Data Intensive Systems (DIS)
KBH-SW7 E25

8. Clustering

AALBORG
UNIVERSITY

Supervised vs. Unsupervised Learning

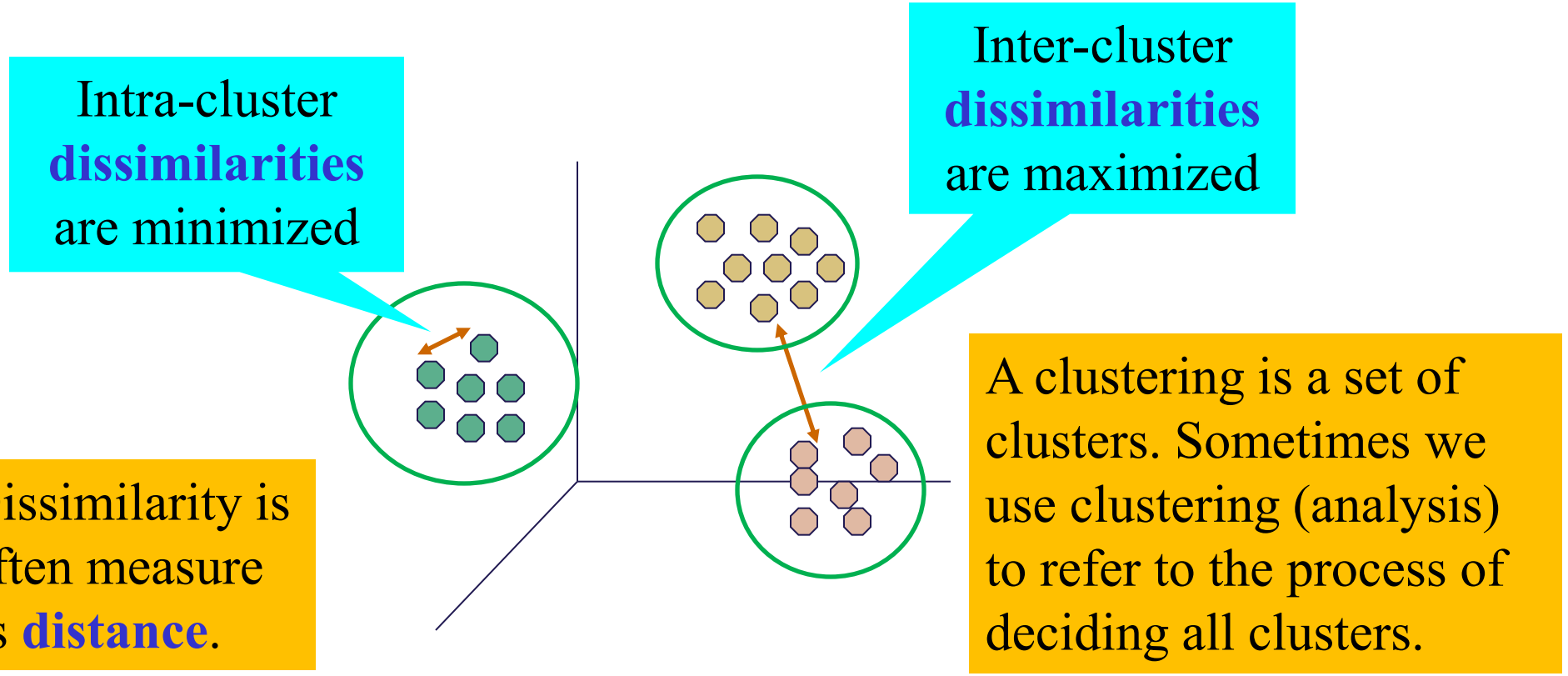
- Supervised learning generalizes from *known examples* to automate decision-making processes.
  - Classification: Predict a discrete value from a *pre-defined* set of class labels
  - Regression: Predict a continuous value from a continuous range
- Unsupervised learning does *not* need any known examples.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

Agenda

- Clustering in general
- k-Means
- Hierarchical clustering
- DBSCAN
- Evaluation of 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



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?

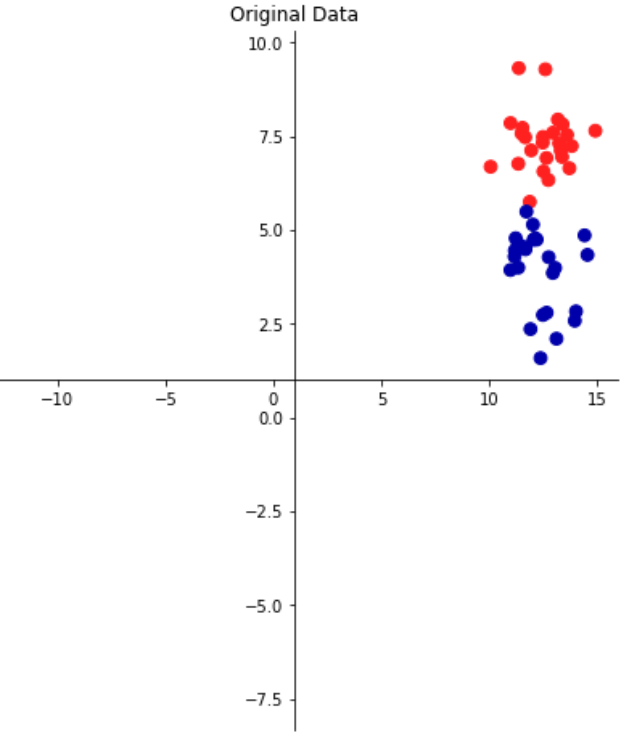
Domain expertise may be needed.

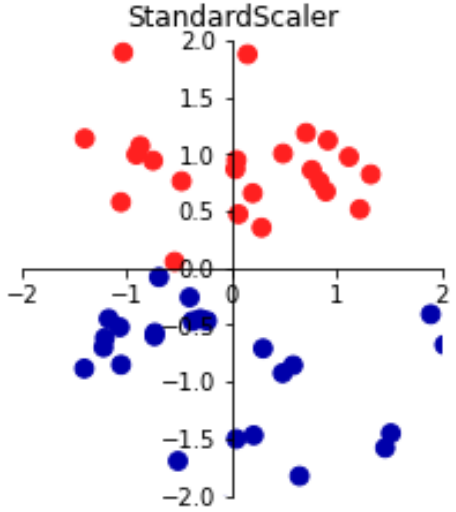
Similarity and Distance

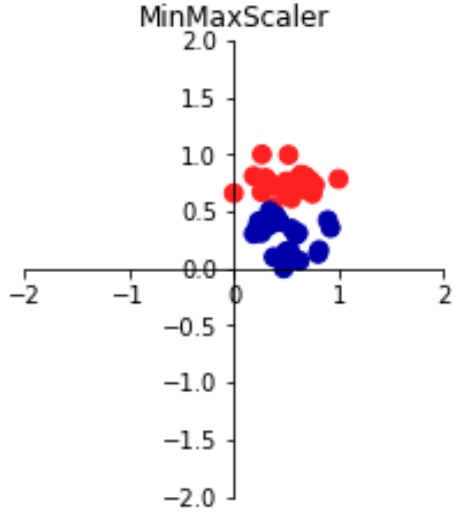
age	income
64	87083.24
33	76807.82
24	12043.60
33	61972.00
78	60120.32
62	40058.42

- 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**

Agenda

- Clustering in general
- k-Means
- Hierarchical clustering
- DBSCAN
- Evaluation of 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

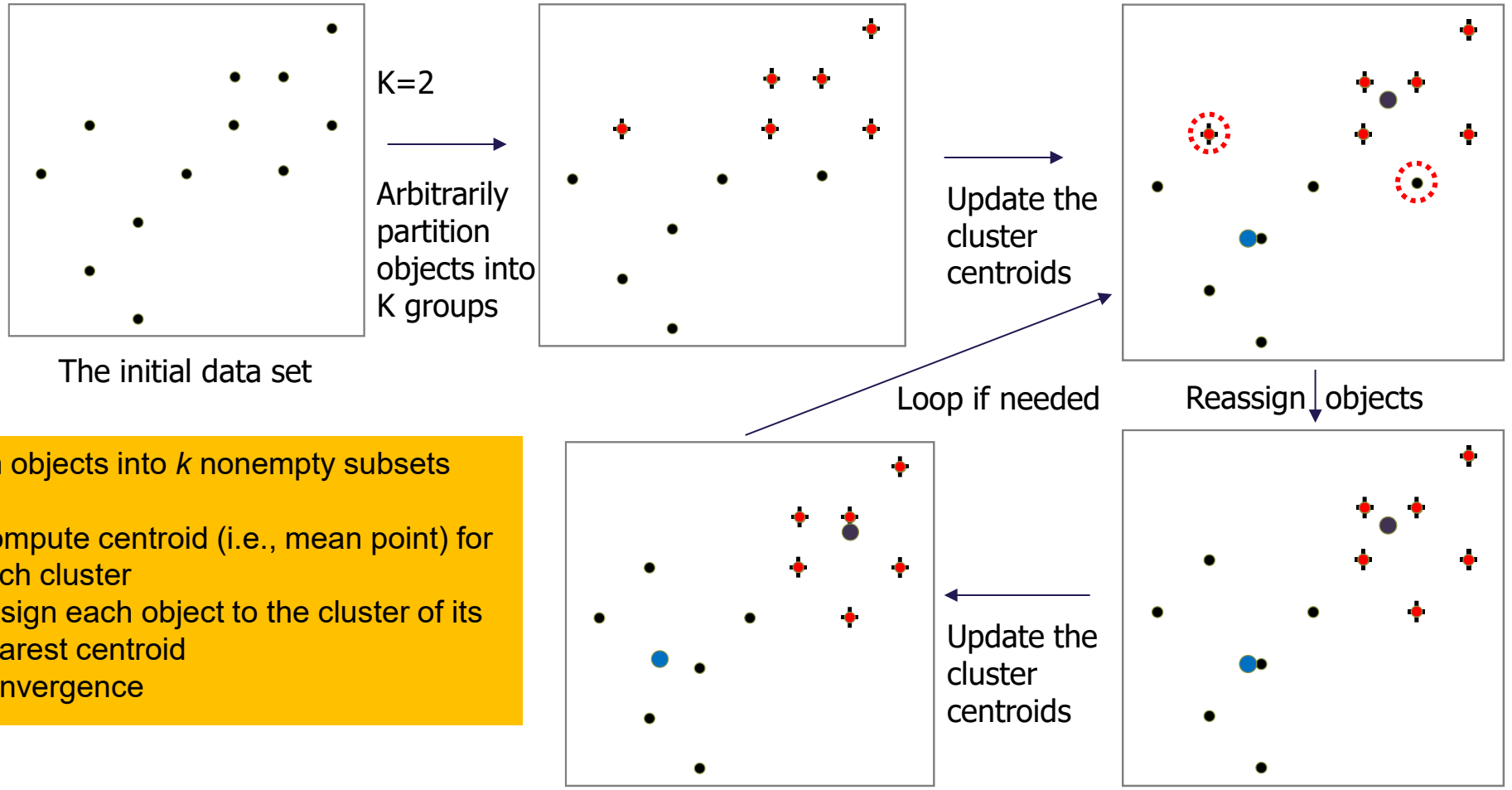
$$centroid = C_m = \frac{\sum_{i=1}^{N}(t_{mi})}{N}$$

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

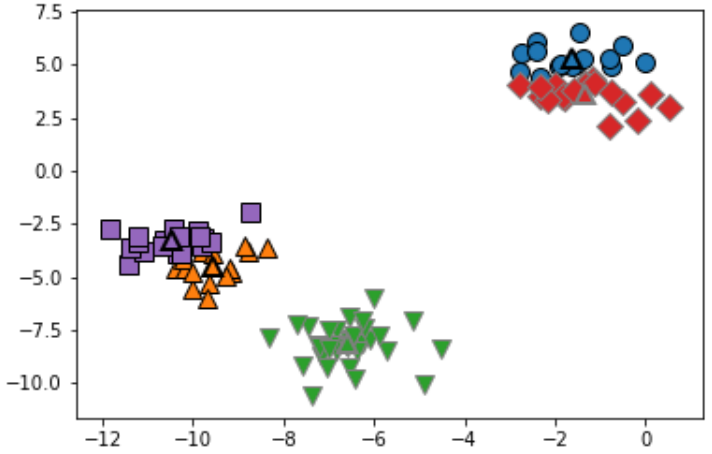


Partition objects into *k* nonempty subsets
Repeat
- Compute centroid (i.e., mean point) for each cluster
- Assign each object to the cluster of its nearest centroid

Until convergence

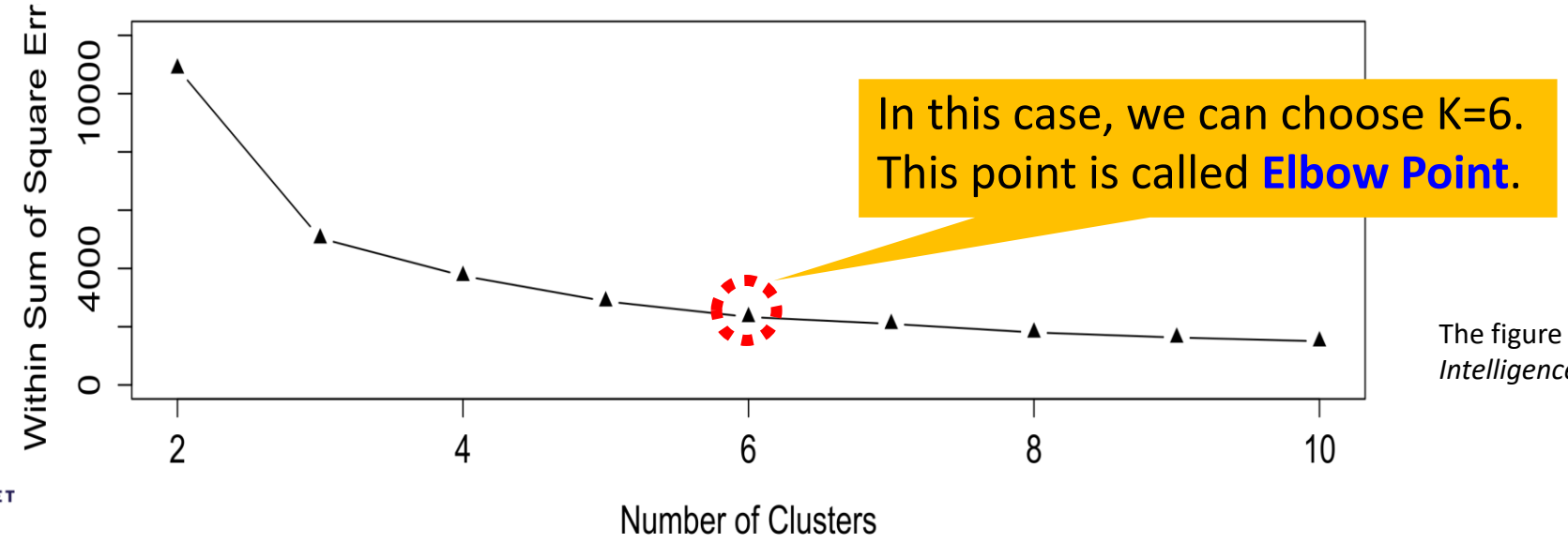
Note on K

- 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 *p*s 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

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

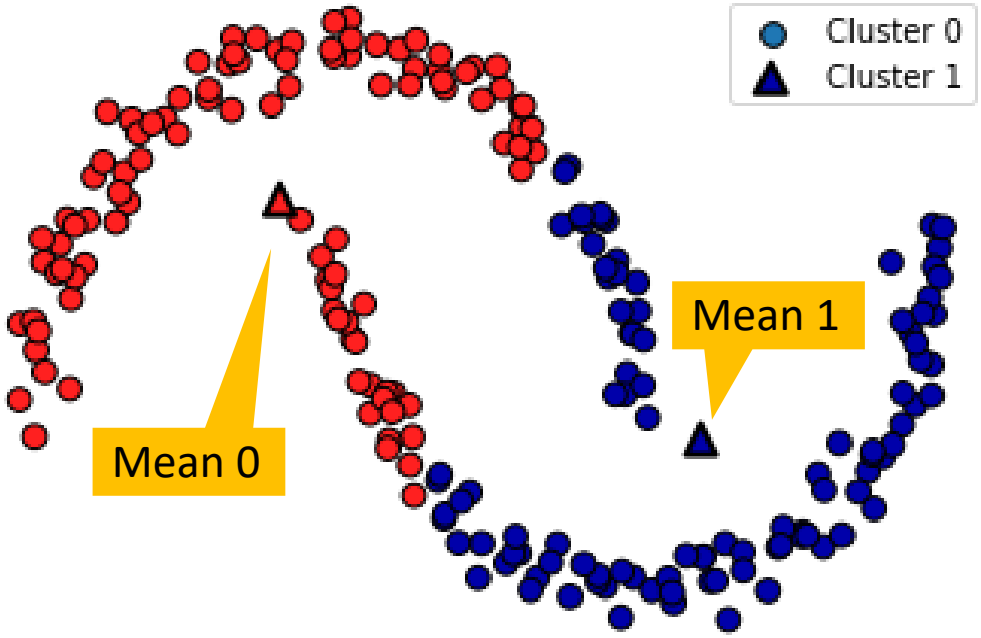
Exercises --- using the same set of numbers:

- 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

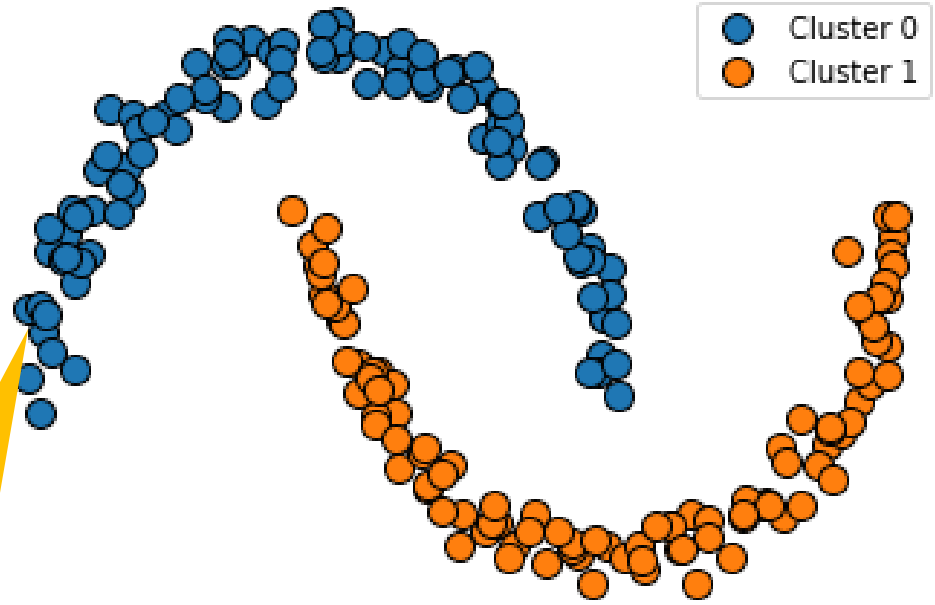
- Applicable only to objects in a *continuous* n-dimensional space
  - We cannot calculate means on categorical values, e.g., {CPH, RO, AAL}
- Initialization matters. Need to specify K, the number of clusters, in advance
  - In literature, there are ways to automatically determine the best k
- Convergence
  - 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



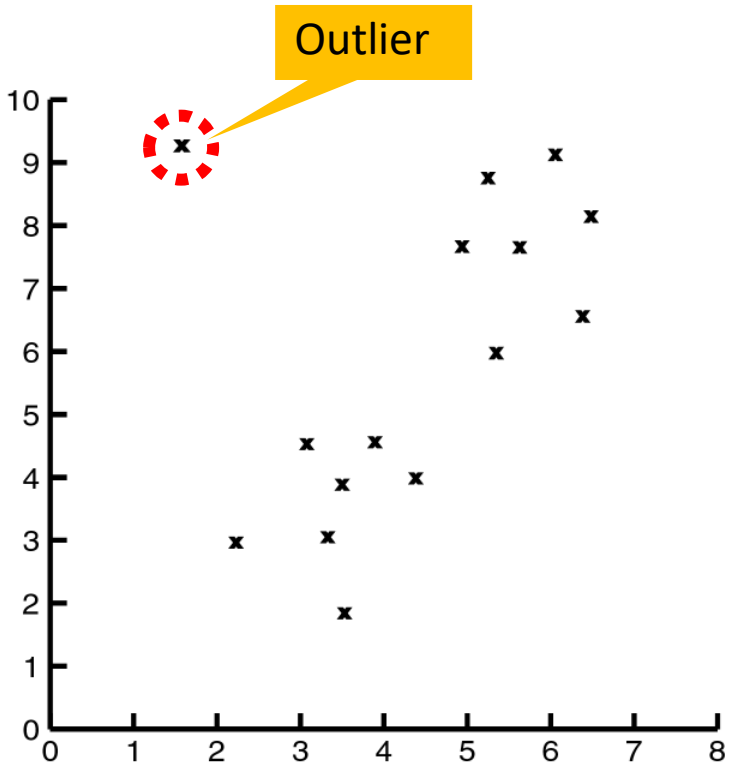
K-means clustering result (K=2)

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

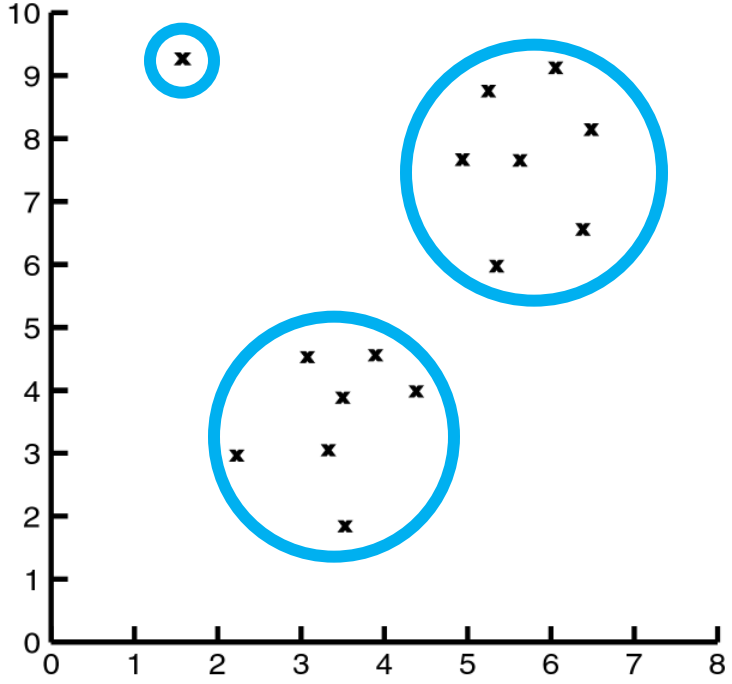


Desired clustering result

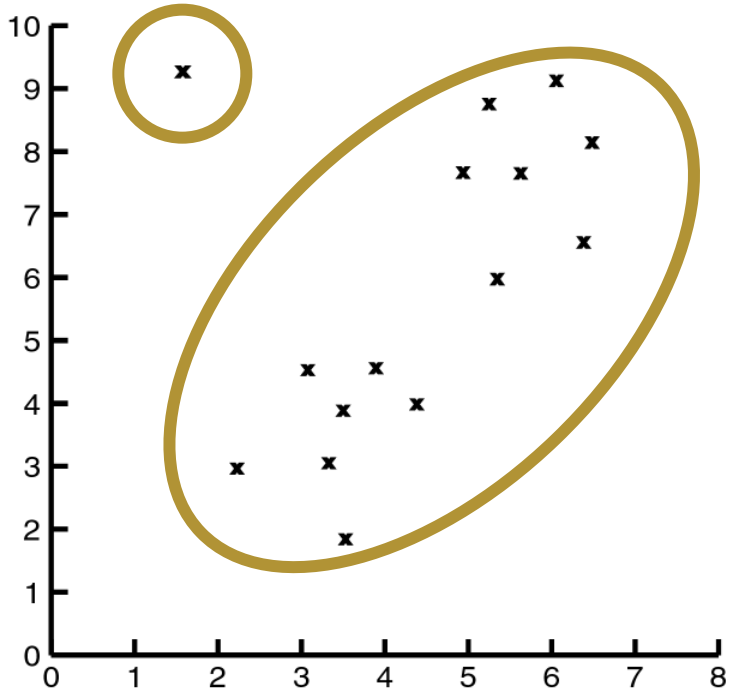
Impact of Outliers on k-Means



Dataset with outlier



K=3



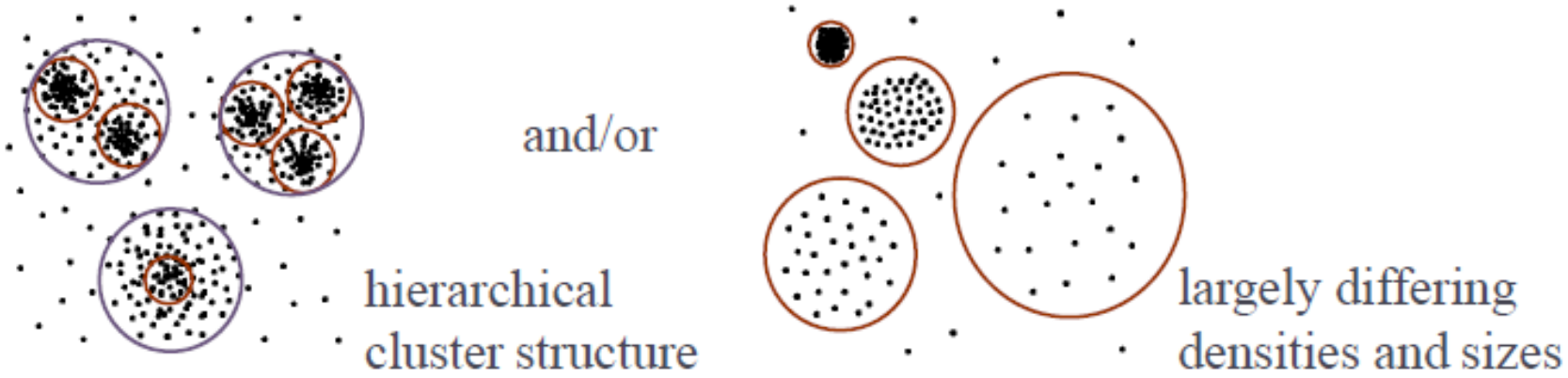
K=2

Agenda

- Clustering problem
- k-Means
- Hierarchical clustering
- DBSCAN
- Evaluation of clustering

Why Hierarchical Clustering?

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

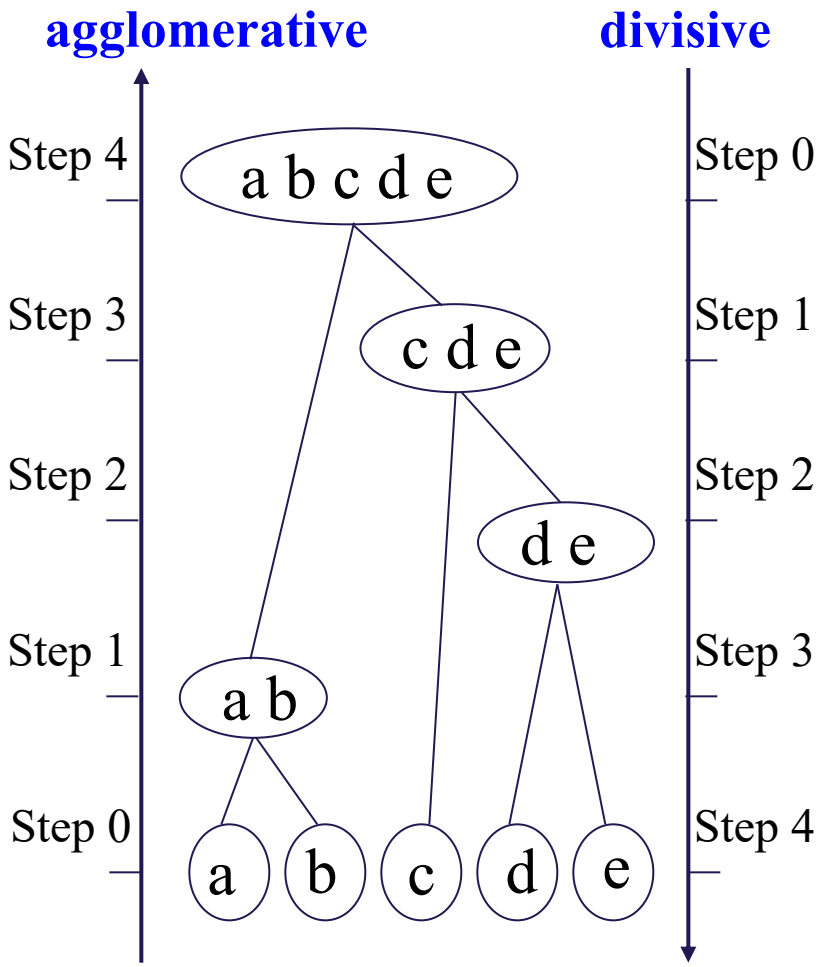


- Hierarchical clustering can handle such situations.
  - Clusters are created in *levels*, actually creating sets of clusters at each level.

Hierarchical Clustering Approaches

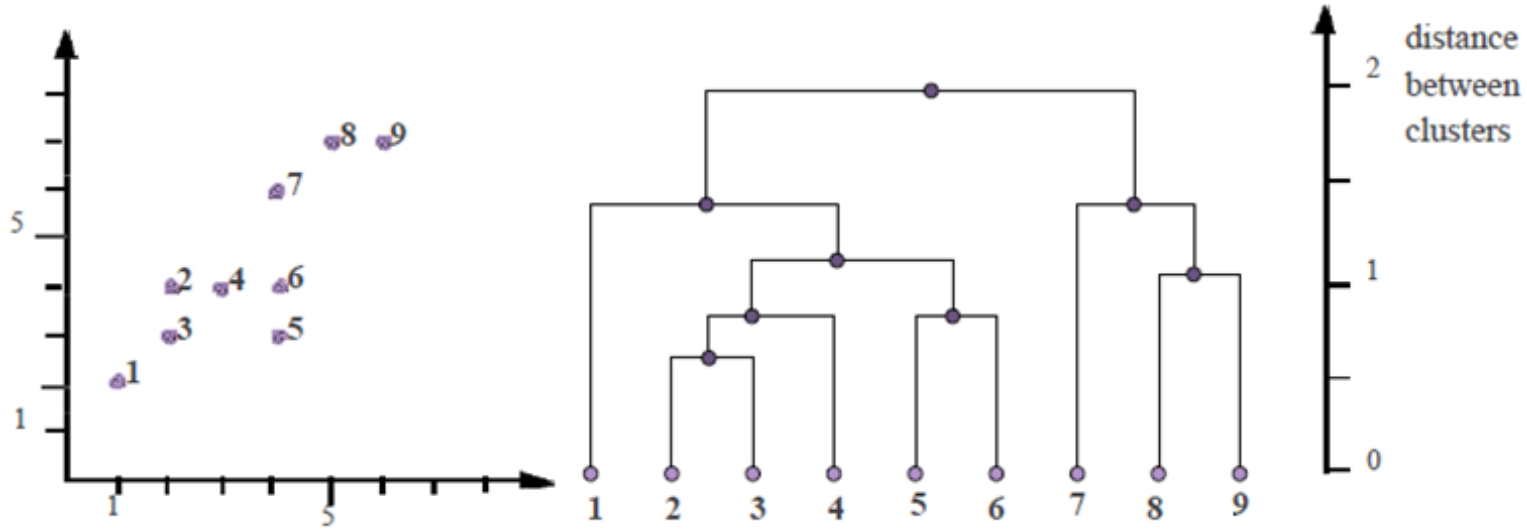
- HC uses distance matrix as clustering criteria. It does not require the number of clusters as an input, but needs a termination condition.

- 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

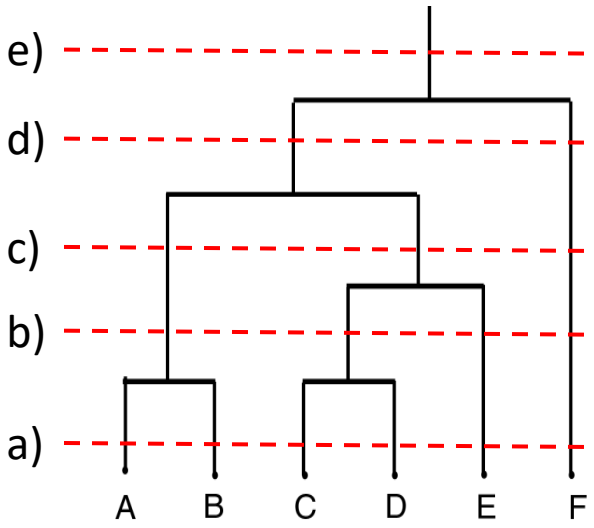


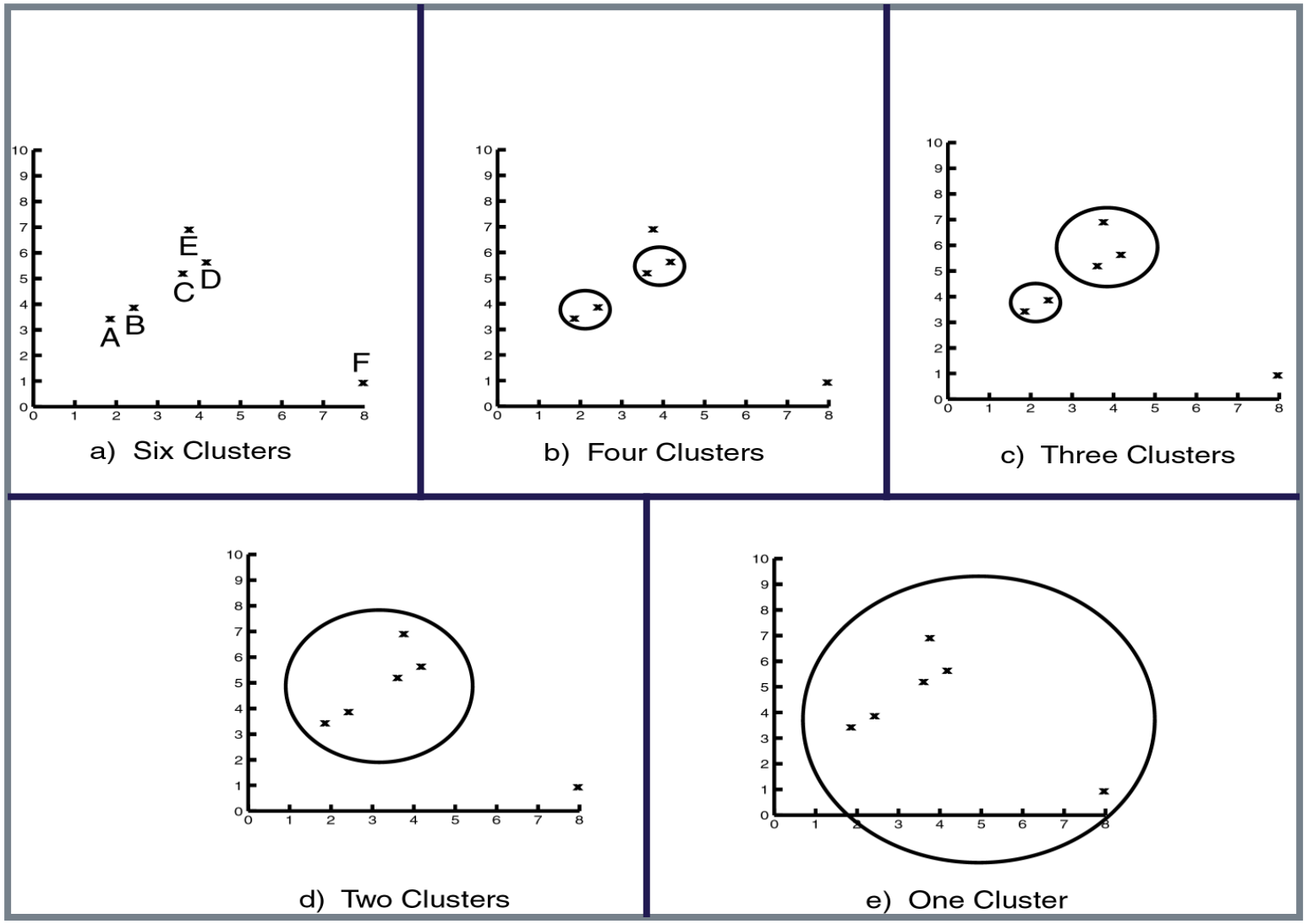
Dendrogram

- **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)





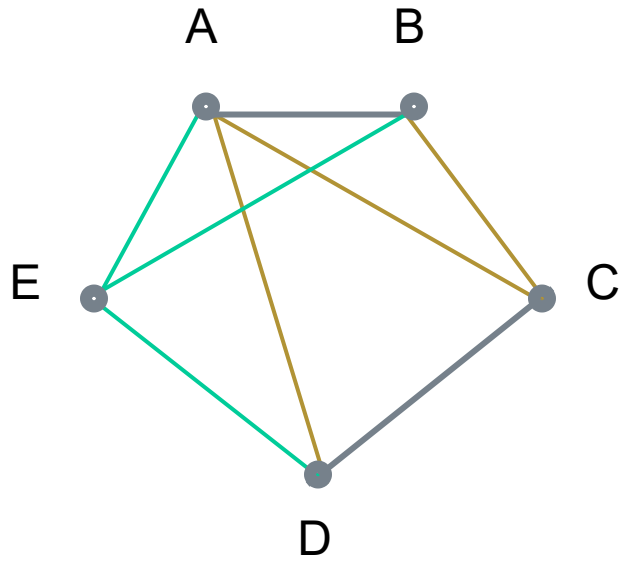
Agglomerative Clustering Algorithm

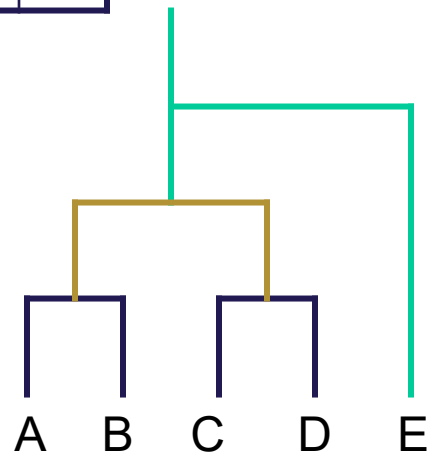
- Most popular hierarchical clustering technique
- Basic algorithm:
  1. Compute an adjacency matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge two clusters if the distance is small enough
  5. Update the adjacency matrix and distance threshold
  6. **Until** only a single cluster 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.



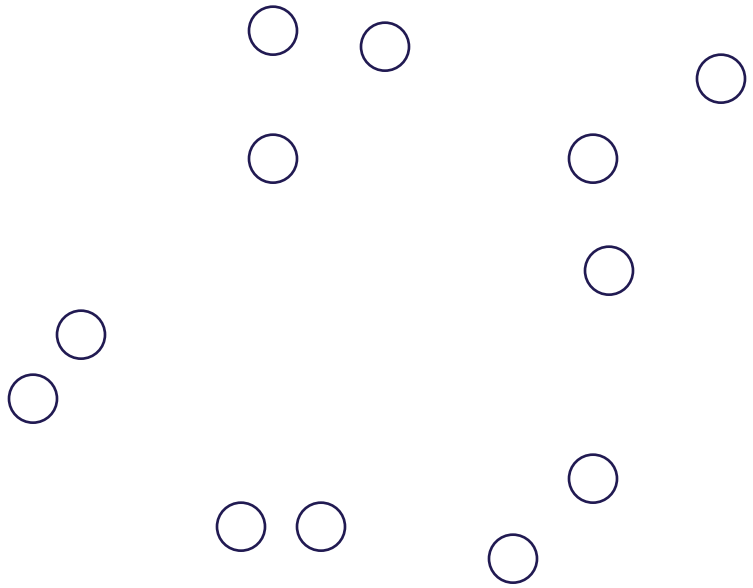


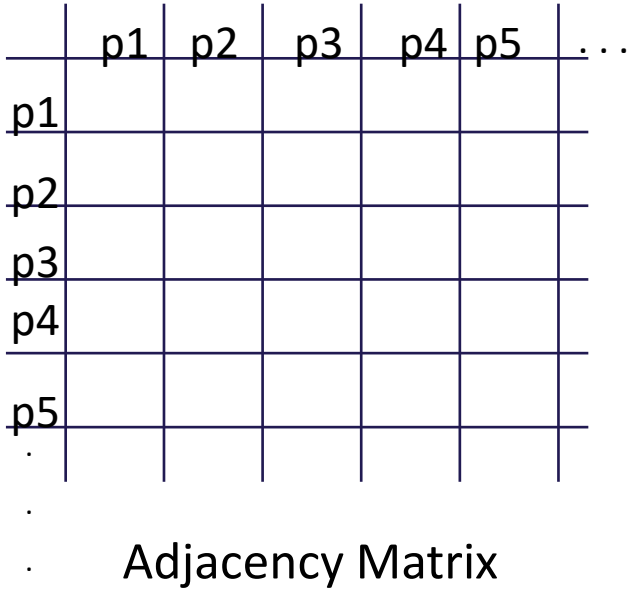
Distance threshold (for *similarity*)

1 2 3

Starting Situation

- Start with clusters of individual points and an adjacency matrix

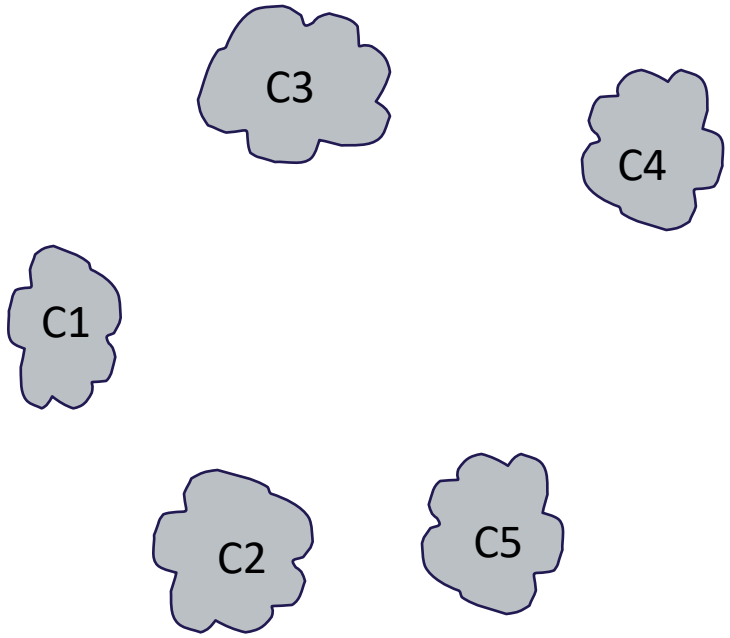




Adjacency Matrix

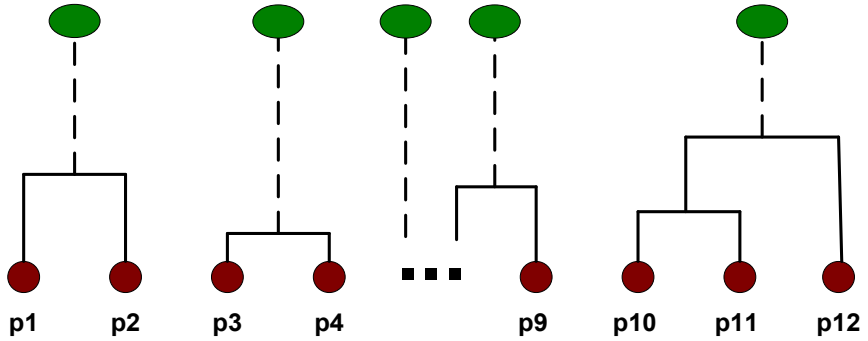
Intermediate Situation

- After some merging steps, we have some clusters



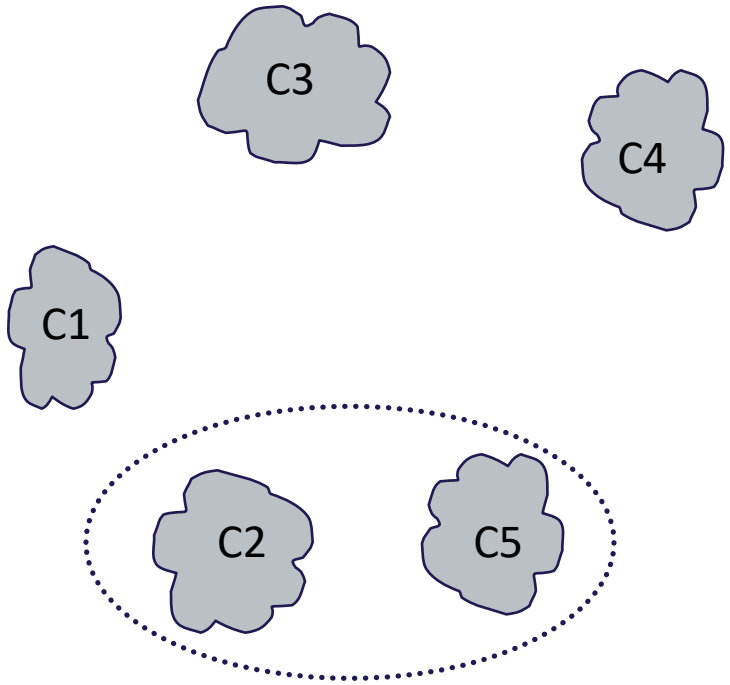
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

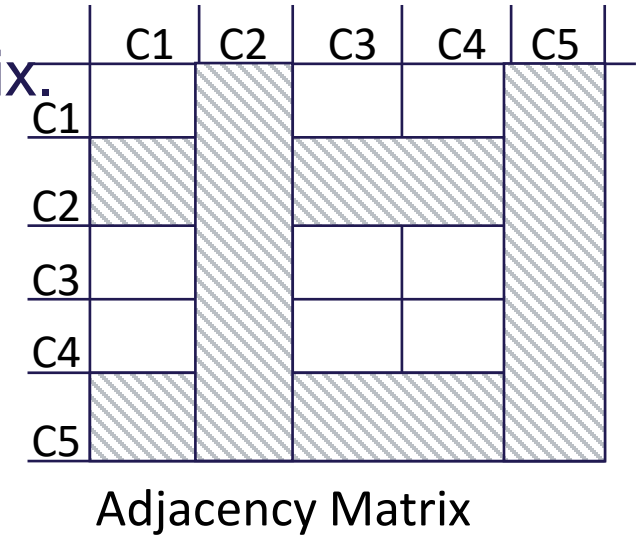
Adjacency Matrix



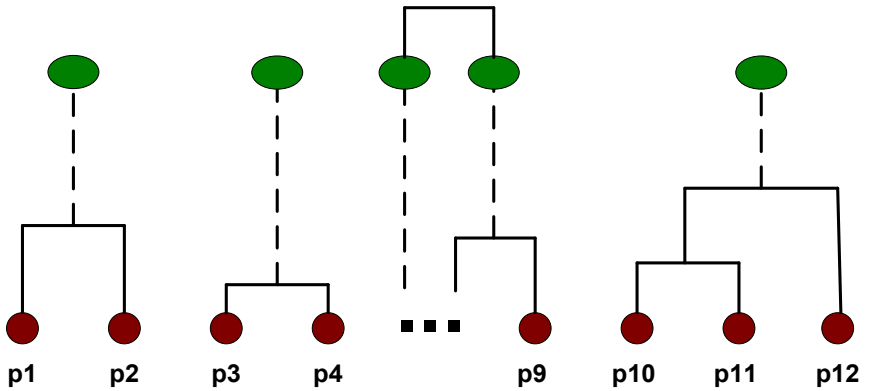
Intermediate Situation (cont.)

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



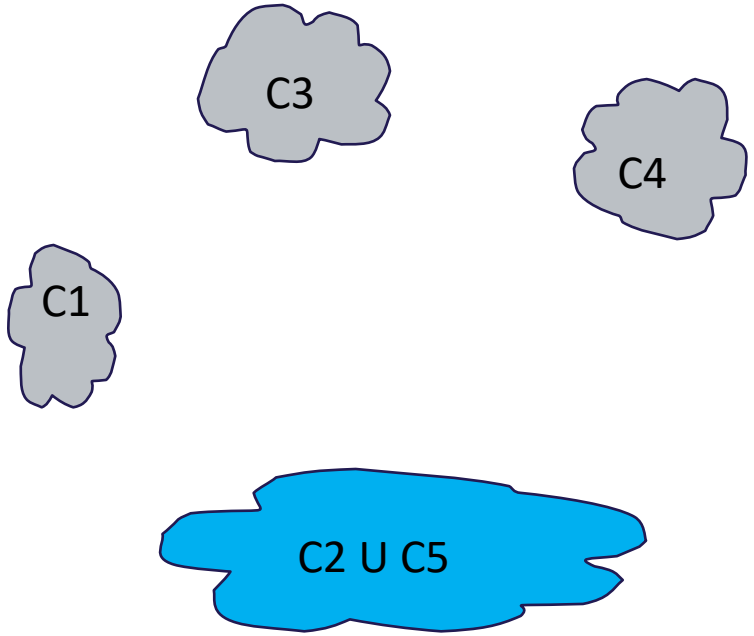


Adjacency Matrix



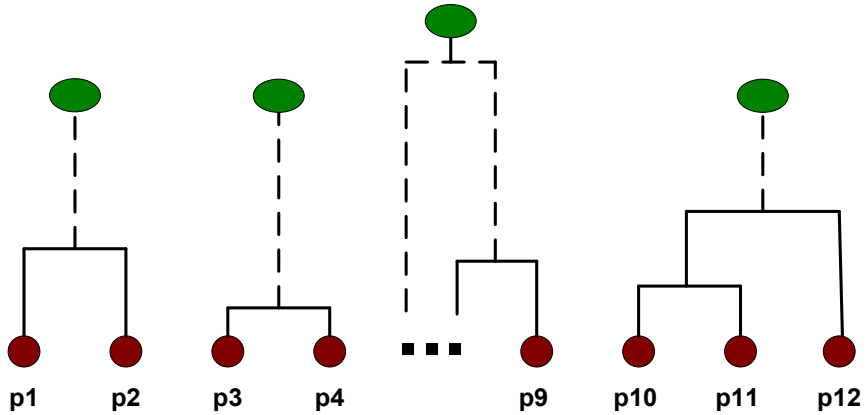
After Merging

- How to update the adjacency matrix?

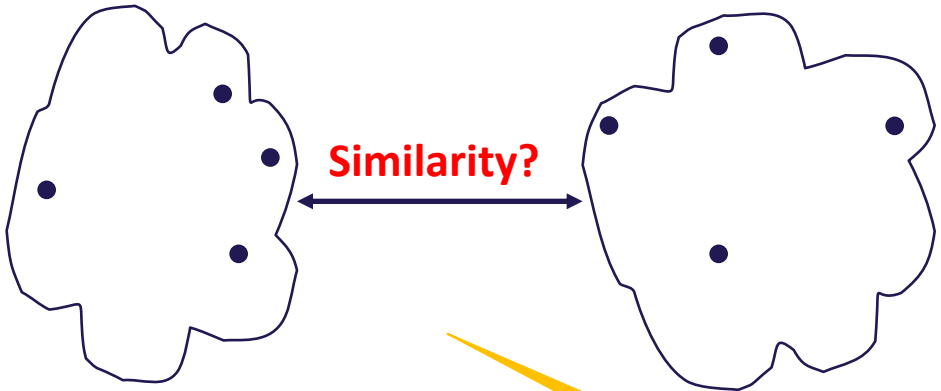


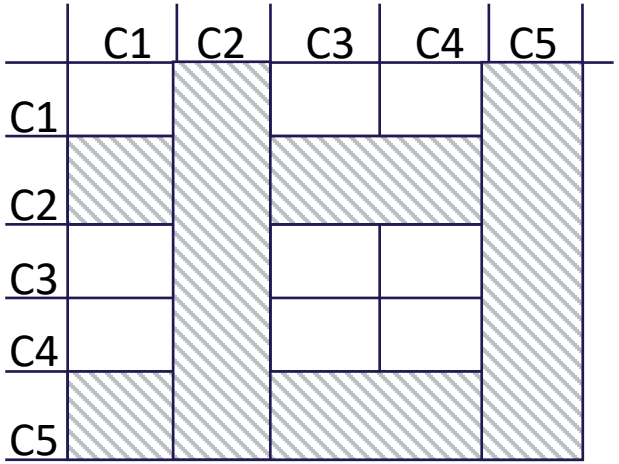
	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity?



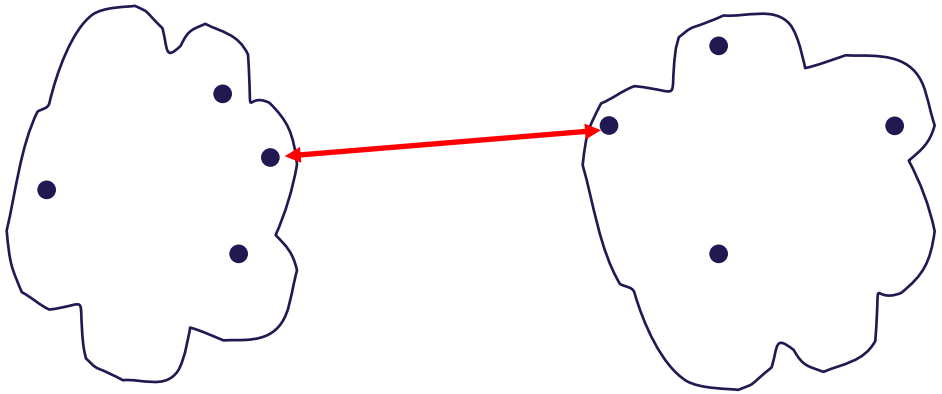


Adjacency Matrix

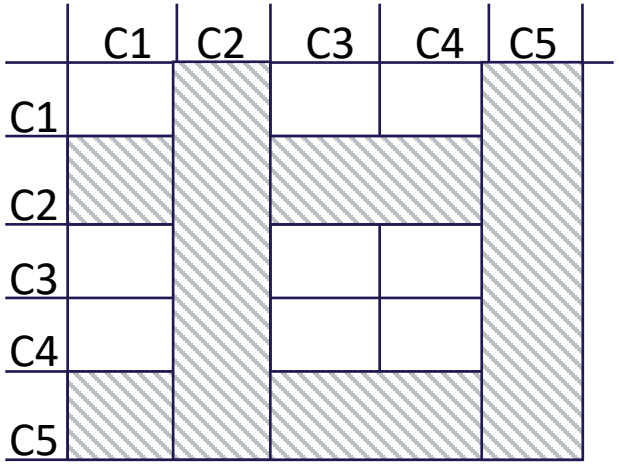
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Objective function

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

How to Define Inter-Cluster Similarity?

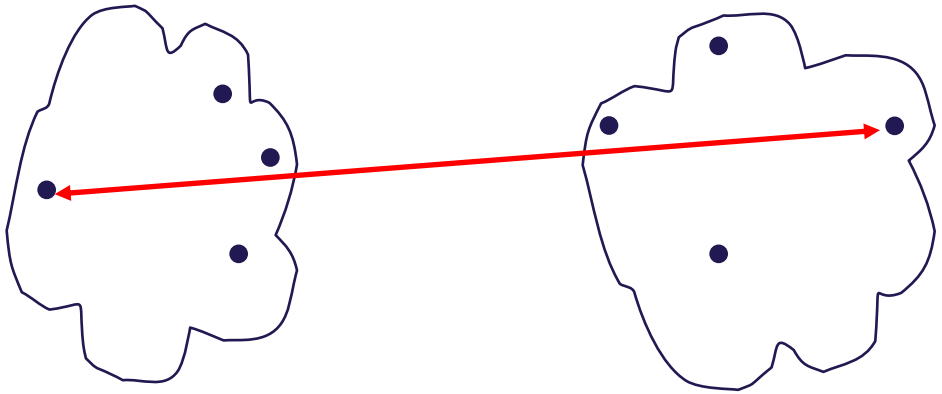


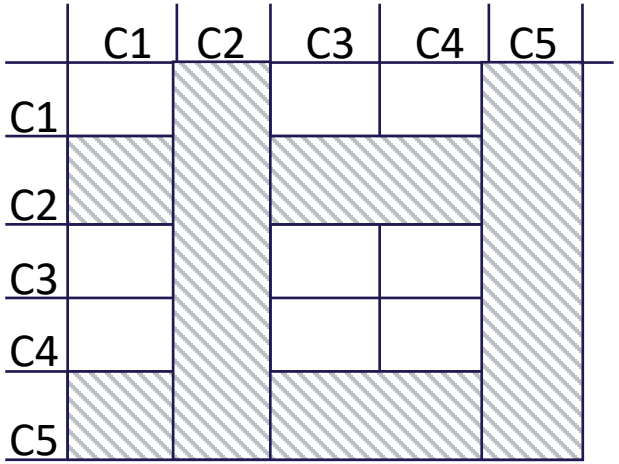
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Objective function



Adjacency Matrix

How to Define Inter-Cluster Similarity?

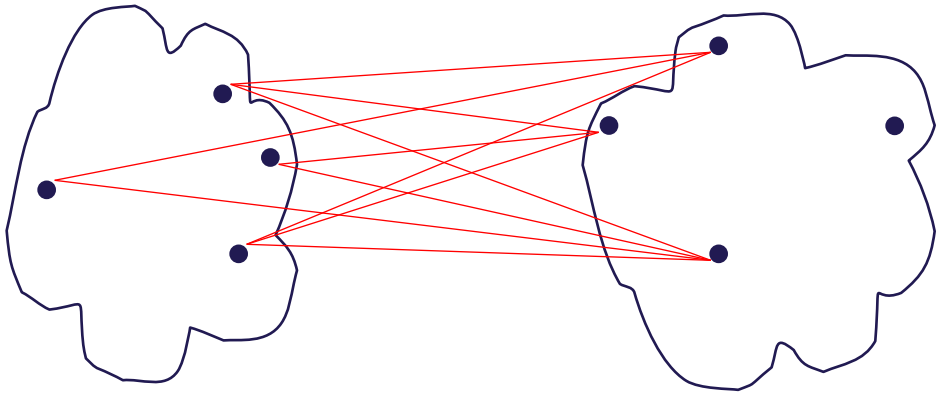


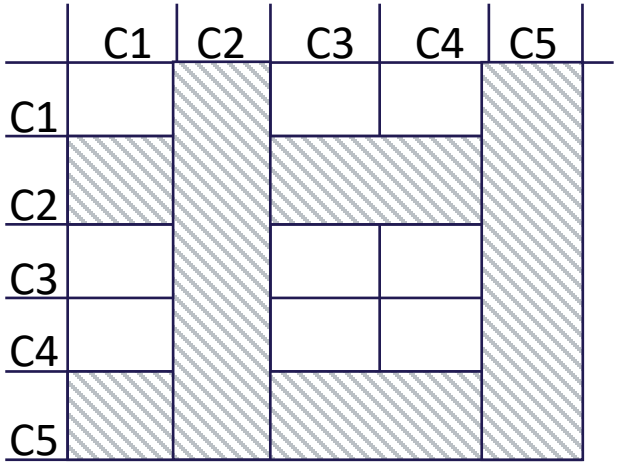


Adjacency Matrix

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

How to Define Inter-Cluster Similarity?

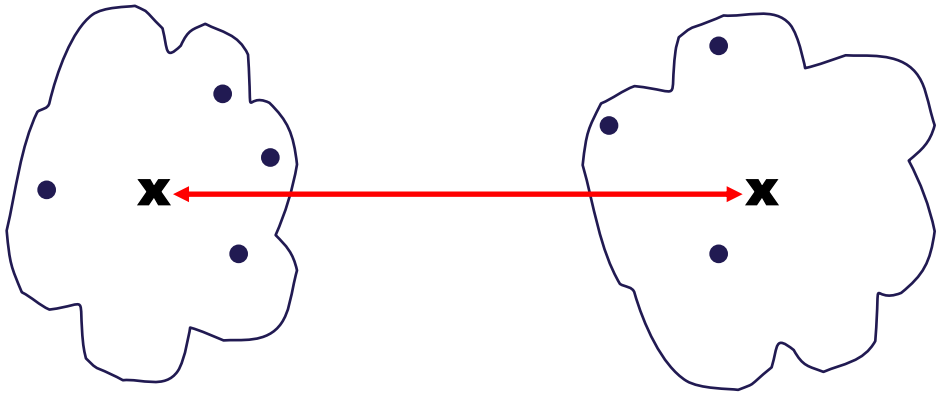


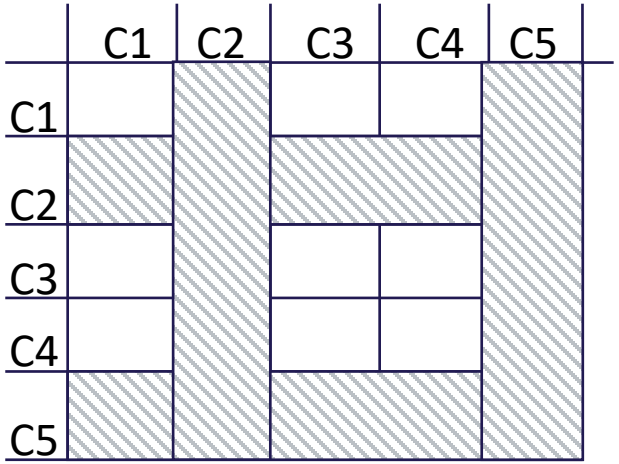


Adjacency Matrix

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

How to Define Inter-Cluster Similarity?





Adjacency Matrix

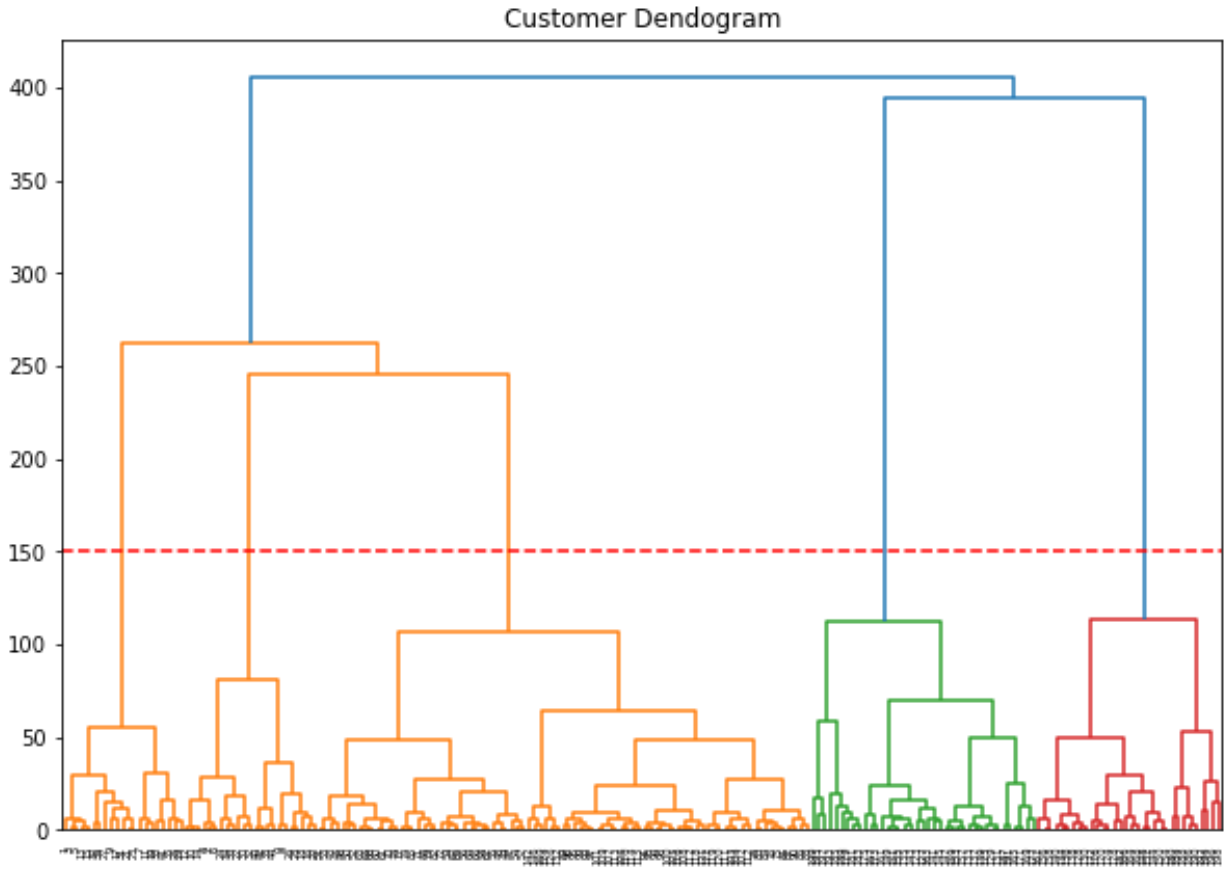
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Objective function

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.

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

Agenda

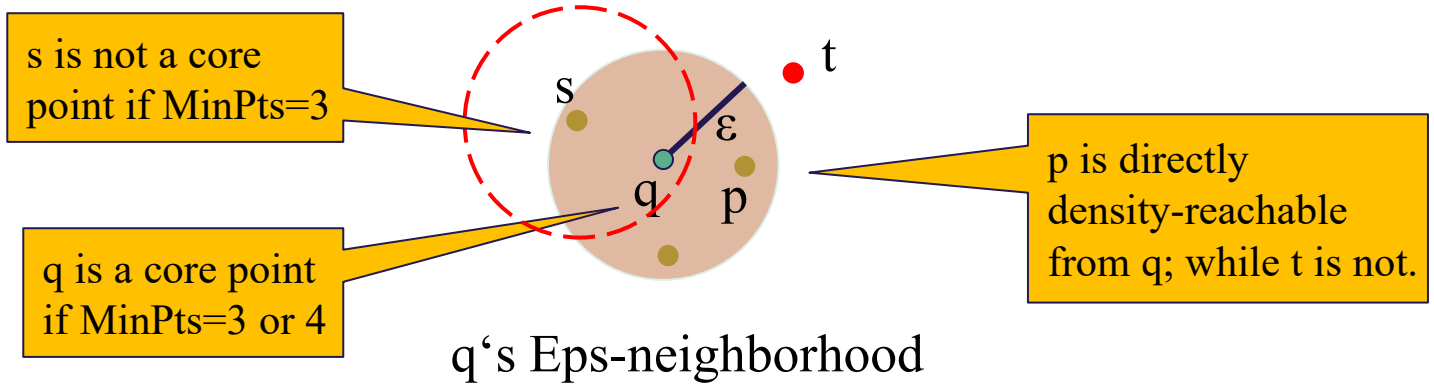
- Clustering in general
- k-Means
- Hierarchical clustering
- DBSCAN
- Evaluation of clustering

DBSCAN

- Density Based Spatial Clustering of Applications with Noise
- Outliers will not effect creation of clusters.
- Algorithm parameters (hyperparameters)
  - **MinPts** – minimum number of points in a cluster
    - Size of a cluster (number of points)
    - **min_samples** in sklearn.cluster.DBSCAN
  - **Eps** – for each point in a cluster there must be another point in it less than this distance away.
    - Distance between points
    - **eps** in sklearn.cluster.DBSCAN

DBSCAN Concepts (1)

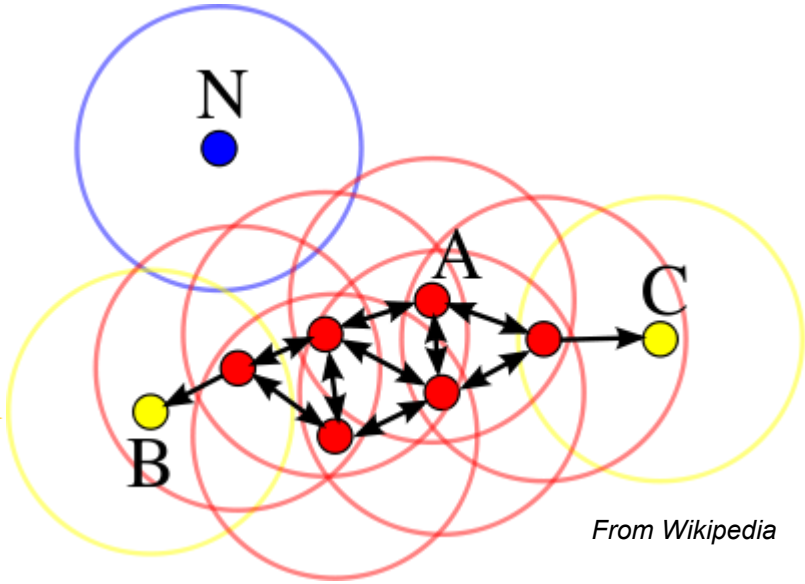
- **Eps-neighborhood**
  - Covers all points within Eps distance of a point.
- **Core point**
  - Whose Eps-neighborhood is dense enough (with at least MinPts points)
- **Directly density-reachable**
  - A point *p* is directly density-reachable from another point *q* if the distance is small (≤ Eps) and *q* is a core point.



DBSCAN Concepts (2)

- **Density-reachable**: A point *p* is density-reachable from another point *q* if there is a *path* from *q* to *p* and the path consists of only core points.
  - I.e., if there is a chain of points $p_1=q$, p_2, …, $p_n=p$ such that p_{i+1} is directly density-reachable from p_i. More specifically,
    1. p_1, …, p_{n-1} are core points;
    2. the distance between each pair ≤ Eps;
    3. *p* may not be a core point.
  - Density-reachable is *not* symmetric.
    - A is not density-reachable from B or C as they are not core.

Assume MinPts=3.
- Red points are core points.
- Points B and C are *density-reachable* from A.
- Point B is not density-reachable from C; and vice versa.

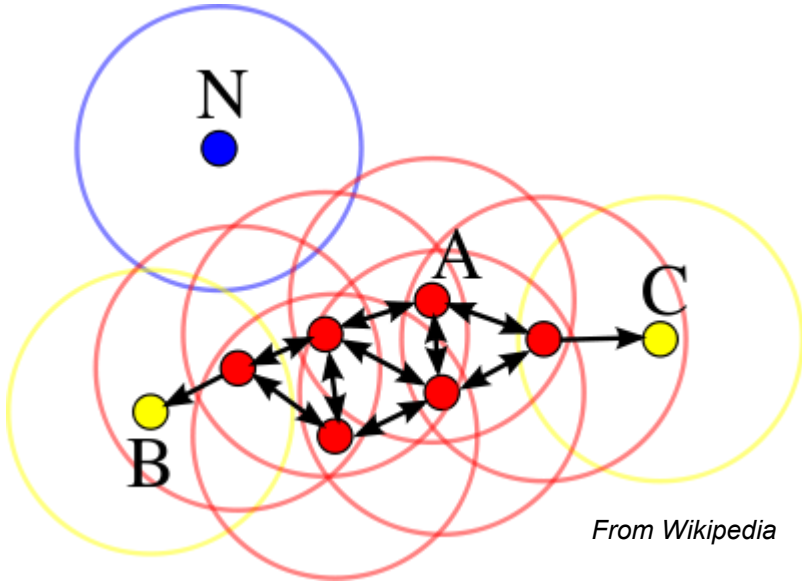


From Wikipedia

DBSCAN Concepts (3)

- **Density-connected**: two points *p* and *q* are density-connected if there is a point *o* such that both *p* and *q* are density-reachable from *o*.
  - B and C are density-connected (via A).
  - Density-connected is symmetric.

- Clusters in DBSCAN
  - A cluster contains at least MinPts points
  - Density-connected points go to the same cluster
    - E.g., all red points plus B and C
- Outliers in DBSCAN
  - Those points not in any cluster



From Wikipedia

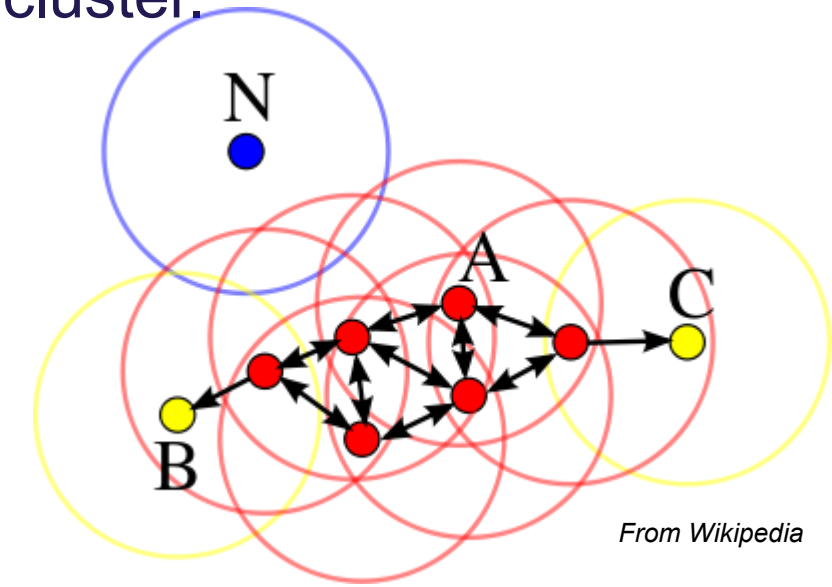
DBSCAN Algorithm

```
DBSCAN(D, eps, MinPts)
   C = 0
   for each unvisited point P in dataset D
      mark P as visited
      NeighborPts = regionQuery(P, eps)
      if sizeof(NeighborPts) < MinPts
         mark P as NOISE
      else
         C = next cluster
         expandCluster(P, NeighborPts, C, eps, MinPts)

expandCluster(P, NeighborPts, C, eps, MinPts)
   add P to cluster C
   for each point P' in NeighborPts
      if P' is not visited
         mark P' as visited
         NeighborPts' = regionQuery(P', eps)
         if sizeof(NeighborPts') >= MinPts
            NeighborPts = NeighborPts joined with NeighborPts'
      if P' is not yet member of any cluster
         add P' to cluster C

regionQuery(P, eps)
   return all points within P's eps-neighborhood
```

From Wikipedia

DBSCAN Properties

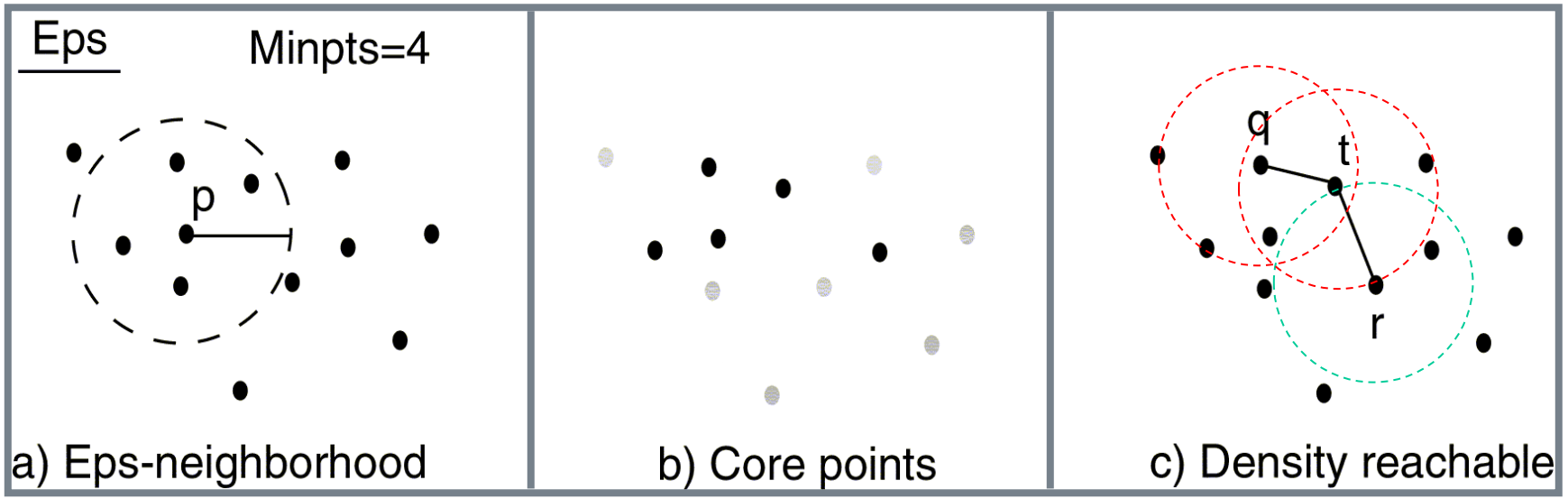
- A cluster satisfies two properties:
  - All points within a cluster are mutually density-connected.
  - If a point *p* is density-connected to any point of a cluster, *p* belongs to the same cluster as well.
- In this example, point N is not included in any cluster. It is a *noise point*, neither a core point nor density-reachable.



From Wikipedia

Another DBSCAN Example

- Point r is not a core point but it is in the Eps-neighborhood of core point t
- Point r is density reachable from q, not vice versa.



Agenda

- Clustering in general
- k-Means
- Hierarchical clustering
- DBSCAN
- Evaluation of clustering

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters
  - high *intra-cluster* similarity: **cohesive** within clusters
  - low *inter-cluster* similarity: **distinctive** between clusters
- The quality of a clustering method depends on
  - the similarity measure used by the method
  - its implementation (e.g., hyperparameters), and
  - its ability to discover *some* or *all* of the hidden patterns

Evaluation of Clustering in Scikit-Learn

- If clustering groundtruth is available
  - Compare the clustering result with the groundtruth by measuring a score
    - Adjusted Rand Index (**ARI**): adjusted_rand_score(groundtruth, clustering_result)
    - Normalized Mutual Information (**NMI**): normalized_mutual_info_score(groundtruth, clustering_result)
- Otherwise
  - **Silhouette score**
  - silhouette_score(X, clustering_results) computes the *compactness* of a cluster
- All scores are in sklearn.metrics.cluster
  - The higher a score is, the better the clustering result.

Rand Index (William M. Rand 1971)

Advanced

- A set $S = \{o_1, \ldots, o_n\}$. Two partitions: $X = \{X_1, \ldots, X_r\}$ and $Y = \{Y_1, \ldots, Y_r\}$
  - a: #pairs of elements in S that are in the same X_i and in the same Y_j
  - b: #pairs of elements in S that are in different X_is and in different Y_js
  - c: #pairs of elements in S that are in the same X_i but in different Y_js
  - d: #pairs of elements in S that are in different X_is but in the same Y_j
- Rand Index $\boldsymbol{R} = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$, where $\binom{n}{2} = \frac{n(n-1)}{2}$ (binomial coefficient)
  - A value between 0 and 1.
  - 0: the two clusterings do not agree on any pair of points.
  - 1: the two clusterings are exactly the same.
- Example
  - Dataset: {A, B, C, D, E}
  - Method 1 Clusters: {{A, B, C}, {D, E}} , Method 2 Clusters: {{A, B}, {C, D}, {E}}
  - a=1: {A, B}; b=5: {A, D}, {A, E}, {B, D}, {B, E}, {C, E}; a+b+c+d=$\binom{5}{2}$=10
  - R = (1+5)/10 = 0.6

Adjusted Rand Index

- A set $S = \{o_1, \ldots, o_n\}$. Two partitions: $X = \{X_1, \ldots, X_r\}$ and $Y = \{Y_1, \ldots, Y_r\}$
- **The contingency table**:
  - Each entry denotes the number of objects in *common* between X_i and Y_j $\quad n_{ij} = |X_i \cap Y_j|$

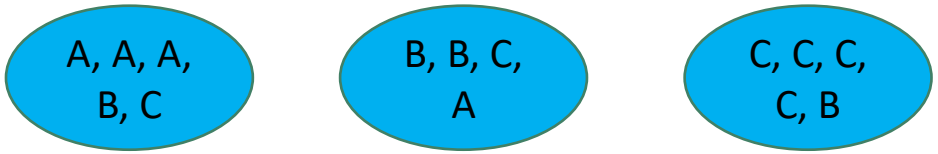
$X \backslash Y$	Y_1	Y_2	$\cdots$	Y_s	sums
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
sums	b_1	b_2	$\cdots$	b_s	

- Adjusted Rand Index

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}$$

Purity Score

- If we have the groundtruth for clustering, the best case would be that each ‘predicted’ cluster contains only objects from the same groundtruth cluster.
  - For each cluster, we ‘label’ it with the most frequent ‘groundtruth’ cluster ‘label’.

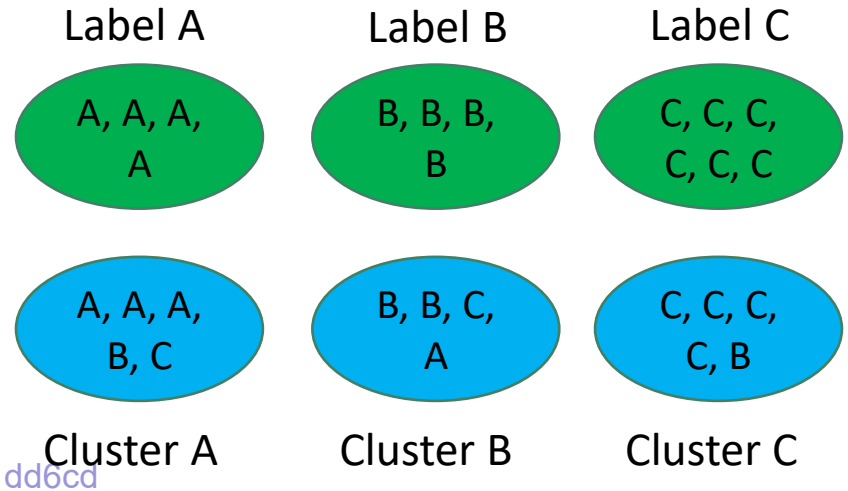


  - **Purity Score** is the average number of ‘correct’ cluster labels cross all clusters.
  - In this example, Purity = (3+2+4)/(5+4+5) = 9/14 = 0.642
- However, if we put each object in its own singleton cluster, we will always get Purity maximized to 1! Therefore, we need to take into account the number of clusters as well.

Normalized Mutual Information

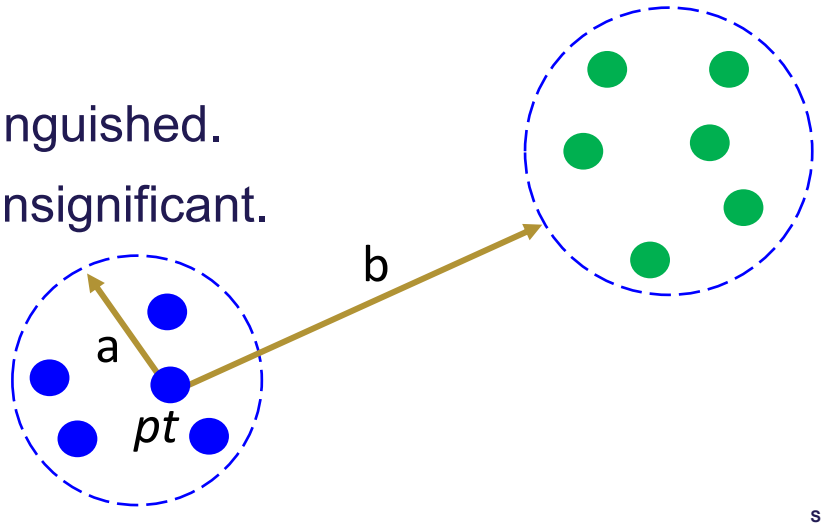
Advanced

- Consider the groundtruth as Y, and the clustering result as C
- Mutual Information tells how Y and C, as two splits, *agree* with each other
  - how much information they share about each other, or how can you know about one of them if you know the other one
  - I(Y; C) = entropy(Y) – entropy(Y|C)
- Normalized Mutual Information
  - NMI(Y, C) = 2*I(Y; C) / (entropy(Y)+entropy(C))
- Entropy is a measure that quantifies uncertainty.
  - Entropy(S) = $-\Sigma p_i * \log_2(p_i)$
  - entropy(Y|C):
    - conditional entropy of labels given the clustering result C
- For more details of entropy and NMI
  - https://course.ccs.neu.edu/cs6140sp15/7_locality_cluster/Assignment-6/NMI.pdf
  - https://towardsdatascience.com/evaluation-metrics-for-clustering-models-5dde821dd6cd



Silhouette Score

- Silhouette score for one point *pt*
  - s(*pt*) = (b - a) / max(a, b)
  - a: the average distance between *pt* and all others in the same cluster (**cohesive**)
  - b: the smallest average distance between *pt* and all points in any other cluster (**distinctive**)
- Silhouette score for a clustering result *X*
  - s(*X*) = (b̄ - ā) / max(ā, b̄)
    - ā, b̄: Average a and b for all points in the dataset
  - 1: Clusters are well apart from each other and clearly distinguished.
  - 0: Clusters are indifferent. The distance between them is insignificant.
  - -1: Clusters are assigned in the wrong way.
- Used when groundtruth is *unavailable*



Applications of Clustering

- **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
- **City-planning**: Identifying groups of houses according to their house type, value, and geographical location
- **Earth-quake studies**: Observed earth quake epicenters should be clustered along continent faults
- **Climate**: understanding earth climate, find patterns of atmospheric and ocean
- **Economics**: market research

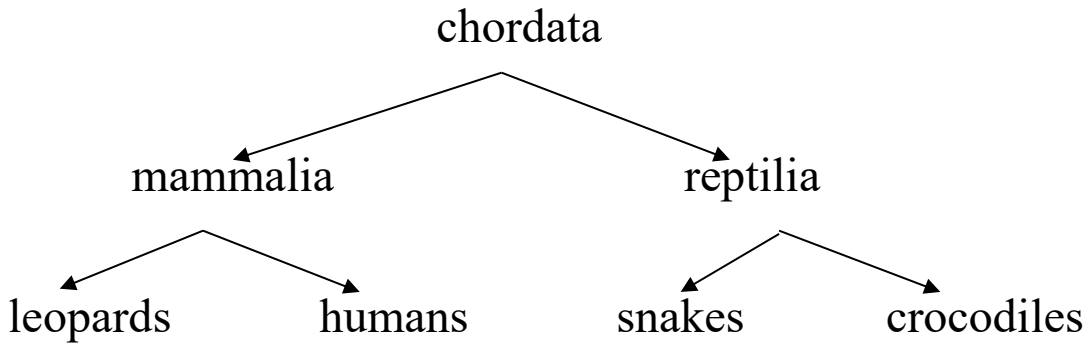
How to choose a clustering method?

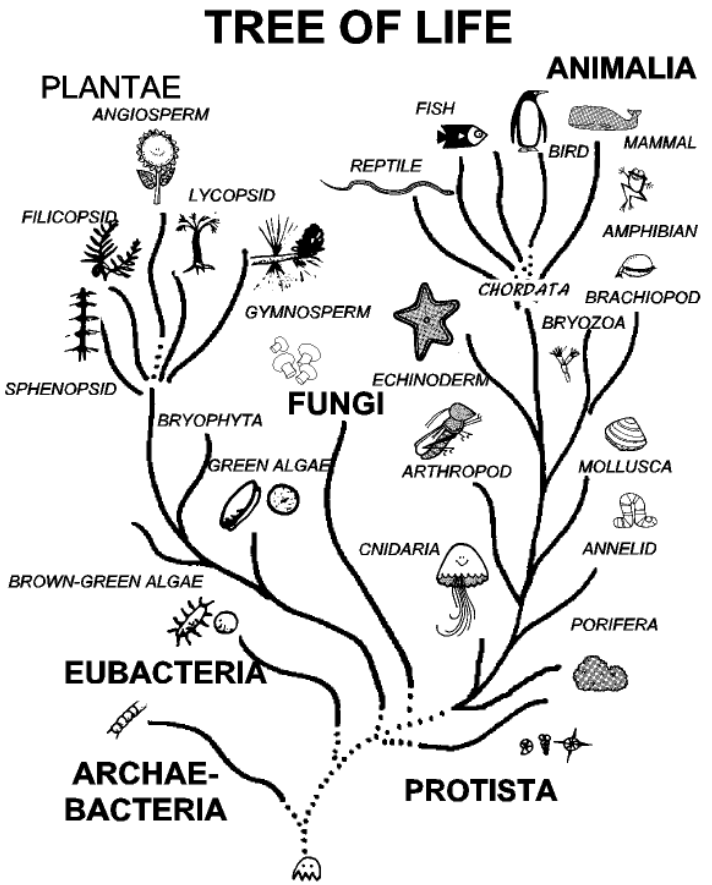
- K-means
  - Only applicable to continuous domains
  - Need to specify k
  - Unsuitable for non-convex shapes
- Agglomerative (hierarchical)
  - If your data is hierarchical
  - If you don’t know how many clusters you should have
- DBSCAN (density based)
  - If your data contains noise or your resulted cluster can be of arbitrary shapes
  - If you want to be able to isolate outliers
- Ask yourself: Which method fits your data best?

Taxonomy

- Biology taxonomy

Which clustering method to use for these datasets?

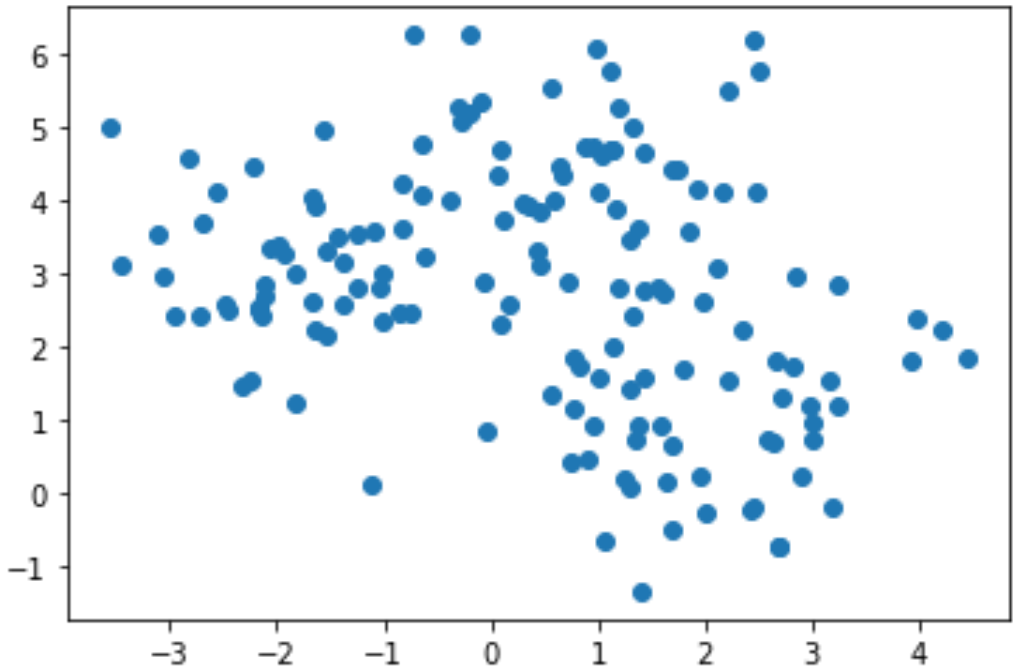




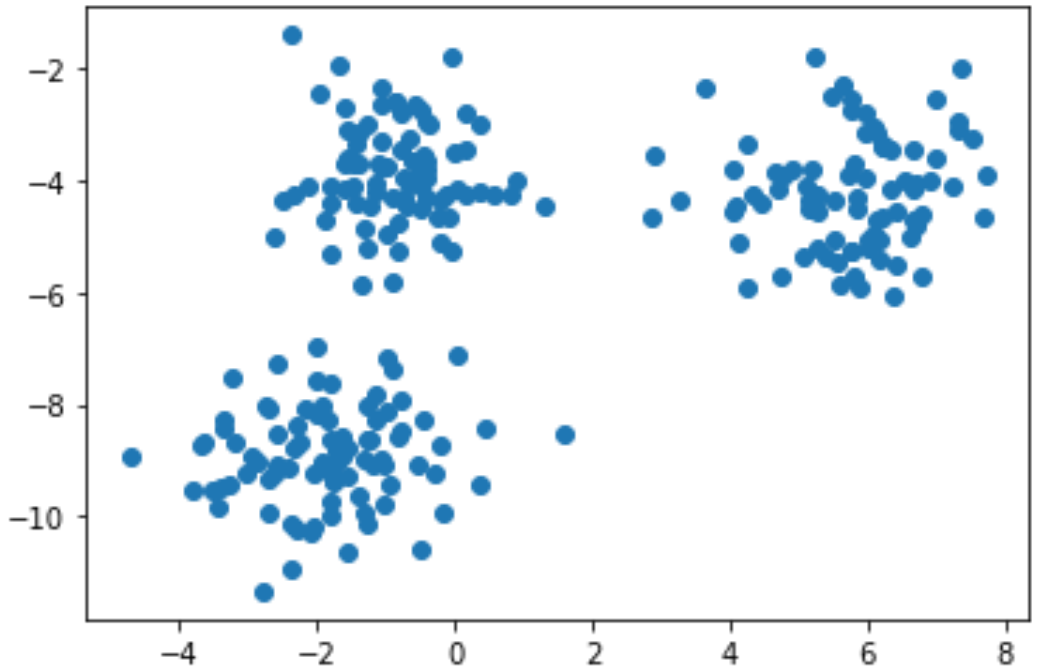
https://www.msnucleus.org/membership/html/k-6/lc/organ/6/lco6_3a.html

Random distributions

- A single random distribution



- Multiple distributions



Which clustering method to use for these datasets?

Special shapes: DBSCAN vs K-means

- Which is by which?

https://github.com/NSHipster/DBSCAN

Last column: https://towardsdatascience.com/understanding-dbscan-and-implementation-with-python-5de75a786f9f

Summary

- Clustering Problem
  - Comparison with classification
- Clustering techniques
  - K-Means clustering
  - Agglomerative clustering
  - Dendrogram
  - DBSCAN
- Clustering evaluation
  - Elbow method
  - Metrics
  - Clustering method choosing

References

- Mandatory reading
  - Jiawei Han, Micheline Kamber, and Jian Pei (Data Mining: Concepts and Techniques, 3rd Edition 2011): Chapter 10

- Further readings
  - https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/
  - Documentations
    - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
    - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html
    - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

Exercises (1)

1. K-means example on page 16
2. Use Euclidean distance based DBSCAN to decide the clustering result for the following data set.
   - A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).
   - MinPts is 2 and Eps is also 2.

- Draw the dendrogram using the *complete* linkage agglomerative clustering for the given distance matrix. Use the *smallest* distance as the merge criterion.

	A	B	C	D
A	0	1	4	6
B		0	2	5
C			0	3
D				0

Exercises (2, hands-on, optional)

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
3. Apply DBSCAN clustering
   - Vary eps and min_samples
   - Visualize the DBSCAN clustering results with their Silhouette scores