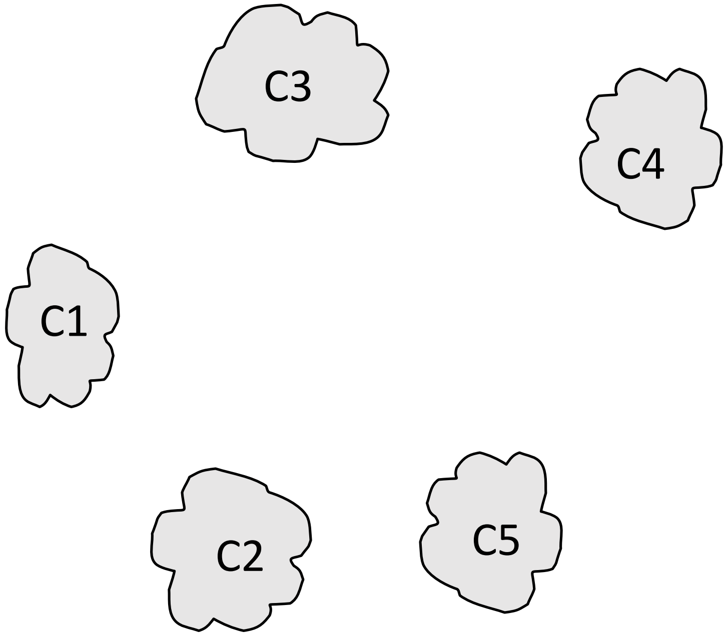
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Adjacency Matrix



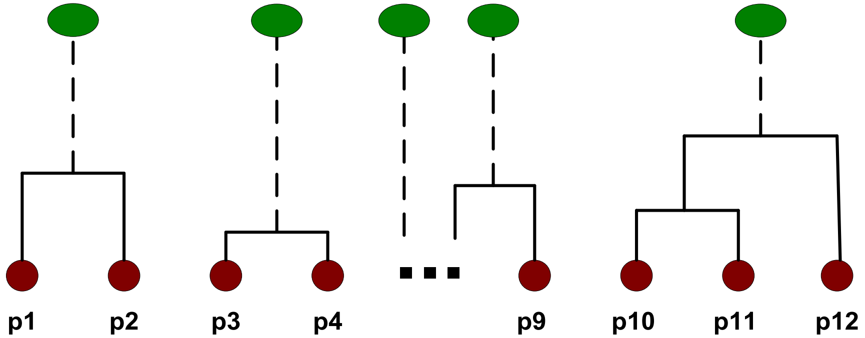
Intermediate Situation

- After some merging steps, we have some clusters



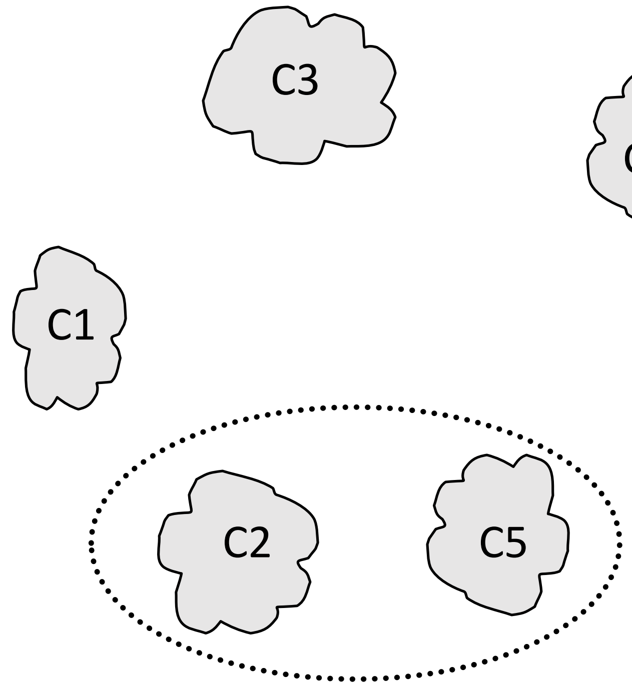
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

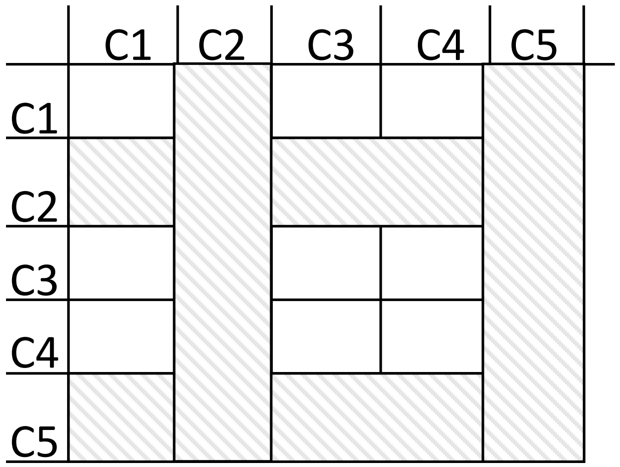
Adjacency Matrix



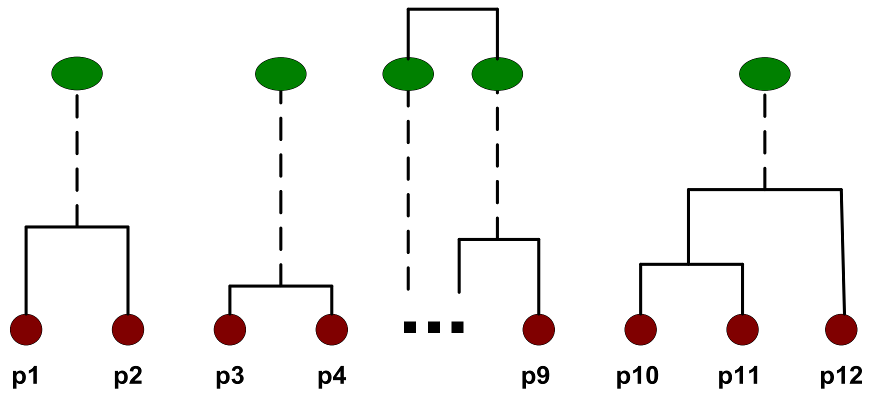
Intermediate Situation (cont.)

- We want to merge two *closest* clusters (e.g., C2 and C5) and update the adjacency matrix.





Adjacency Matrix

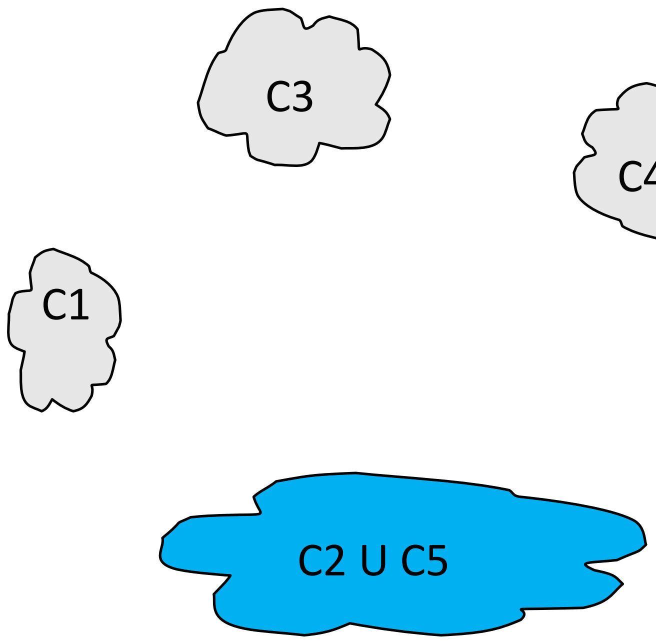


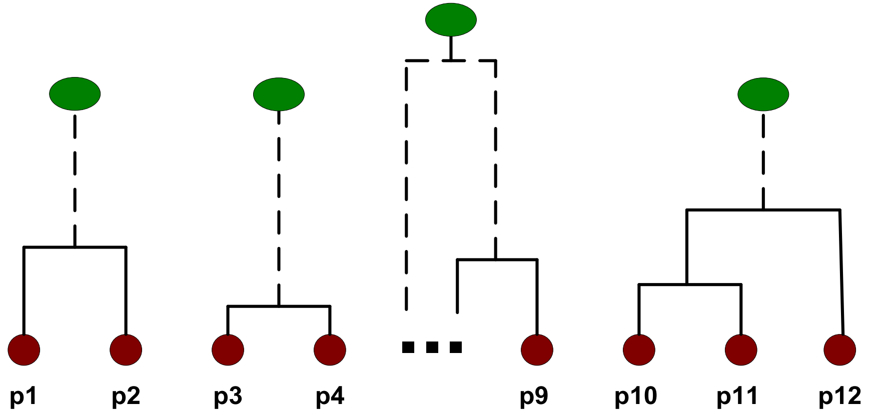
After Merging

- How to update the adjacency matrix?

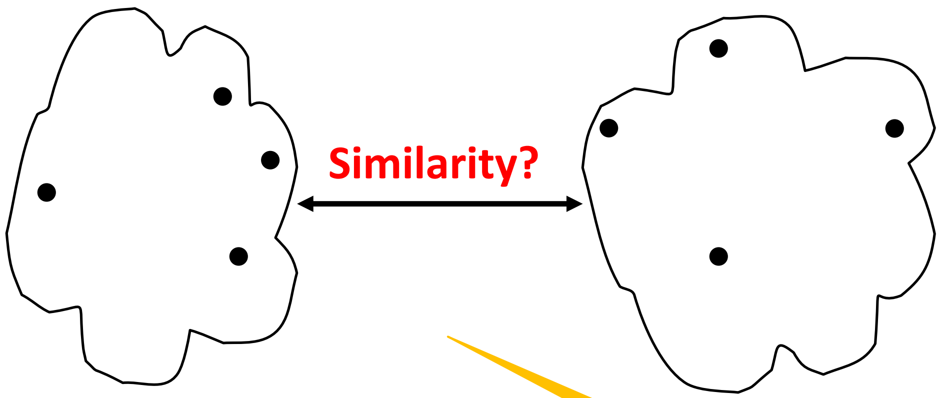
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix

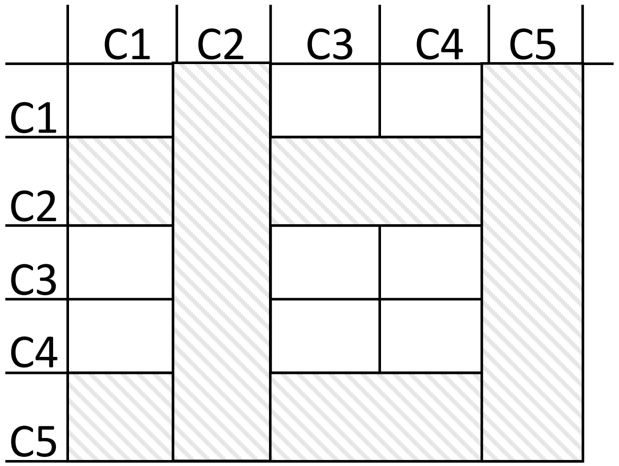




How to Define Inter-Cluster Similarity?



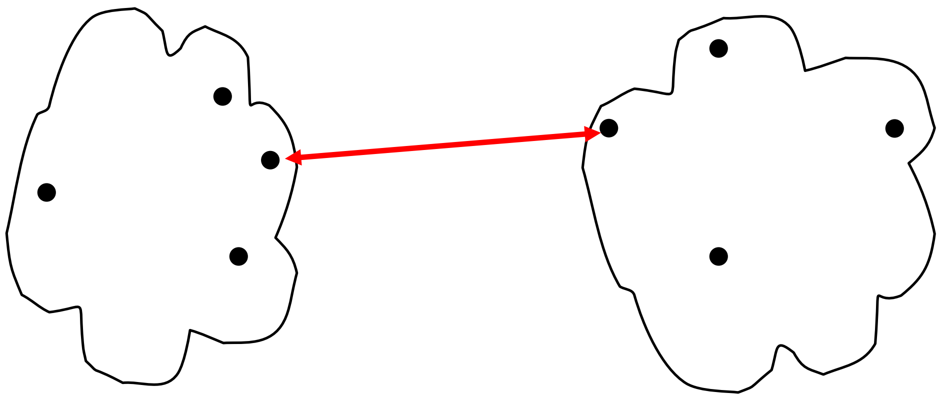
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Objective function

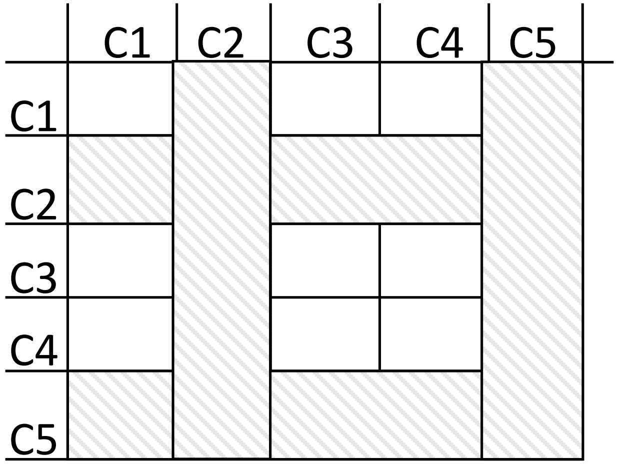


Adjacency Matrix

How do we find/decide the *closest* pair of clusters?

How to Define Inter-Cluster Similarity?

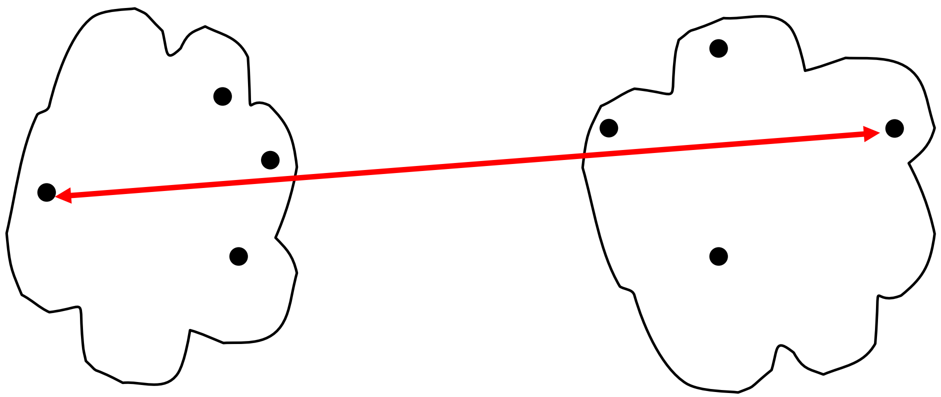


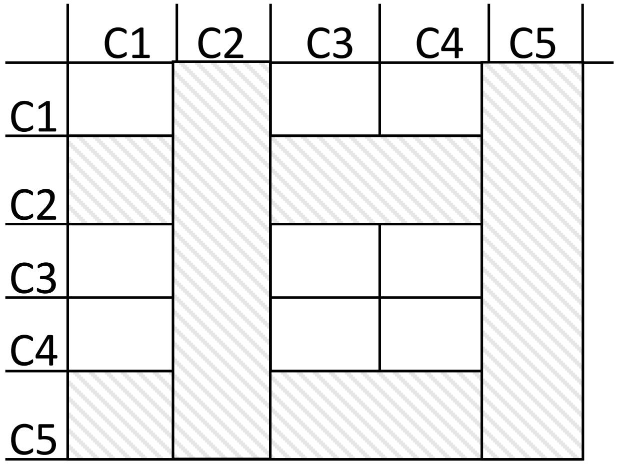


Adjacency Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Objective function

How to Define Inter-Cluster Similarity?

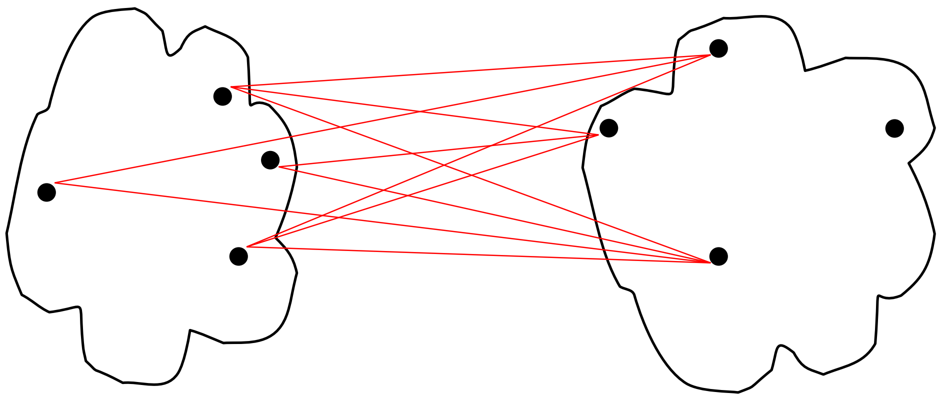


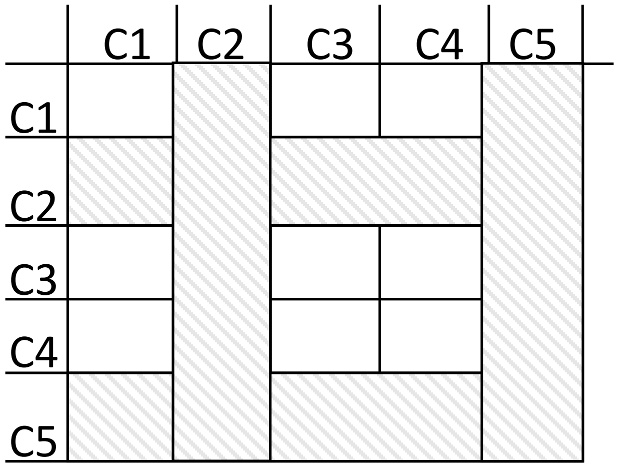


Adjacency Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Objective function

How to Define Inter-Cluster Similarity?

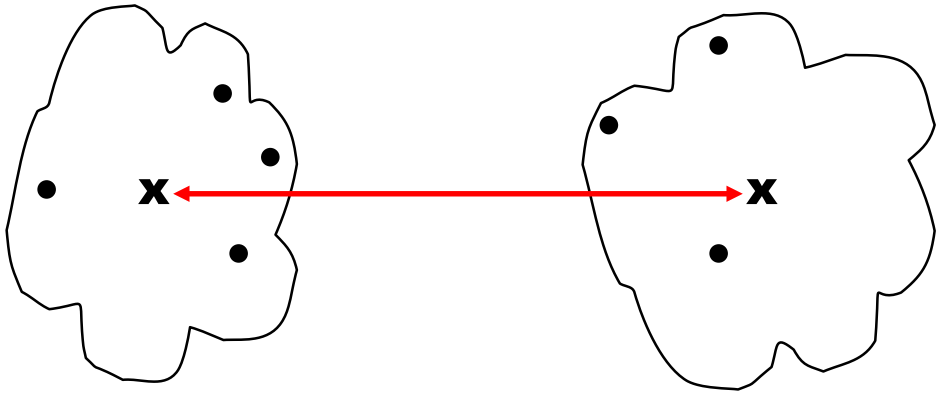


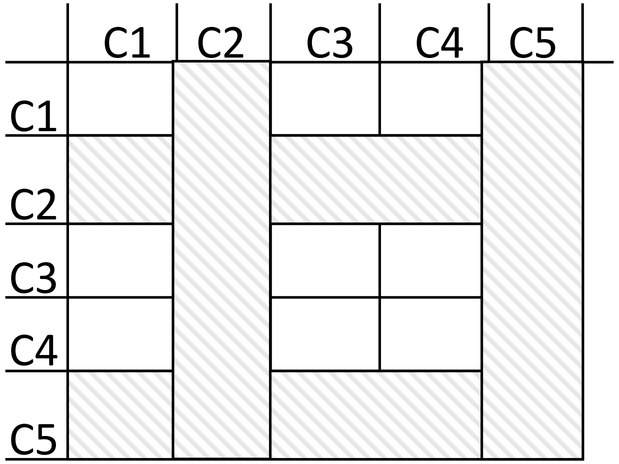


Adjacency Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Objective function

How to Define Inter-Cluster Similarity?





Adjacency Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Objective function

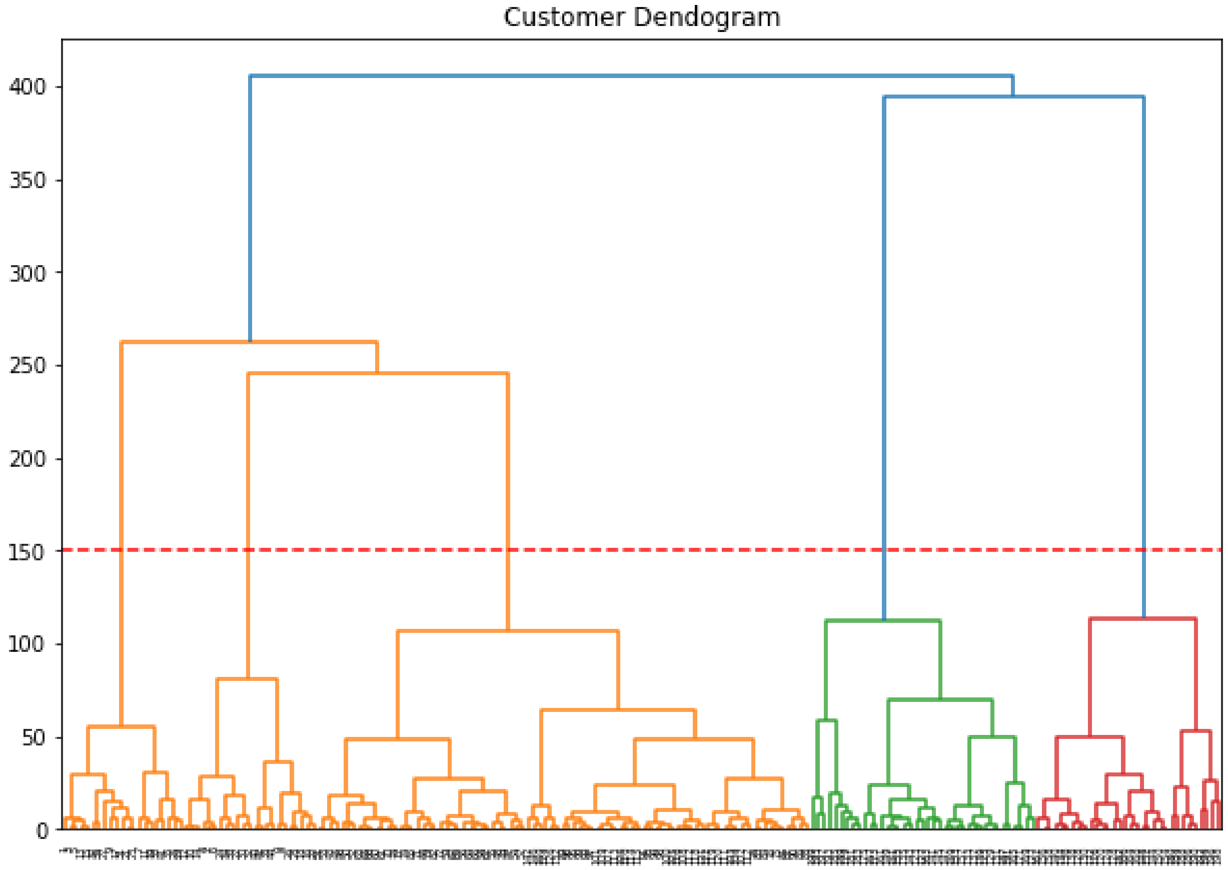
Inter-Cluster Similarity: Ward's Method

- Similarity of two clusters measured as **increase** in sum of squared error (SSE) when they are merged
  - Say we may merge clusters C1 and C2 into Cm
  - Increase = SSE(Cm) – SSE(C1) – SSE(C2)
  - Refer to Slide 15 for SSE

- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical "analogue" of K-means
  - Can be used to initialize K-means

‘Best’ Number of Clusters from Dendrogram

- Locate the largest vertical difference between nodes
  - Avoid to merge very distant or dissimilar clusters
- Draw a horizontal line through it.
  - If more options, choose the largest vertical difference again
- Count the vertical lines it intersects
  - The *optimal* number of clusters.



Single, Complete and Average Link

- Another way to view hierarchical algorithm is as a process that *creates links* between elements in order of *increasing* distance
  - MIN – Single Link: merges two clusters X and Y when a *single pair* of elements is linked

    $$dist\_sl(X,Y) = \min_{x \in X, y \in Y} dist(x,y)$$

  - MAX – Complete Link: merges two clusters when *all pairs* of elements have been linked

    $$dist\_cl(X,Y) = \max_{x \in X, y \in Y} dist(x,y)$$

  - AVG – Average Link: merges two clusters when average pair of elements have been linked

    $$dist\_al(X,Y) = \frac{1}{|X| \cdot |Y|} \cdot \sum_{x \in X, y \in Y} dist(x,y)$$

Example in Jupyter Notebook

- Agglomerative clustering
  - Age-Income dataset
  - How smaller clusters are merged into larger clusters.
  - Lecture7_AggClustering_age-income.ipynb

- Dendrogram
  - Customer shopping data
    - Annual Income and Spending Score
  - Deciding number of clusters
  - Effect of different inter-cluster similarity measures
  - Lecture7_Dendrogram_shopping.ipynb

Summary

- Clustering Problem
  - Comparison with classification
- Clustering techniques
  - K-Means clustering
  - Agglomerative clustering
  - Dendrogram
- Elbow method

References

- Mandatory reading
  - Muller and Guido: Introduction to Machine Learning with Python, O'Reilly, 2016
    - Chapter 3: Clustering: k-Means Clustering, Agglomerative Clustering
- Further readings
  - https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/
  - Documentation
    - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
    - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

Exercises

1. K-means example on page 17
2. Work with the bikes dataset (in Moodle) in Jupyter Notebook
    1. Apply K-means clustering
        - Vary k, e.g., 2, 3, 4, 5 …
        - Use the Elbow method to find the best k
        - Visualize the K-means clustering result of the best k
    2. Apply agglomerative clustering
        - Show the procedure of how 10 clusters are merged until a single cluster is obtained
        - Draw the dendrogram with linkage=ward
        - Figure out the best number of clusters
        - Generate the corresponding clustering result, and visualize it