Data & Things

(Spring 25)

Friday February 21

Lecture 10: Clustering

Jens Ulrik Hansen

Roskilde Universitet

Introduction to machine learning

- Supervised learning generalizes from *Labeled data* to facilitate future predictions of label based on input data.
  - Given a collection of feature variables $X_1, X_2, \ldots, X_p$, can we find a model that approximately predict a response variable Y?
  - Two stages: Training and prediction
  - **Classification**: Predict a discrete value from a *pre-defined* set of class labels
  - **Regression**: Predict a continuous value from a continuous range
- Unsupervised learning find patterns in *unlabeled data*. It works on input data only.
  - *We are only given the feature variable $X_1, X_2, \ldots, X_p$ and no response variable or true labels Y – we have to look for patterns in $X_1, X_2, \ldots, X_p$ without having an example of what we are looking for and thereby no obvious way of testing model performance*
  - Only one stage, work on the entire dataset
  - E.g., clustering, customer segmentation, outlier detection for website access patterns, etc.

Outline of this lecture

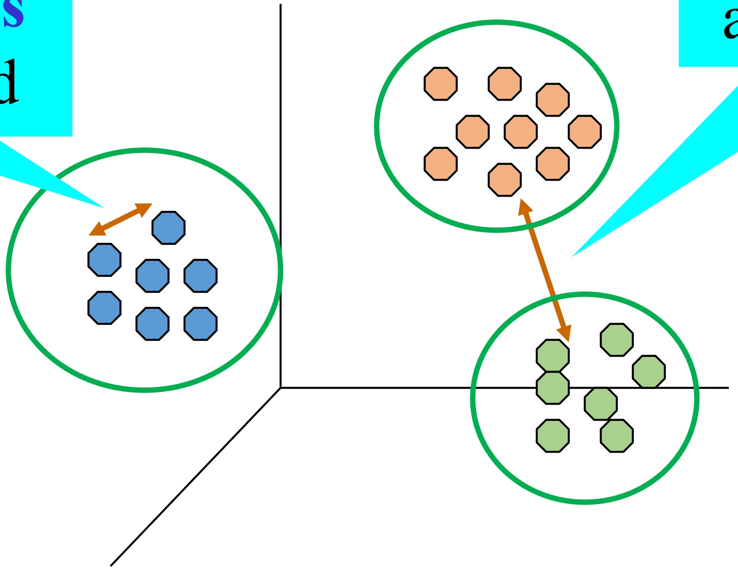
- Clustering as an example of unsupervised learning
- K-means clustering
- Hierarchical clustering
- DBSCAN clustering
- Evaluation of clustering models

Clustering as an example of unsupervised learning

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

Clustering as an example of unsupervised learning

- Another example (for two feature variables):

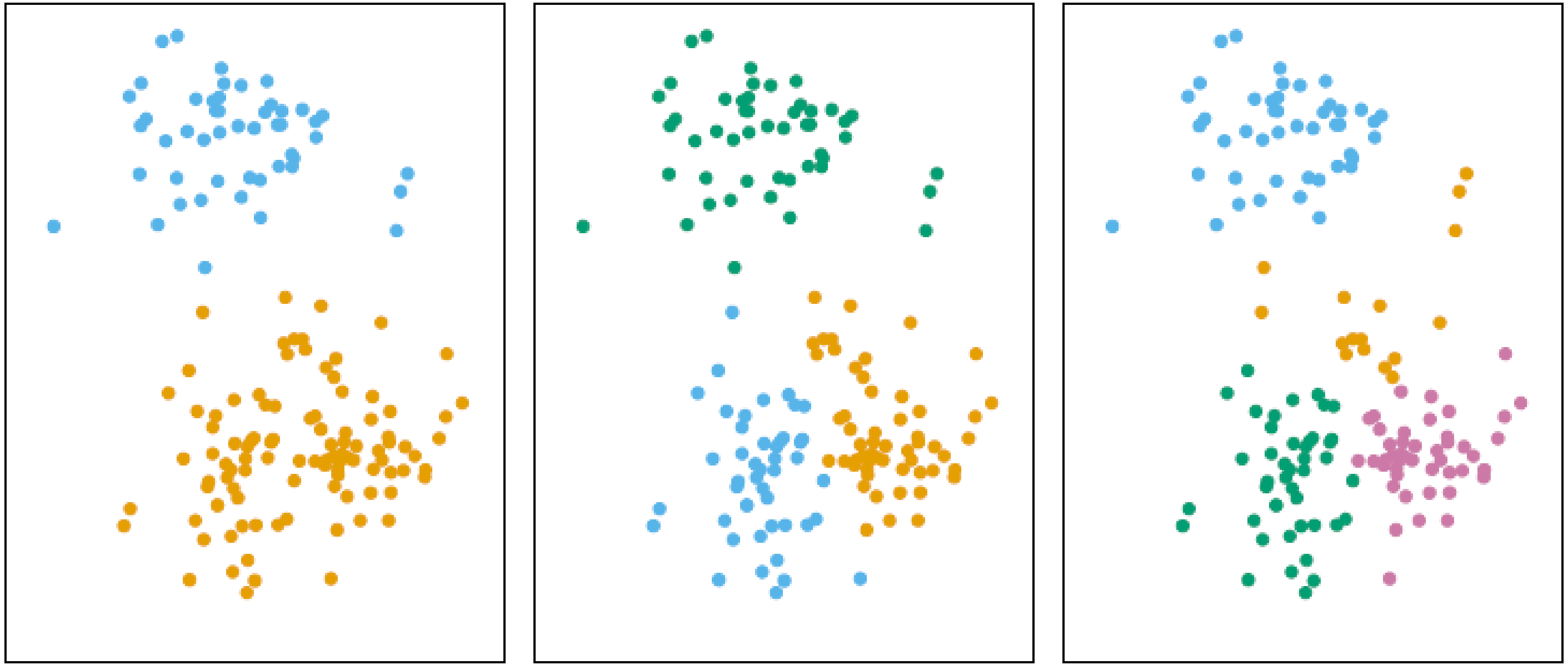


Figure 12.7 from James, G., Witten, D., Hastie, T., Tibshirani, R., and Taylor, J. (2023). *An Introduction to Statistical Learning – with Applications in Python*. Springer

Clustering as an example of unsupervised learning

- **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**: Compute similarities between data points and group similar data points into the same cluster.
  - 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***

Clustering as an example of unsupervised learning

Classification (supervised)

- Predefined classes
  - Number of classes
  - Meaning of classes
- Work for any number of points (in the prediction stage)
  - Given a point, a classifier (trained model) assigns it to a class

Clustering (unsupervised)

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

Clustering as an example of unsupervised learning

- **Examples of application of clustering**
  - *Market segmentation* (Customer segmentation): If we can group our costumers into groups of similar people, based on demographics or shopping behavior for instance, we can target the groups separately with different marketing campaigns.
  - *Image segmentation*: We can use clustering to segment images into different regions, for instance useful in Medical Image Analysis.
  - *Gene analysis*: Group similar gene expressions to understand genes' functions and regulatory mechanisms
  - *Topic Modeling*: Group a collection of text document (like news articles) into different topics (non-predefined topics)
  - …

Clustering as an example of unsupervised learning

- **Basic Steps of Clustering**
    1. Feature selection
    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.

Clustering as an example of unsupervised learning

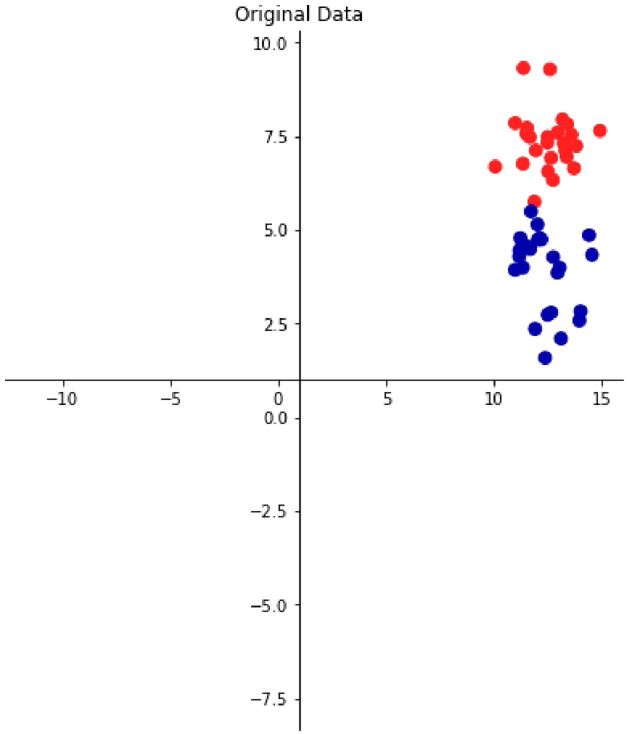
- **Data scaling before Clustering**

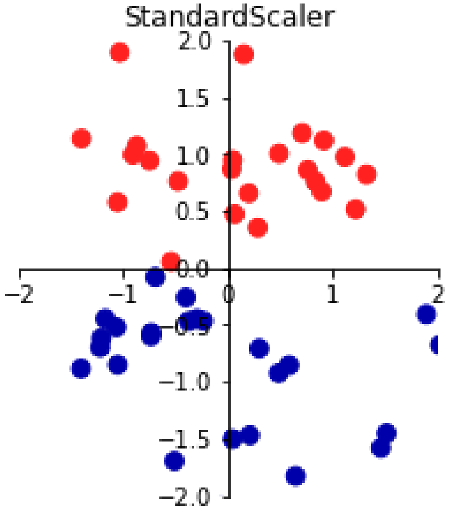
age	income
64	87083.24
33	76807.82
24	12043.60
33	61972.00
78	60120.32
62	40058.42

- 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

Clustering as an example of unsupervised learning

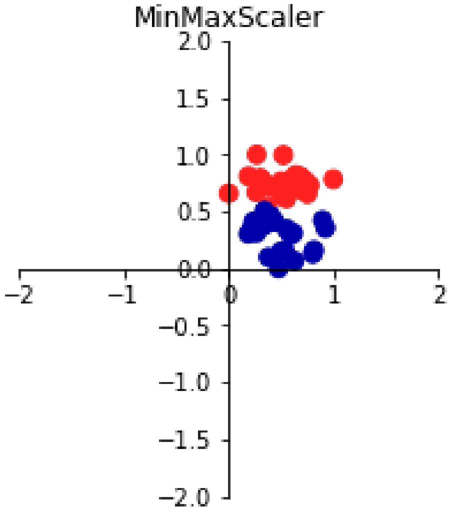
- **Preprocessing and Scaling**





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]

Roskilde Universitet

Outline of this lecture

- Clustering as an example of unsupervised learning
- K-means clustering
- Hierarchical clustering
- DBSCAN clustering
- Evaluation of clustering models

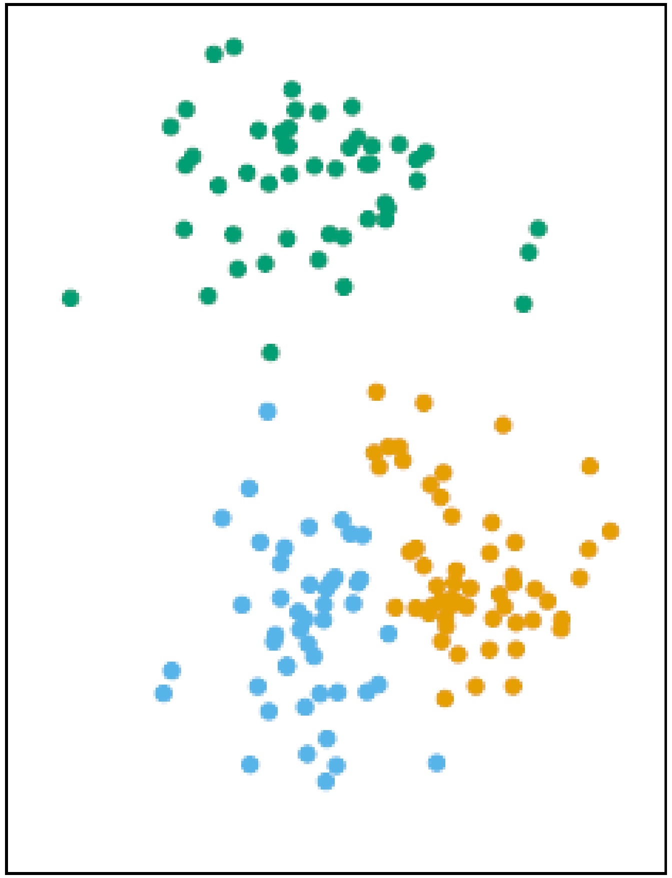
K-means clustering

- A clustering is a set of clusters (set of points) such that:
  - All data points belong to exactly one cluster
- **The K-means clustering problem**
  - Can we choose K clusters such that we minimize the *within-cluster variation*?

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

  - If we measure within-cluster variation with squared Euclidian distance, it reduces to:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$



Roskilde Universitet

K-means clustering

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

Initialization

1. Partition all points *randomly* into K nonempty subsets (clusters)

Iterations

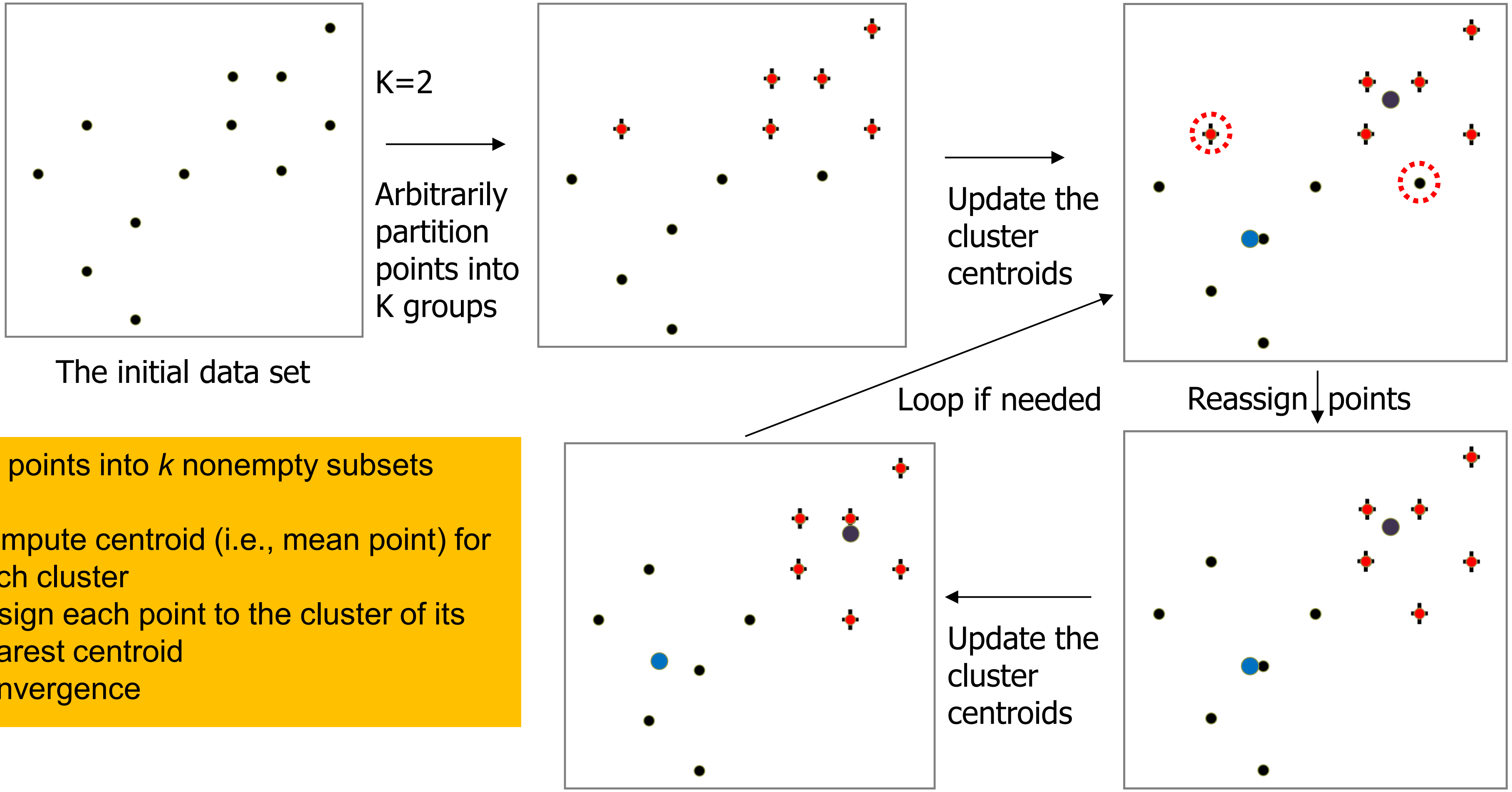
2. Compute the centroids of the clusters of the current partitioning
   - The centroid is the center, i.e., mean point, of a cluster
3. Assign each point to the cluster with the *nearest* centroid

Convergence

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

K-means clustering

An Example of K-Means Clustering



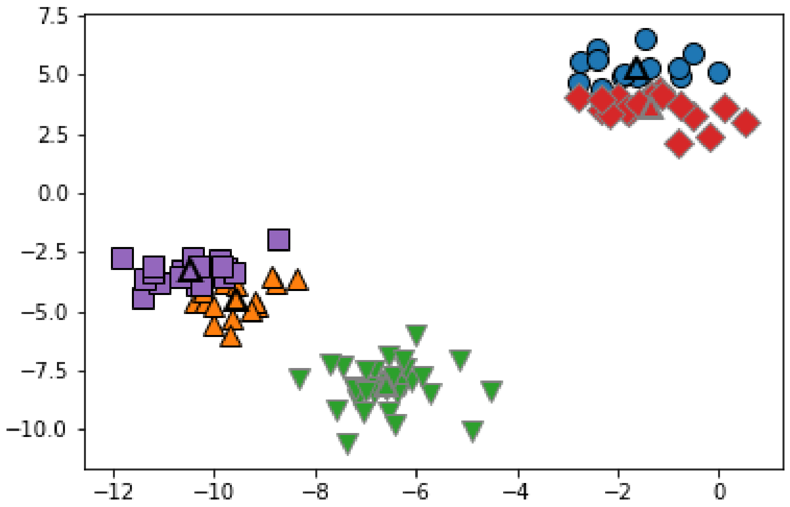
Partition points into *k* nonempty subsets
Repeat
- Compute centroid (i.e., mean point) for each cluster
- Assign each point to the cluster of its nearest centroid

Until convergence

Roskilde Universitet

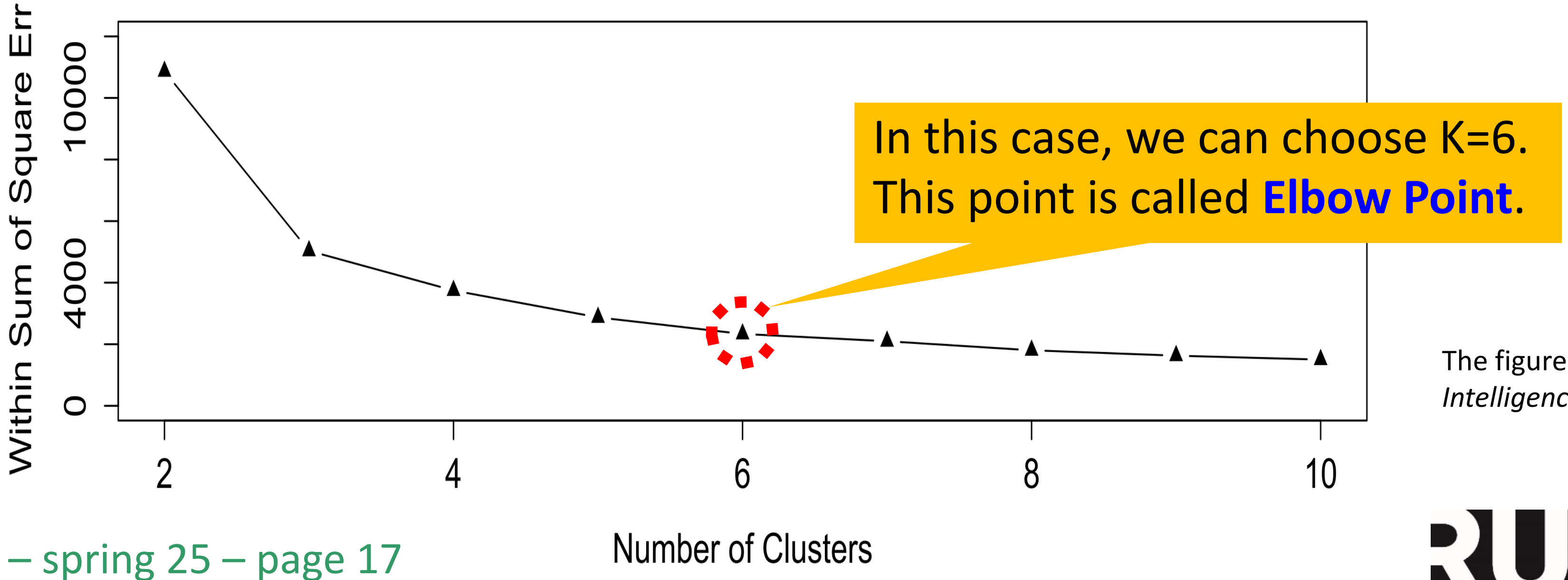
K-means clustering

- **Notes on K**
  - Different initializations, might affect the final clustering – K-means is only locally optimal
  - 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.
    - Only useful if the data is not high-dimensional



K-means clustering

- **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 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*.



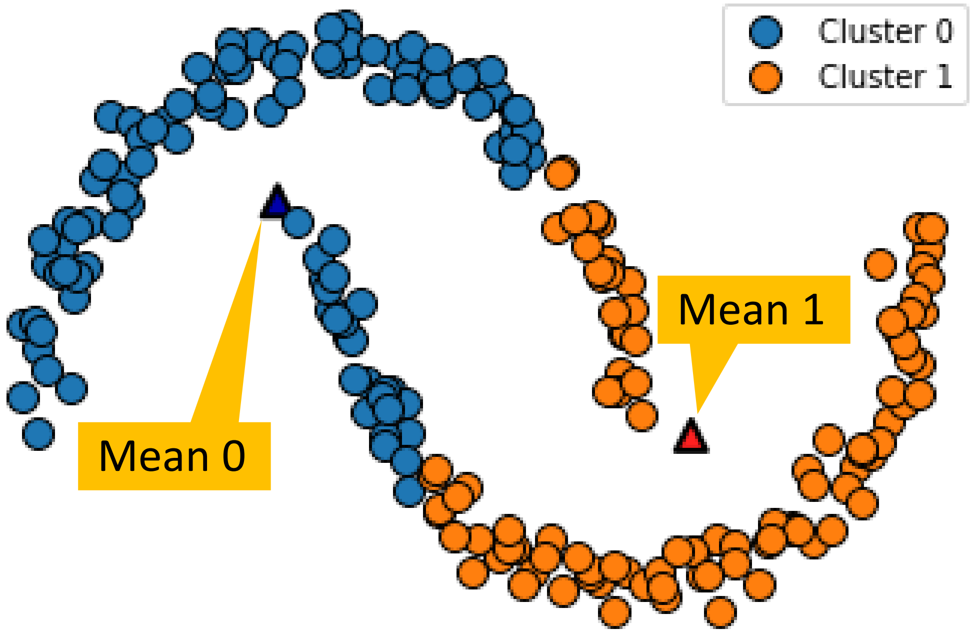
The figure is from *Introduction to R for Business Intelligence* by Jay Gendron

K-means clustering

- **Weaknesses of K-Means**
  - Applicable only to points in a *continuous* n-dimensional space
    - We cannot calculate means on categorical values.
  - Initialization matters.
    - Need to specify K, the number of clusters, in advance
      - In the literature, there are ways to automatically determine the best k
    - Different random initializations can create different final clusterings – K-means is only locally optimal
  - Convergence
    - Stop condition can be 'Relatively few points change the clusters'.
    - Often terminates at a *local* optimal.
  - Sensitive to noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes

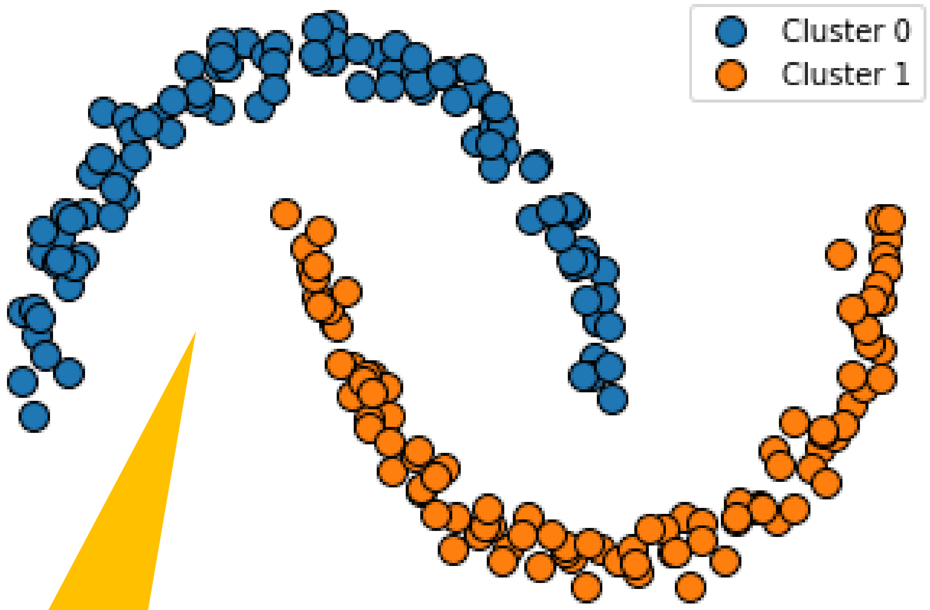
K-means clustering

K-means on an example of non-convex Shapes



K-means clustering result (K=2)

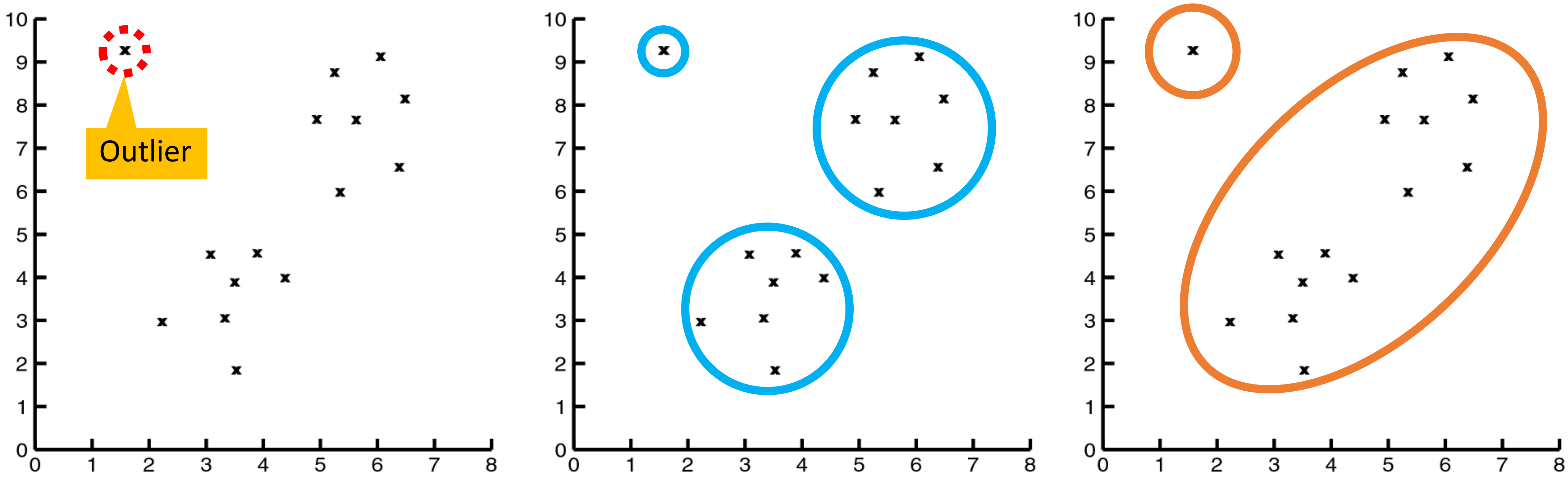
Density Based Spatial Clustering of Applications with Noise (**DBSCAN**)



Desired clustering result

K-means clustering

- **Impact of Outliers on k-Means**



Dataset with an outlier

K=3

K=2

K-means clustering

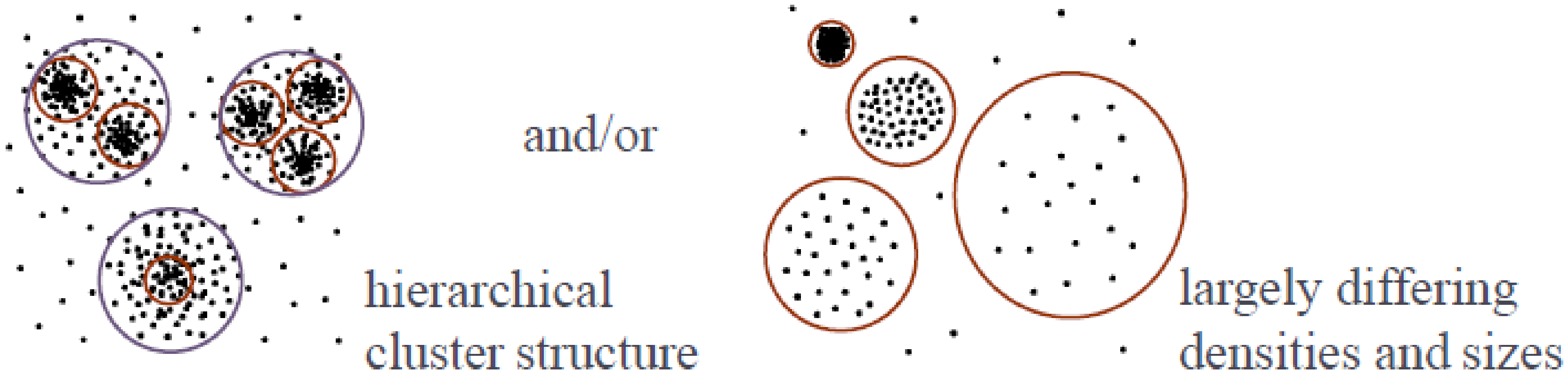
- Let us look at examples in the notebook “K-Means clustering.ipynb”.

Outline of this lecture

- Clustering as an example of unsupervised learning
- K-means clustering
- Hierarchical clustering
- DBSCAN clustering
- Evaluation of clustering models

Hierarchical clustering

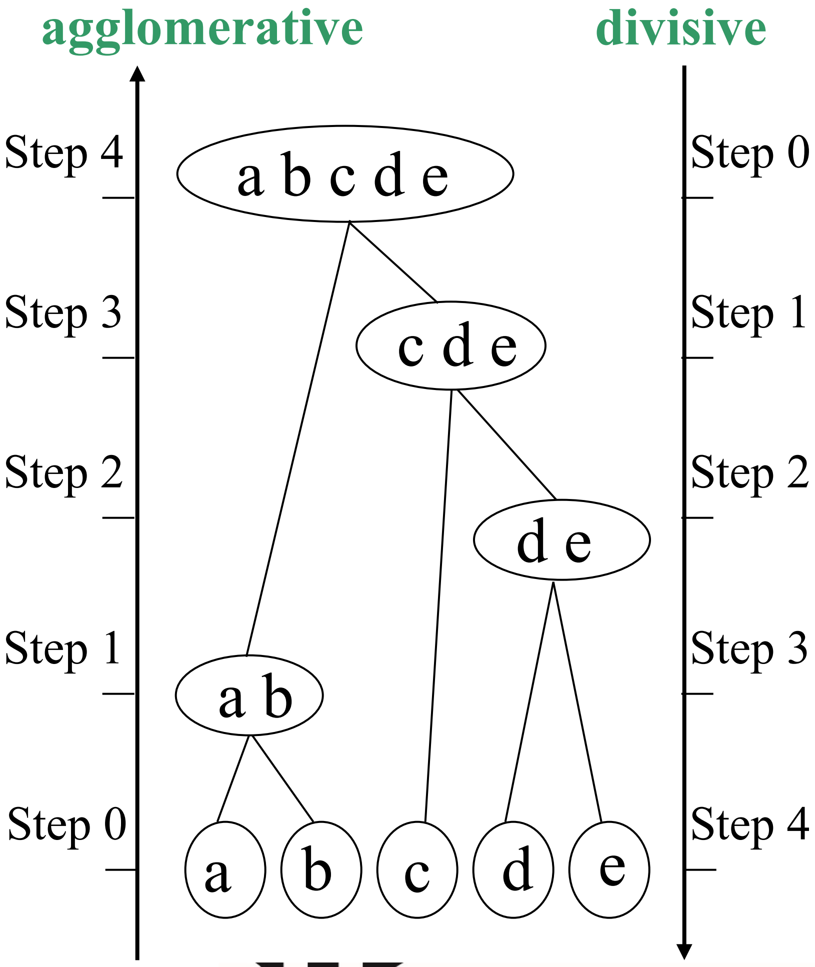
- **Why Hierarchical Clustering?**
  - No need to specify the number of clusters beforehand



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

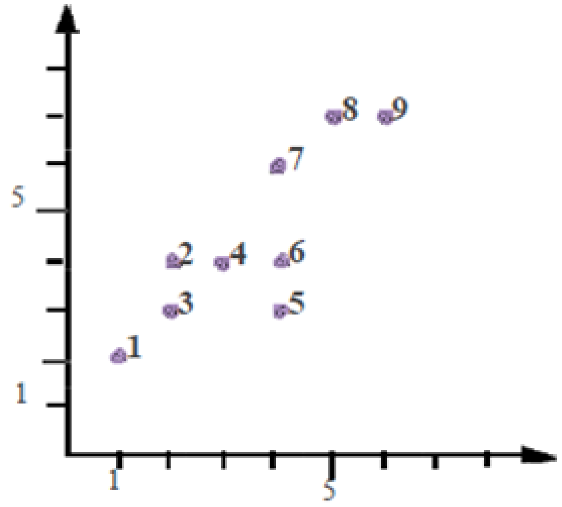
Hierarchical clustering

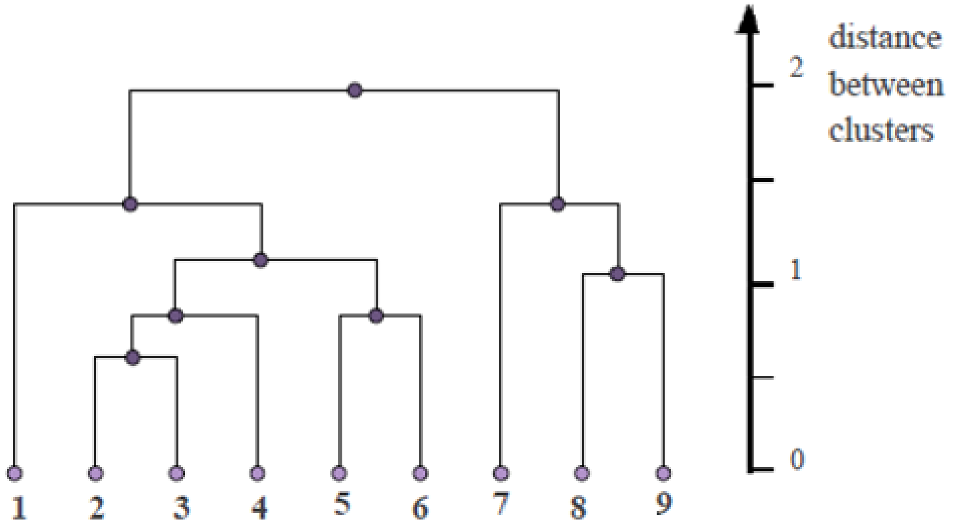
- **Hierarchical Clustering Approaches**
  - Hierarchical Clustering needs a *distance metrics,* but do not need a fixed number of desired clusters in advance.
  - Output: a hierarchy of potential clusterings
  - **Agglomerative** clustering algorithms
    - Initially each item in its own cluster
    - Iteratively clusters are merged together
    - Bottom Up
  - **Divisive** clustering algorithms
    - Initially all items in one cluster
    - Large clusters are successively divided
    - Top Down



Hierarchical clustering

- **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.

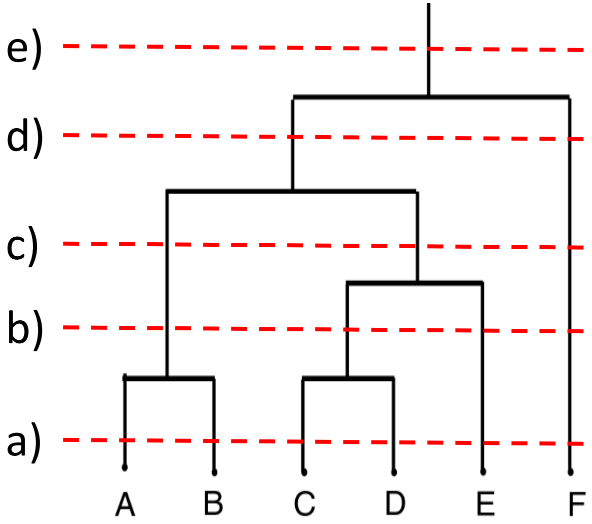


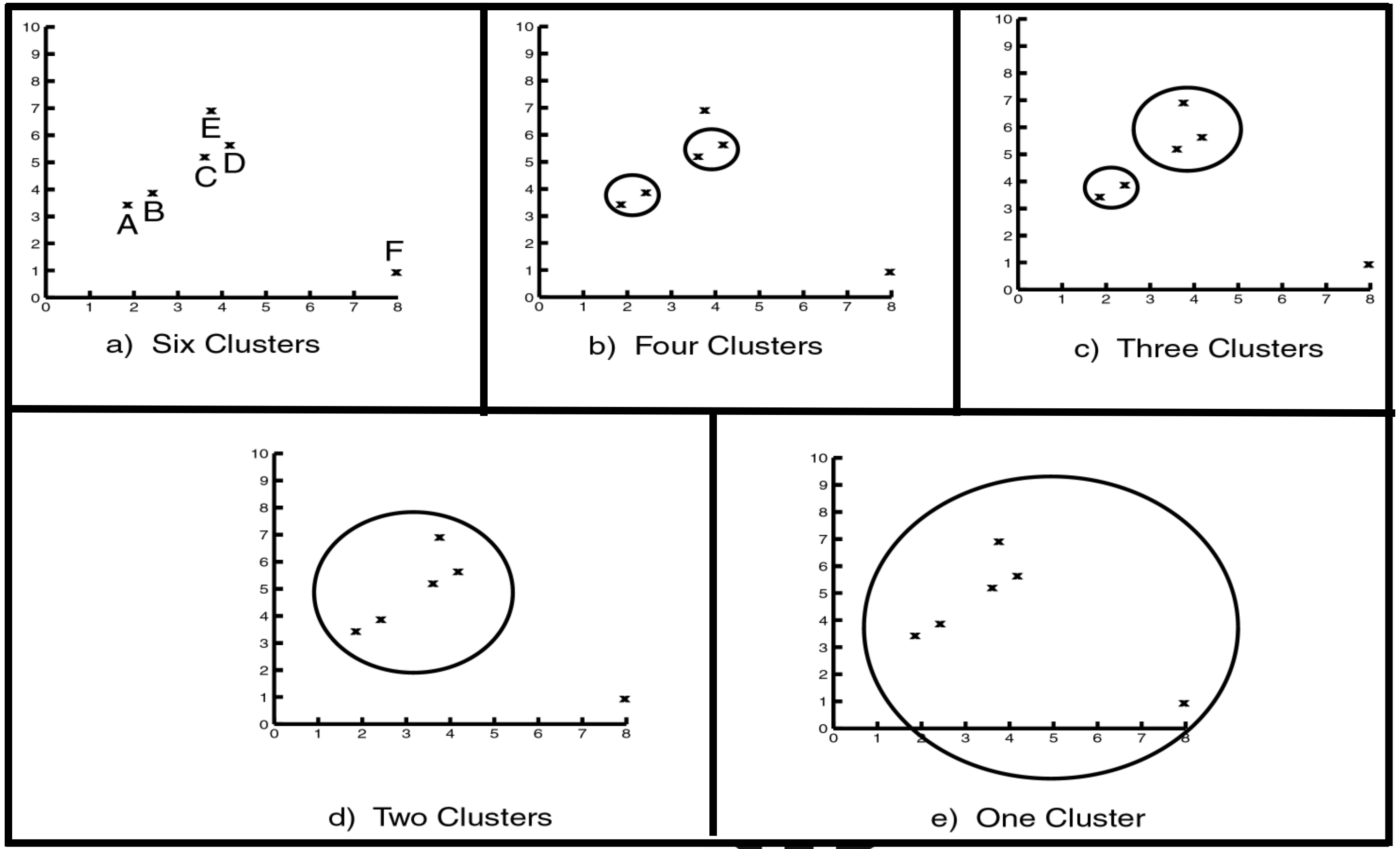


Levels of Clustering (Agglomerative)

Levels of Clustering (Agglomerative)

- Each horizontal line in the Dendrogram, correspond to a particular set of clusters
  - where the vertical lines are intersected, the subtrees corresponds to the cluster
- The higher we cut the fewer clusters





Hierarchical clustering

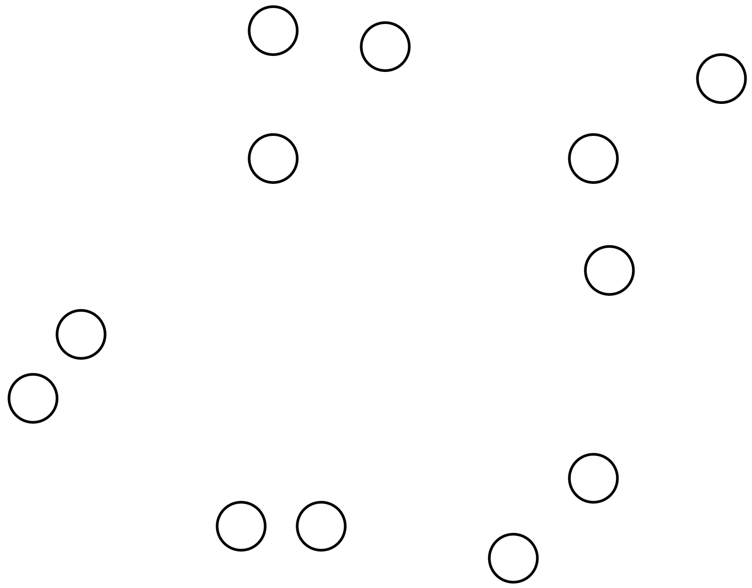
- **The Agglomerative Clustering Algorithm**
  - Most popular hierarchical clustering technique

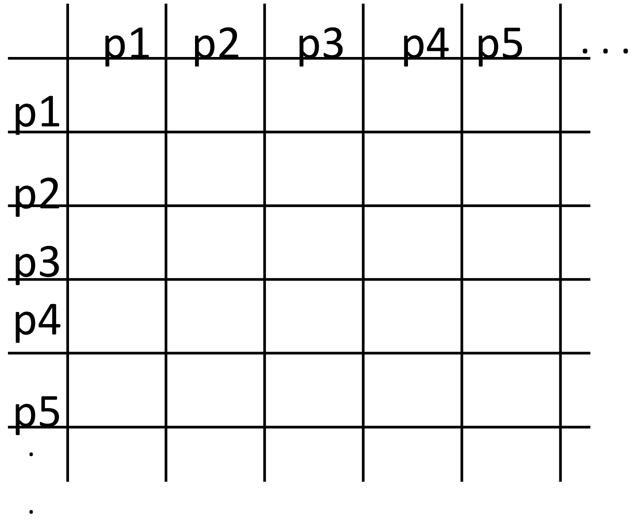
  - **Basic algorithm**:
    1. Compute an *adjacency matrix*
    2. Let each data point be a cluster
    3. **Repeat**
       1. *Merge* two clusters if the distance is small enough
       2. *Update* the adjacency matrix
    4. **Until** only a single cluster remains

  - Key operation: ***computing similarity of two clusters***
    - Different ways to define distance between clusters produce different clustering results.

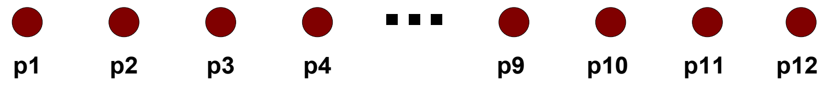
Hierarchical clustering

- **Agglomerative Clustering**
  - Start with clusters of individual points and an adjacency matrix



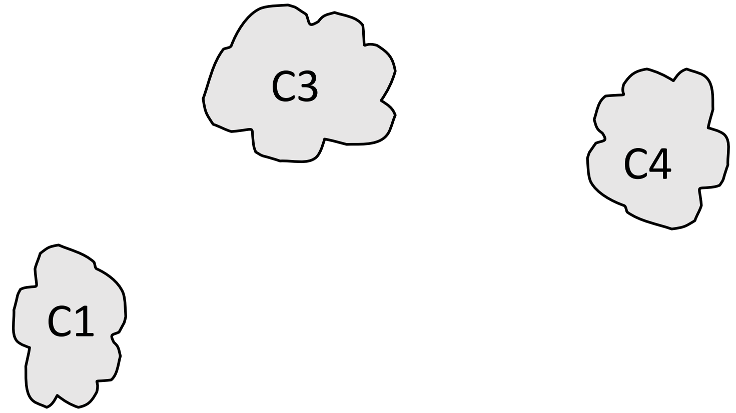


Adjacency Matrix



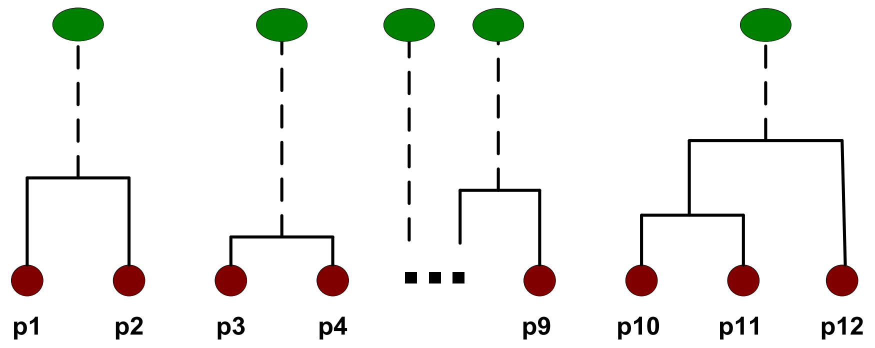
Hierarchical clustering

- After some merging steps, we have some clusters



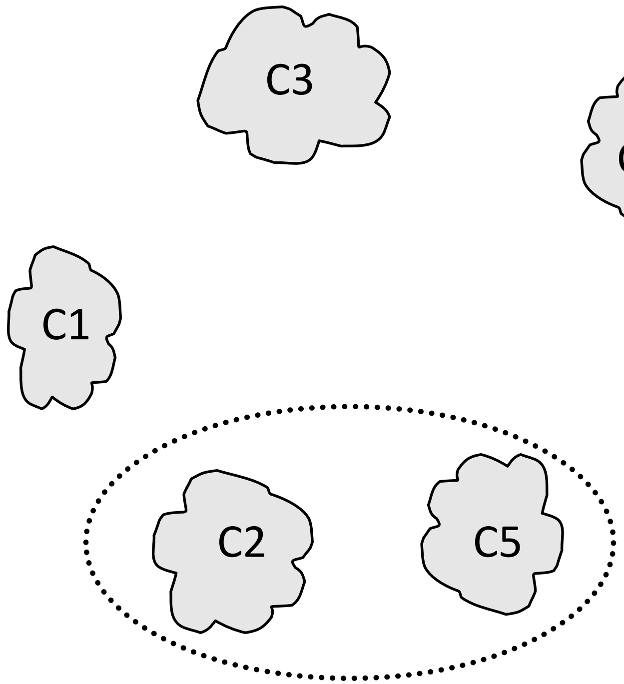
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

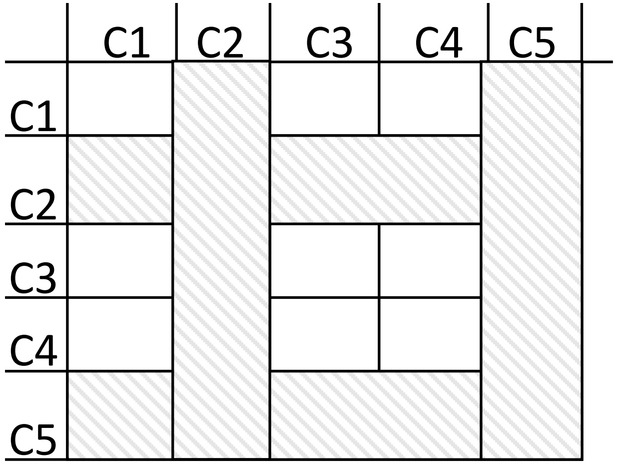
Adjacency Matrix



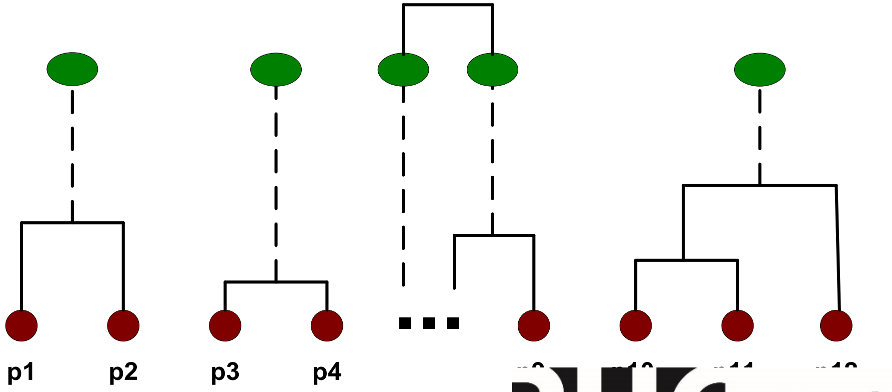
Hierarchical clustering

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



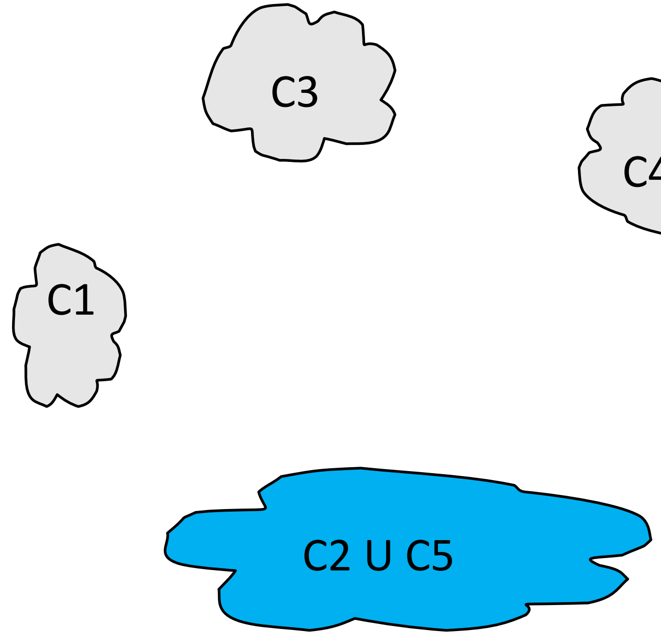


Adjacency Matrix



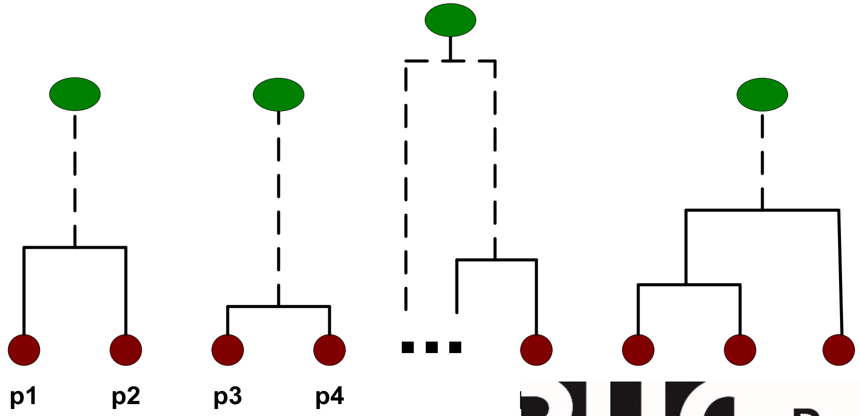
Hierarchical clustering

- How to update the adjacency matrix?

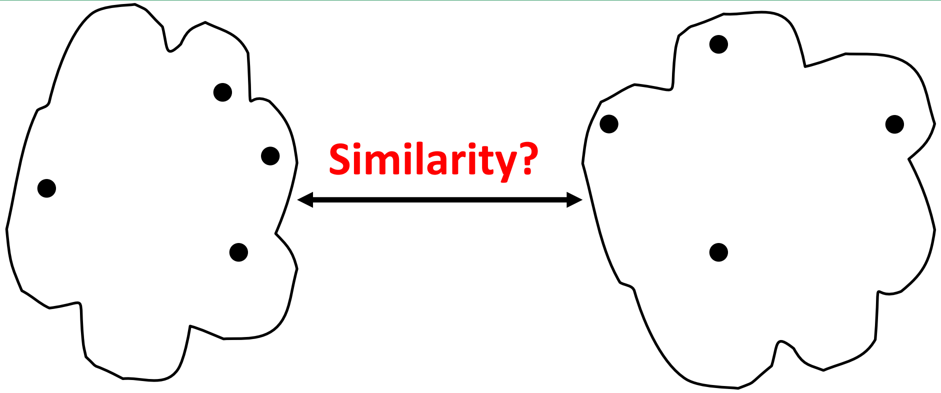


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



Hierarchical clustering

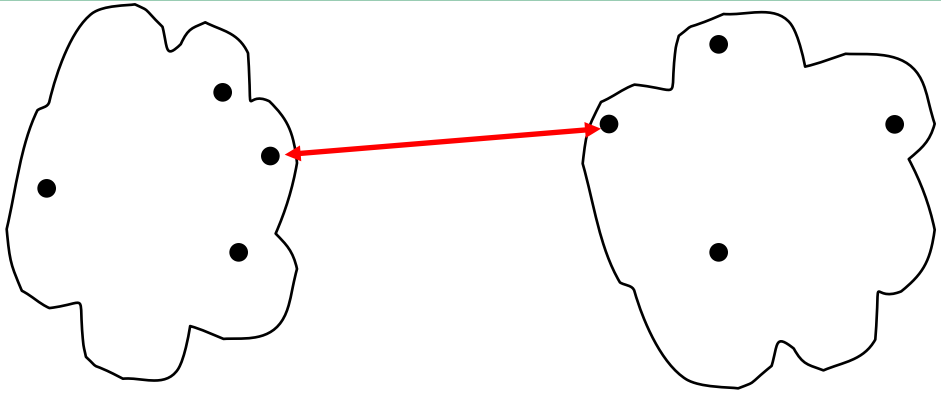


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Adjacency Matrix

- **How to Define Inter-Cluster Similarity? (/linkage)?**
  - Single (MIN)
  - Complete (MAX)
  - Average
  - Centroid

Hierarchical clustering

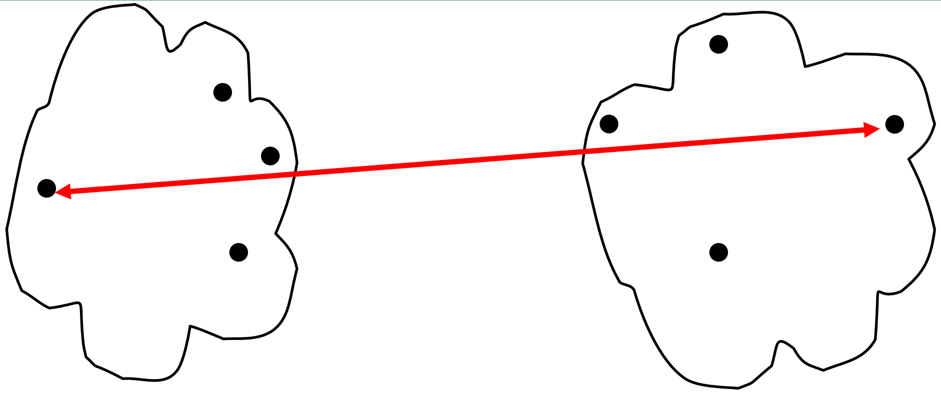


- **How to Define Inter-Cluster Similarity (/linkage)?**
  - Single (MIN)
  - Complete (MAX)
  - Average
  - Centroid

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

Adjacency Matrix

Hierarchical clustering



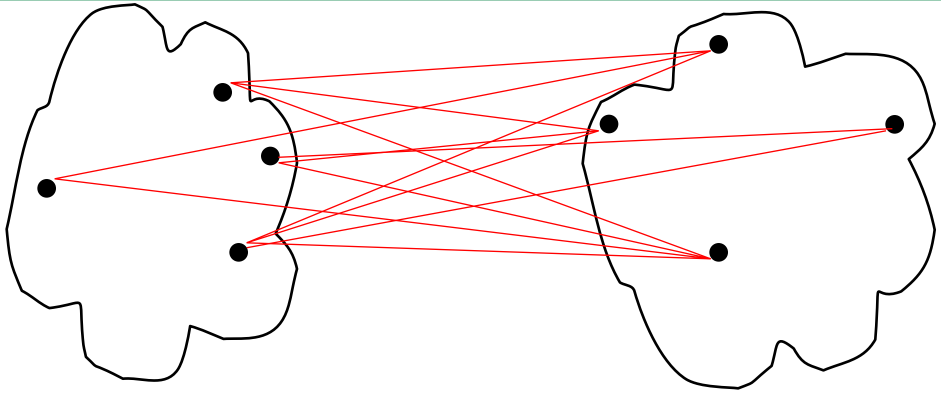
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Adjacency Matrix

- **How to Define Inter-Cluster Similarity? (/linkage)?**
  - Single (MIN)
  - Complete (MAX)
  - Average
  - Centroid

Roskilde Universitet

Hierarchical clustering

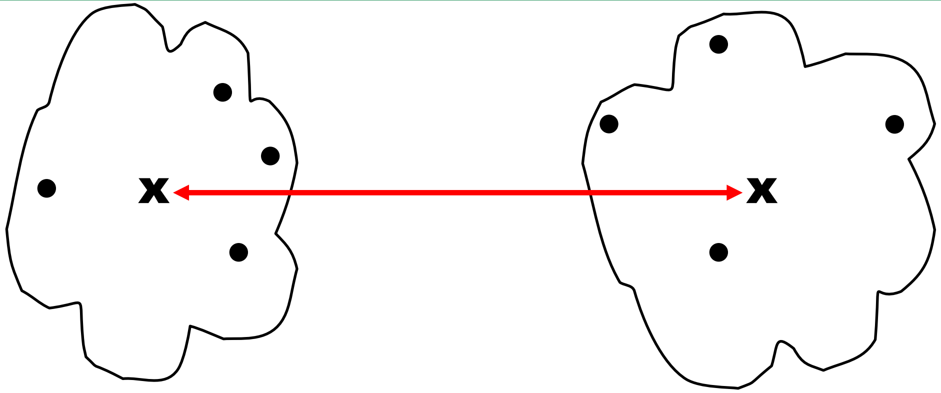


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Adjacency Matrix

- **How to Define Inter-Cluster Similarity? (/linkage)?**
  - Single (MIN)
  - Complete (MAX)
  - Average
  - Centroid

Hierarchical clustering



- **How to Define Inter-Cluster Similarity? (/linkage)?**
  - Single (MIN)
  - Complete (MAX)
  - Average
  - Centroid

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

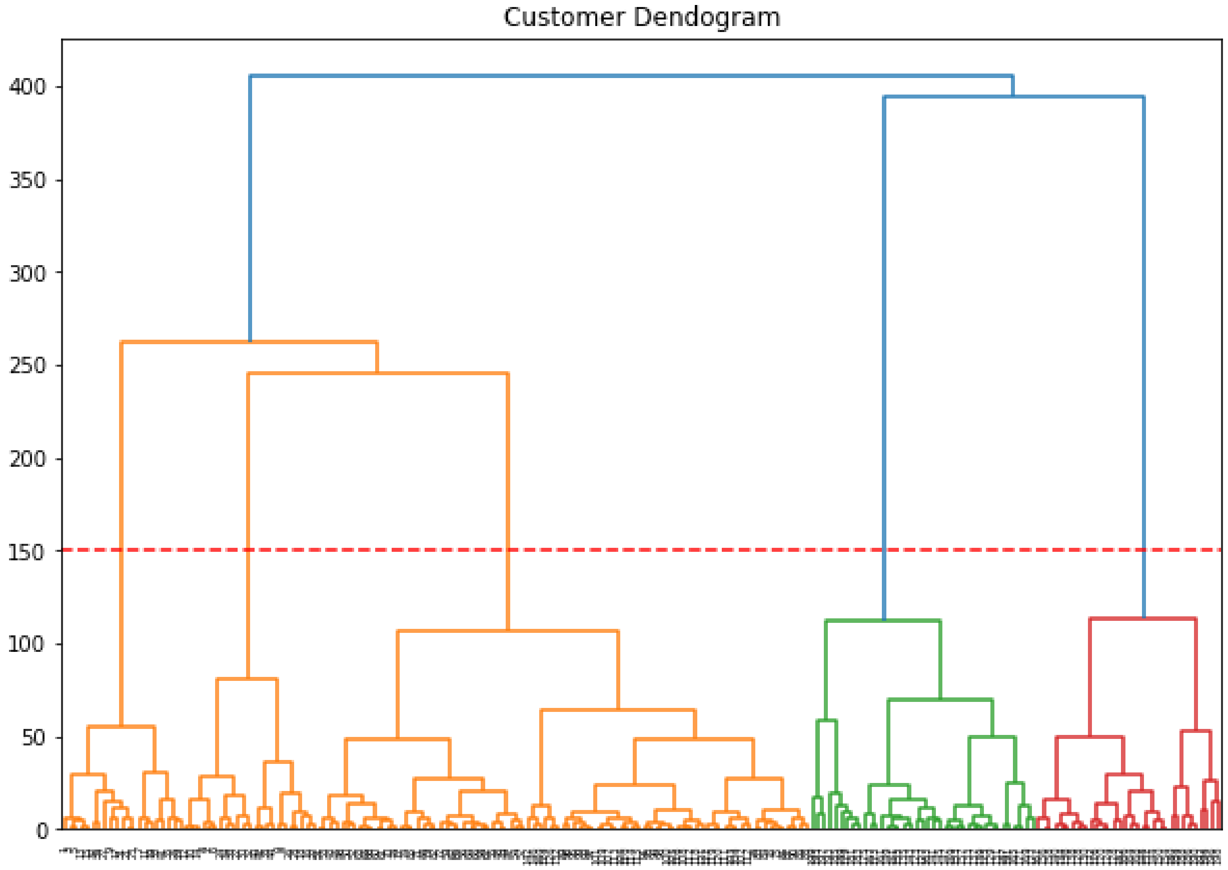
Adjacency Matrix

Hierarchical clustering

- **Another Inter-Cluster Similarity: Ward's Method**
  - Similarity of two clusters measured as **increase** in sum of squared error (SSE) when they are merged
  - Less susceptible to noise and outliers
  - Biased towards globular clusters
  - Hierarchical "analogue" of K-means
  - Default in Scikit-Learns's *AgglomerativeClustering* method:
    - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

Hierarchical clustering

- **Deciding the number of Clusters from a 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.



Hierarchical clustering

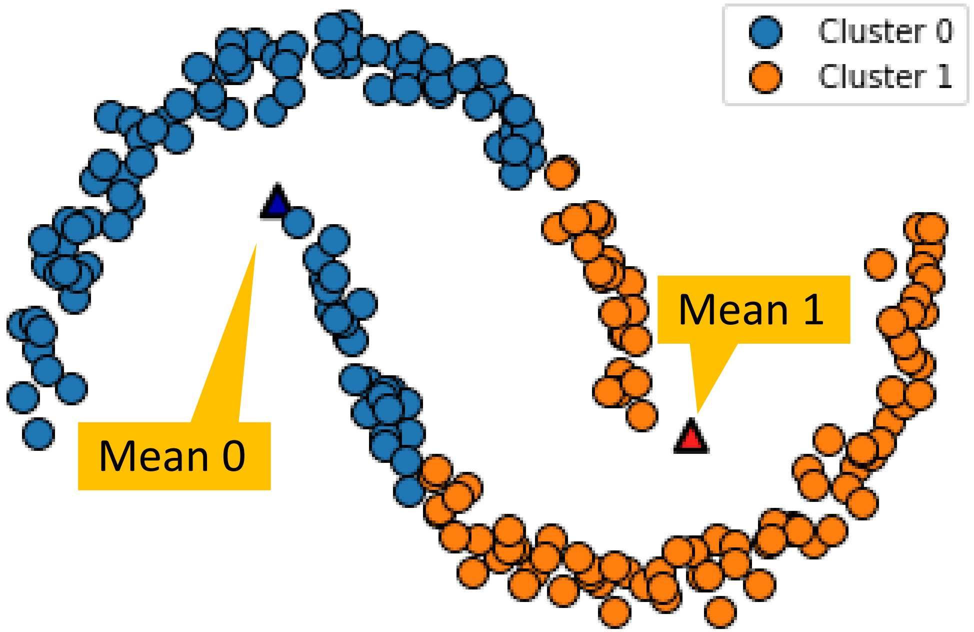
- Let us look at examples in the notebook “Hierarchical clustering.ipynb”.

Outline of this lecture

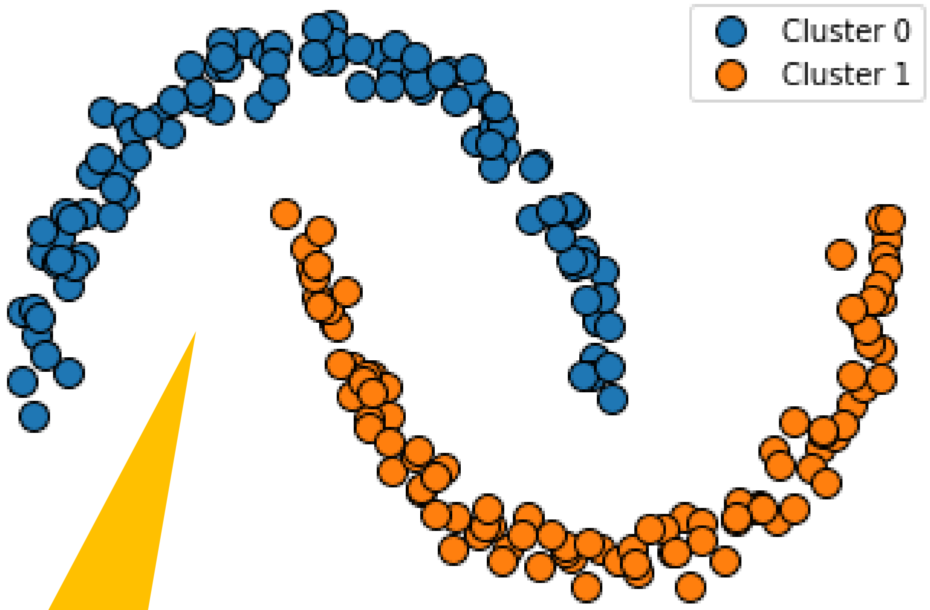
- Clustering as an example of unsupervised learning
- K-means clustering
- Hierarchical clustering
- DBSCAN clustering
- Evaluation of clustering models

DBSCAN clustering

Recall this example – how do we cluster the "right" way here



K-means clustering result (K=2)



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

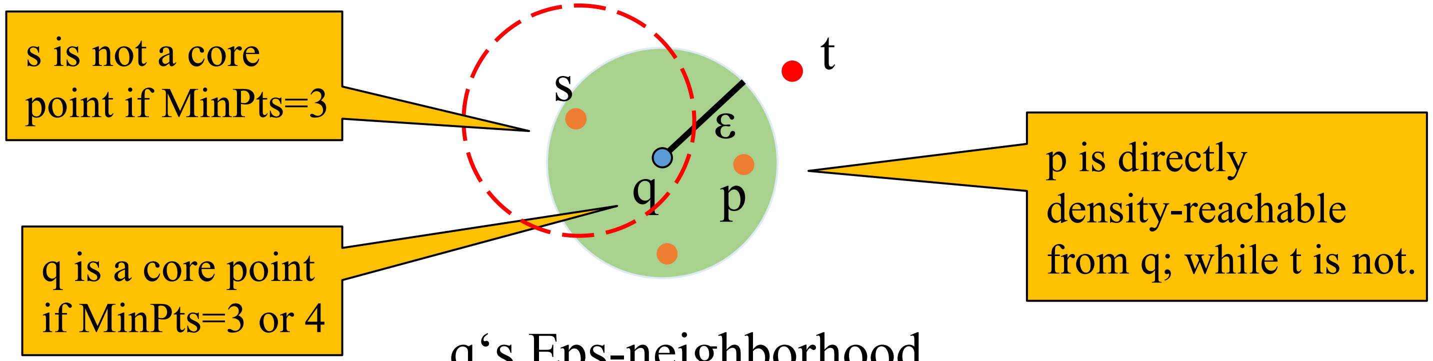
Desired clustering result

DBSCAN clustering

- **DBSCAN: D**ensity **B**ased **S**patial **C**lustering of **A**pplications with **N**oise
  - The algorithm's parameters (hyper-parameters):
    - **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 at least another one point in it less than this distance away.
      - Distance between points
      - **eps** in sklearn.cluster.DBSCAN

DBSCAN clustering

- **Eps-neighborhood**
  - Given a point, its Eps-neighborhood is all points within Eps distance of the given point.
- **Core point**
  - Points 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.



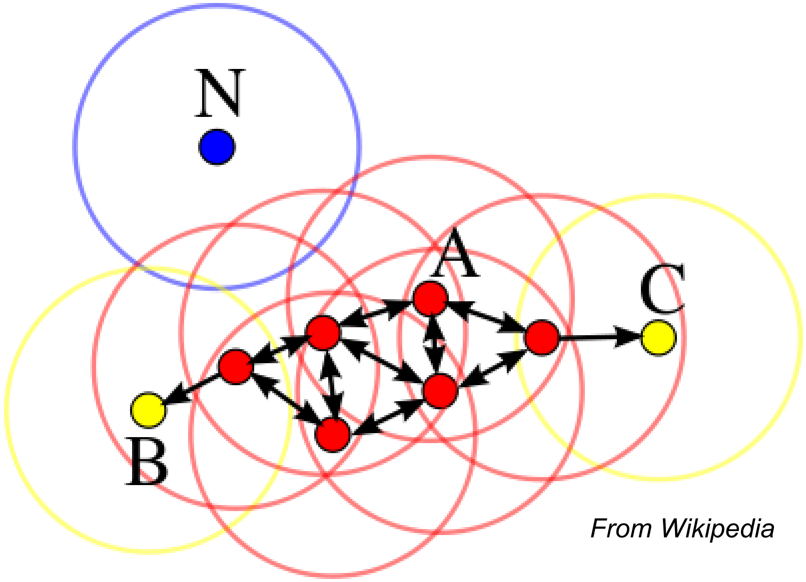
q's Eps-neighborhood

Roskilde Universitet

DBSCAN clustering

- **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.

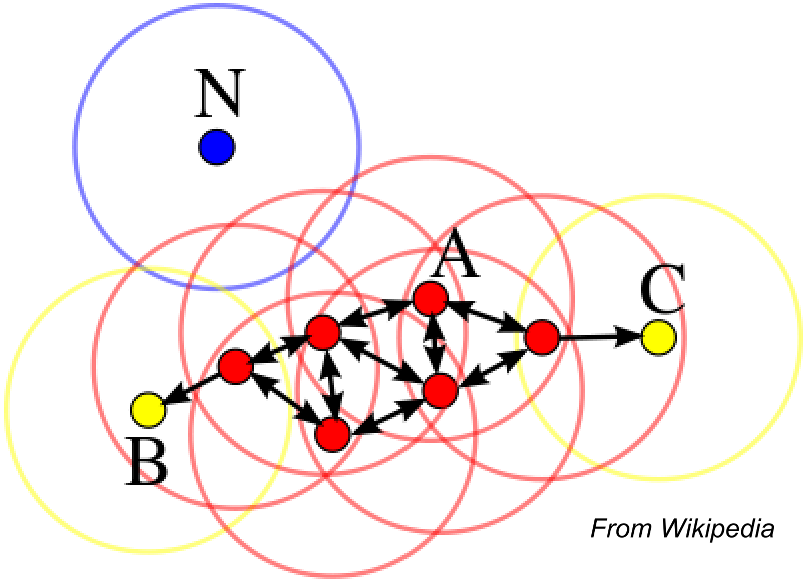
Assume MinPts=3.
- Red points are core points.
- Points B and C are *density-reachable* from A.
- Point B is not density-reachable from C; and vice versa.



From Wikipedia

DBSCAN clustering

- **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 that are not in any cluster
  - Outliers will note effect creation of clusters



From Wikipedia

DBSCAN clustering

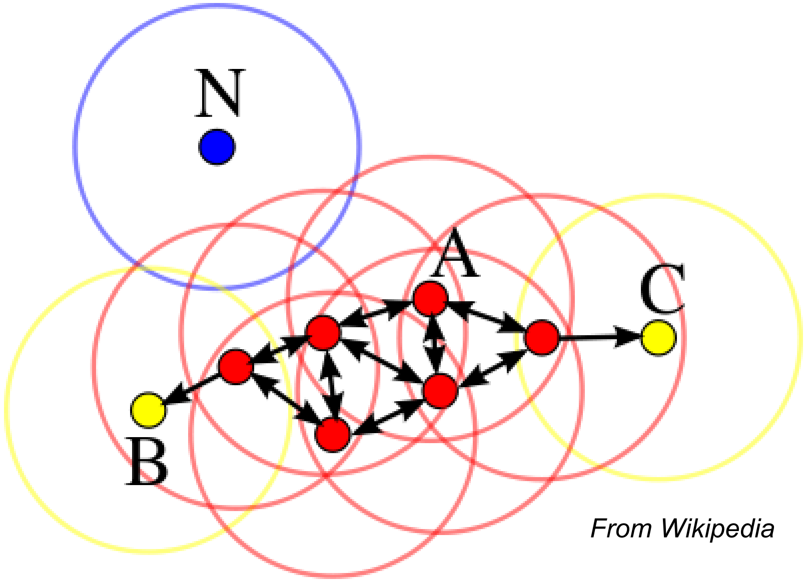
- The DBSCAN algorithm

```
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 clustering

- 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

DBSCAN clustering

- Let us look at examples in the notebook “DBSCAN clustering.ipynb”.

Outline of this lecture

- Clustering as an example of unsupervised learning
- K-means clustering
- Hierarchical clustering
- DBSCAN clustering
- Evaluation of clustering models

Evaluation of clustering models

- Quality: **What is a 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., hyper-parameters), and
    - its ability to discover *some* or *all* of the hidden patterns

Evaluation of clustering models

- **Rand Index (William M. Rand 1971) – measure difference between clusterings**
  - A set of points $S = \{o_1, \ldots, o_n\}$. Two clusterings: $X = \{X_1, \ldots, X_r\}$ 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

Evaluation of clustering models

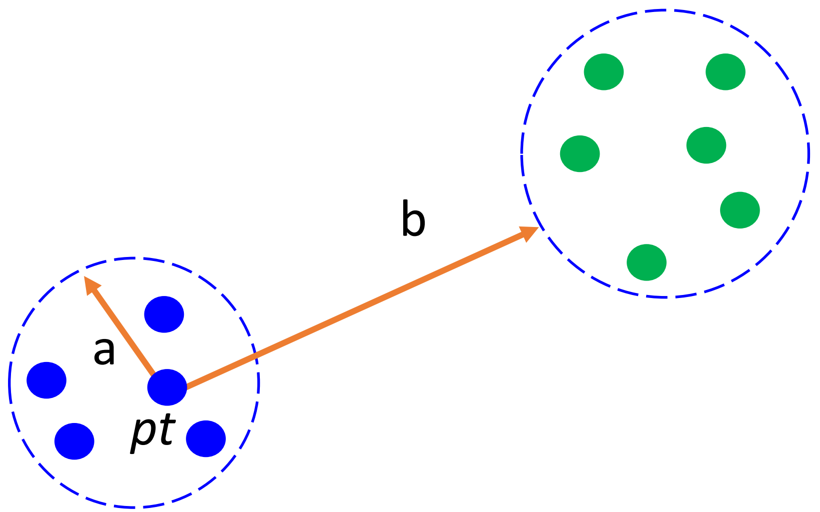
- **Adjusted Rand Index – measure difference between clusterings**
  - A set of points $S = \{o_1, ..., o_n\}$. Two clusterings: $X = \{X_1, ..., X_r\}$ and $Y = \{Y_1, ..., Y_r\}$
  - **The contingency table**: $n_{ij} = |X_i \cap Y_j|$
    - Each entry denotes the number of objects in common between X_i and 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: $ARI = \dfrac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}}$

Evaluation of clustering models

- **Silhouette Score – measure the quality of a clusterings**
  - 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*) = ($\bar{b}$ - $\bar{a}$) / max($\bar{a}$, $\bar{b}$)
      - $\bar{a}$, $\bar{b}$: 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.



Roskilde Universitet

Evaluation of clustering models

- Evaluation of Clustering in Scikit-Learn
  - If you have clustering groundtruth
    - Compare the clustering result with the groundtruth by measuring a score
      - Adjusted Rand Index (**ARI**): adjusted_rand_score(groundtruth, clustering_result)
  - Otherwise
    - silhouette_score(X, clustering_results) computes the compactness of a cluster
  - All scores are in sklearn.metrics.cluster

Evaluation of clustering models

- Let us look at examples in the notebook “Evaluation of clustering models.ipynb”.

Exercises

- Do the exercise in the notebook “Exercises in Clustering.ipynb”