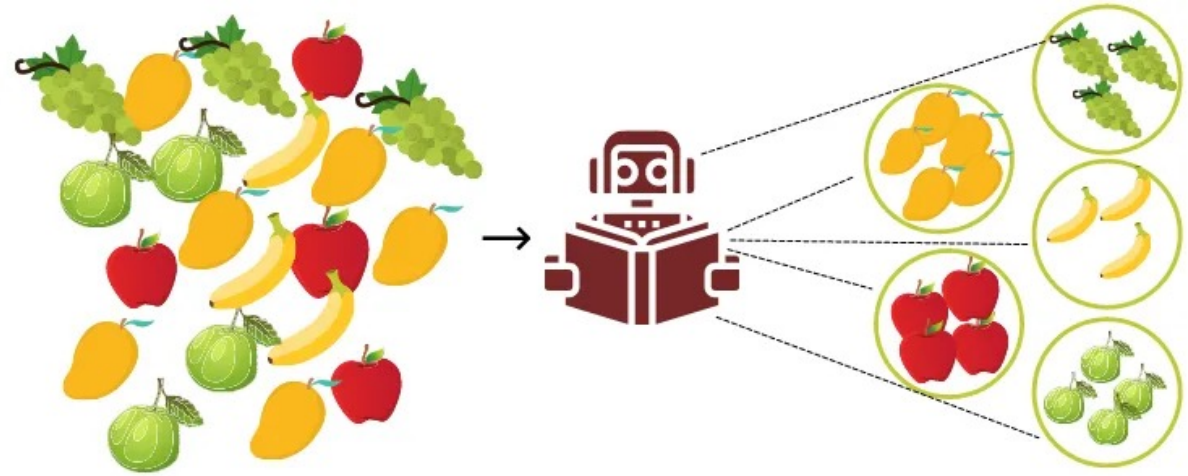
Lecture 4

Unsupervised learning
Clustering |

Golnaz Taheri, PhD
Senior Lecturer, Stockholm University

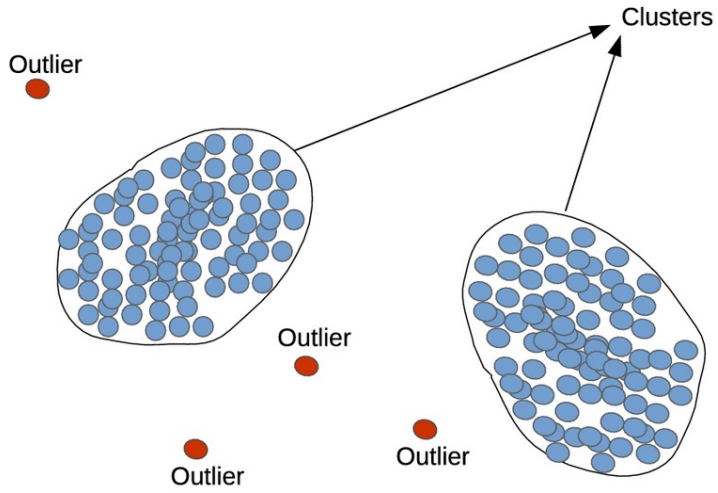
What is clustering?

- The organization of unlabeled data into similarity groups called clusters.
- A cluster is a collection of data items which are “similar” between them, and “dissimilar” to data items in other clusters.



What is Outliers

- Outliers are objects that do not belong to any cluster or form clusters of very small cardinality

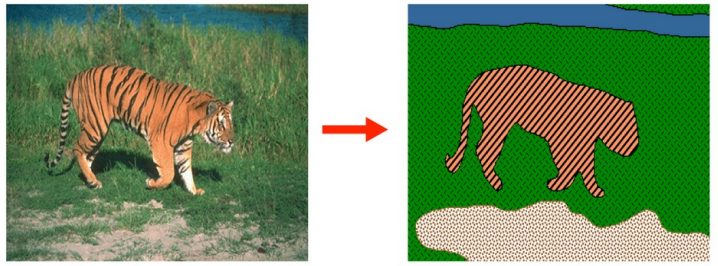


- In some applications we are interested in discovering outliers, not clusters (outlier analysis)

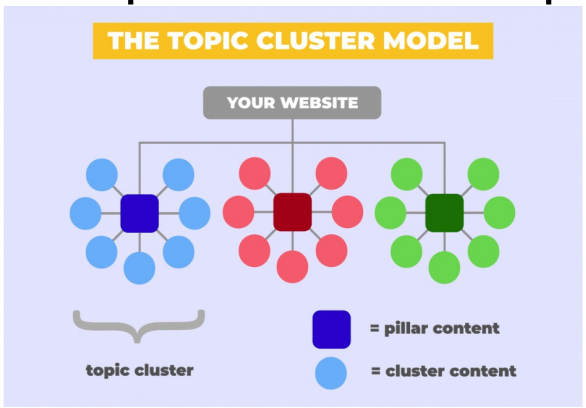
Stockholms universitet

Applications of clustering?

- Image Processing
  - cluster images based on their visual content



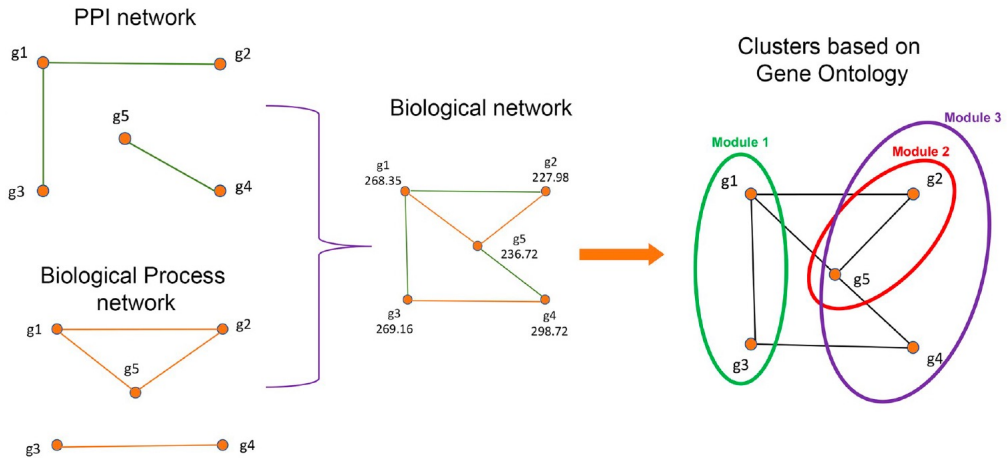
- Web
  - Cluster webpages based on their content
  - Cluster groups of users based on their access patterns on webpages

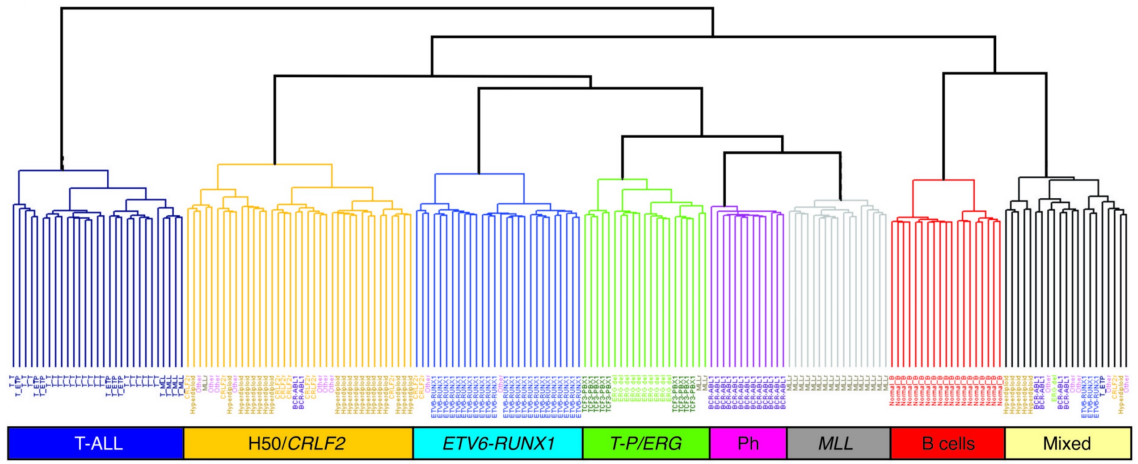


Stockholms universitet

Applications of clustering?

- Computational Biology
  - Gene-set clustering (modules) analysis
  - Gene expression profile for diseases

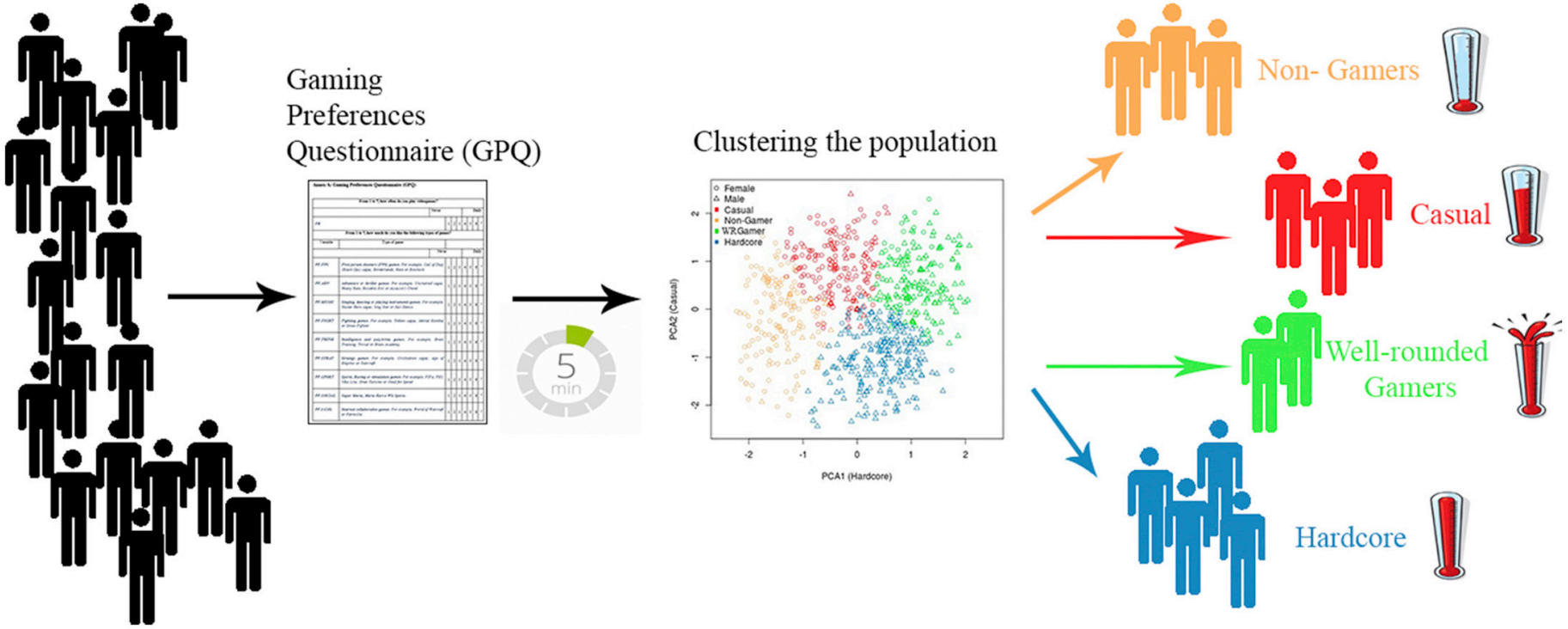




Stockholms universitet

Applications of clustering?

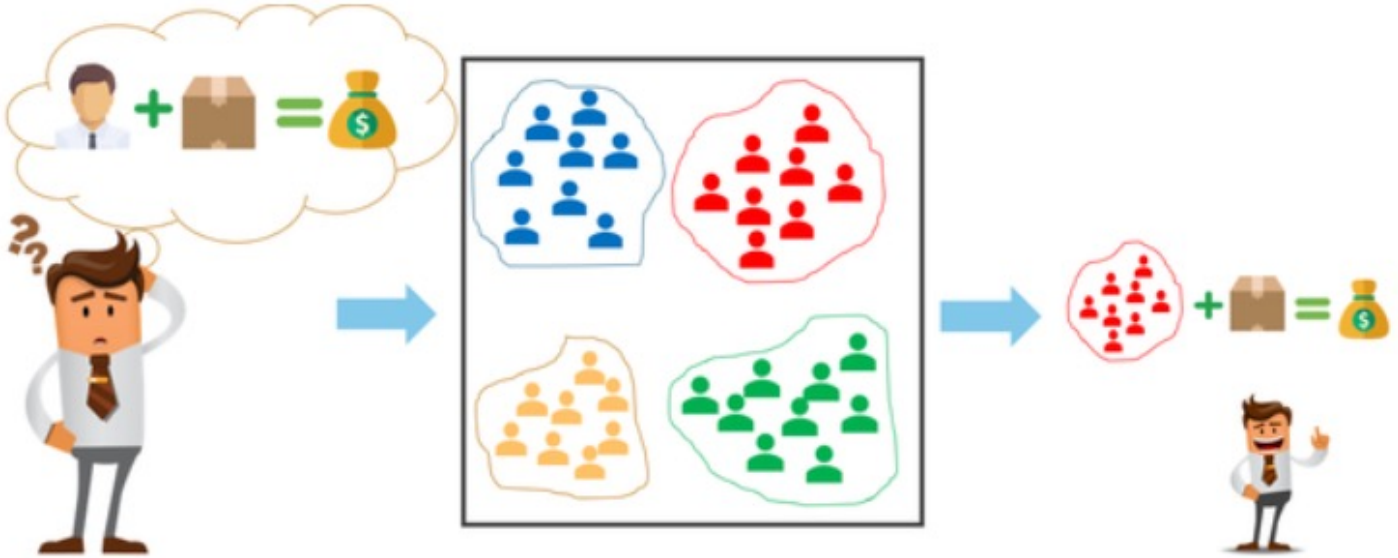
- Interaction design
  - Player segmentation



Stockholms universitet

Applications of clustering?

- Interaction design
  - Customer segmentation for product



Identifying the potential customer for selling products

Using clustering on the customer base

Selling products to the identified customer

The clustering task

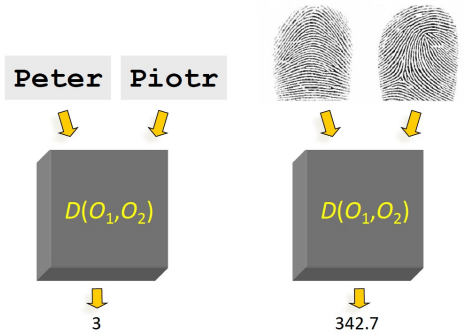
- Group observations into groups so that the observations belonging in the same group are similar, whereas observations in different groups are different

- Basic questions:
  - What does “similar” mean?
  - How to find a good partition of the observations?
  - What is a good partition of the objects? I.e., how is the quality of a solution measured?

Defining Distance Measures

- A distance measure is an objective score that summarizes the relative difference between two objects in a problem domain.

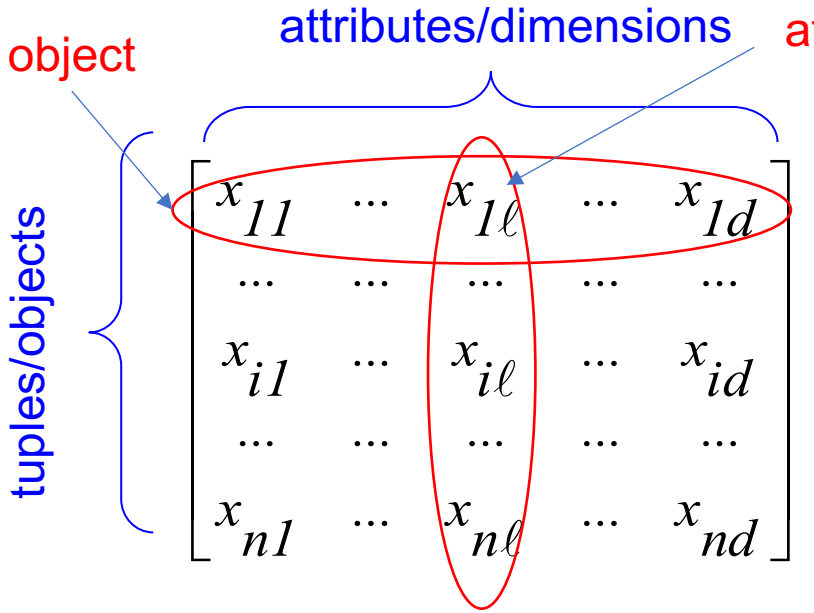
- Definition: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) between O_1 and O_2 is a real number denoted by $D(O_1, O_2)$



Stockholms universitet

Data Structures

- *Data* matrix
  - n objects
  - d attributes/dimensions per object



- *Distance* matrix

objects

objects

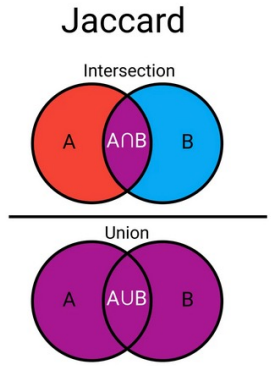
$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \ldots & \ldots & 0 \end{bmatrix}$$

10

Stockholms universitet

Distance (dissimilarity) measures

- **Jaccard similarity**
- (number of observations in both sets) / (number in either set)



$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

- If two datasets share the exact same members, their Jaccard Similarity Index will be 1. Conversely, if they have no members in common then their similarity will be 0.

- Jaccard distance between binary vectors A and B

$$Jdist(A,B) = 1 - JSim(A,B)$$

Stockholms universitet

The Jaccard similarity: Example

- **Jaccard similarity**
- similarity of a store's customers.
  - We have a binary attribute that corresponds to an *item purchased* at the store, where 1 indicates that a specific item was purchased and 0 indicates that a product was not purchased.

	item1	item2	item2	item4	item5	item6	item7	item8	item9
C1	0	1	0	0	0	1	0	0	1
C2	0	0	1	0	0	0	0	0	1
C3	1	1	0	0	0	1	0	0	0

$$J(C1, C2) = \frac{1}{1+1+2} = 0.25$$

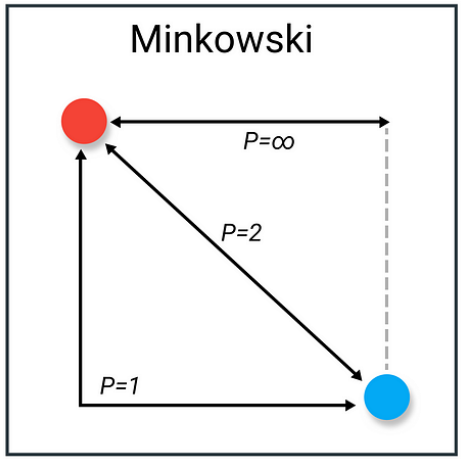
$$J(C1, C3) = \frac{2}{2+1+1} = 0.5$$

Distance (dissimilarity) measures

- **Minkowski Distance**
- The L_p-norm
- It is the generalized form of Euclidean and Manhattan Distance.

$$L_p(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$

where p is a positive integer

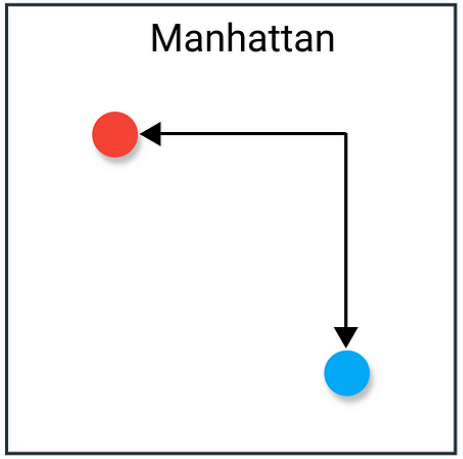


Stockholms universitet

Distance (dissimilarity) measures

- **Manhattan Distance**
- It is the sum of absolute differences between points across all the dimensions.
- Manhattan Distance is also known as city block distance
- It corresponds to the L_1 -norm of a difference between vectors and vector spaces.

$$L_1(x,y) = \sum_{i=1}^{d} |x_i - y_i|$$



Stockholms universitet

Manhattan distance: Example

- **Manhattan Distance**

$$L_1(x,y) = \sum_{i=1}^{d} |x_i - y_i|$$

$$X = [2, 2, 3, -1]$$
$$Y = [1, 4, 4, 0]$$

$$L_1(x,y) = |2-1| + |2-4| + |3-4| + |-1-0|$$
$$= |1| + |-2| + |-1| + |-1|$$
$$= 1 + 2 + 1 + 1 = 5$$

Euclidean distance: Example

- **Euclidean Distance**
- represents the shortest distance between two vectors.
- It is the square root of the sum of squares of differences between corresponding elements.
- It is the shortest path between source and destination which is a straight line
- It corresponds to the L_2 -norm of a difference between vectors and vector spaces.

$$L_2(x,y) = \sqrt{(|x_1 - y_1|^2 + |x_2 - y_2|^2 + \ldots + |x_d - y_d|^2)}$$

Distance (dissimilarity) measures

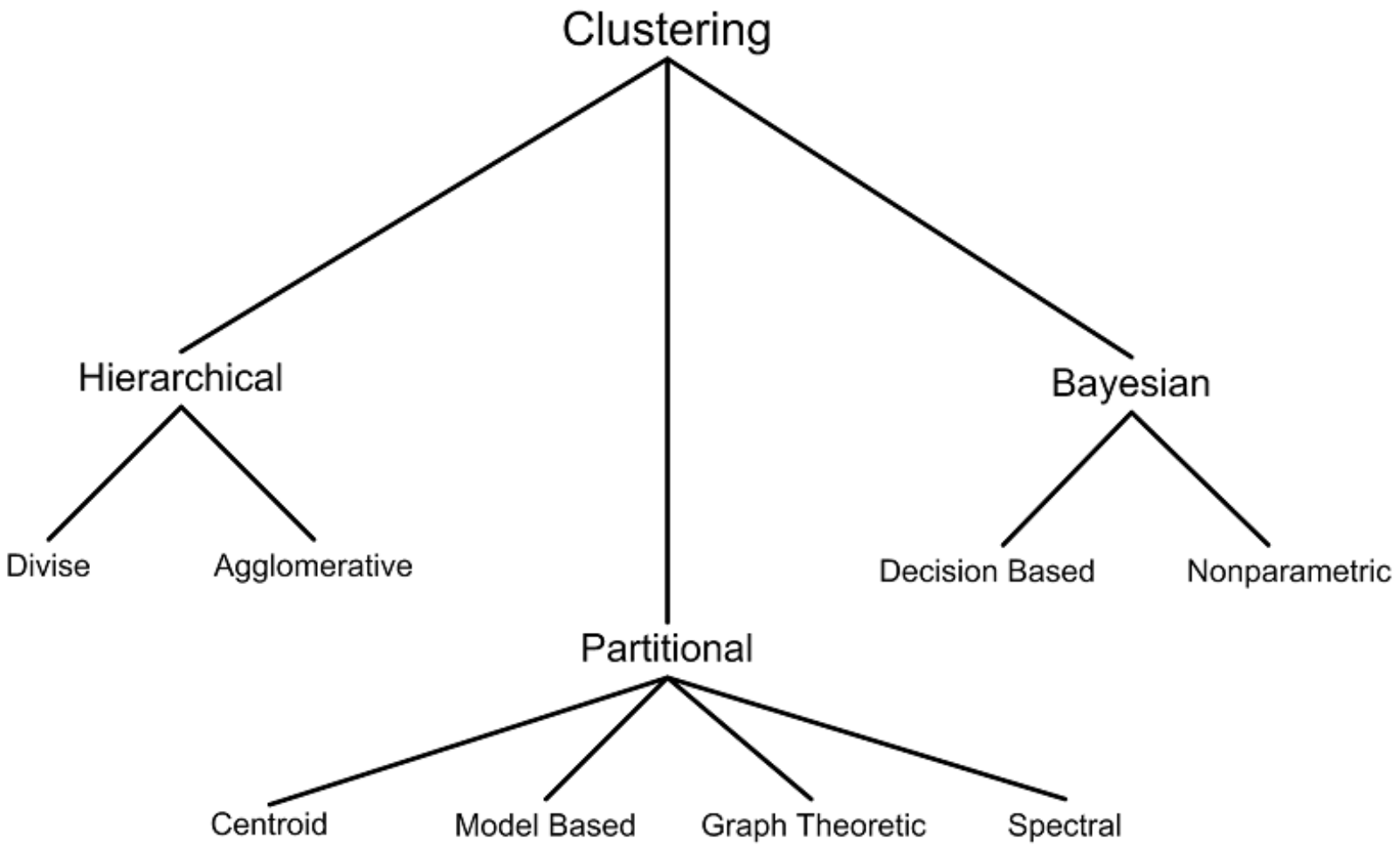
- **Euclidean Distance**

$$L_2(x,y) = \sqrt{(|x_1 - y_1|^2 + |x_2 - y_2|^2 + \ldots + |x_d - y_d|^2)}$$

$$X = [2, 2, 3, -1]$$
$$Y = [1, 4, 4, 0]$$

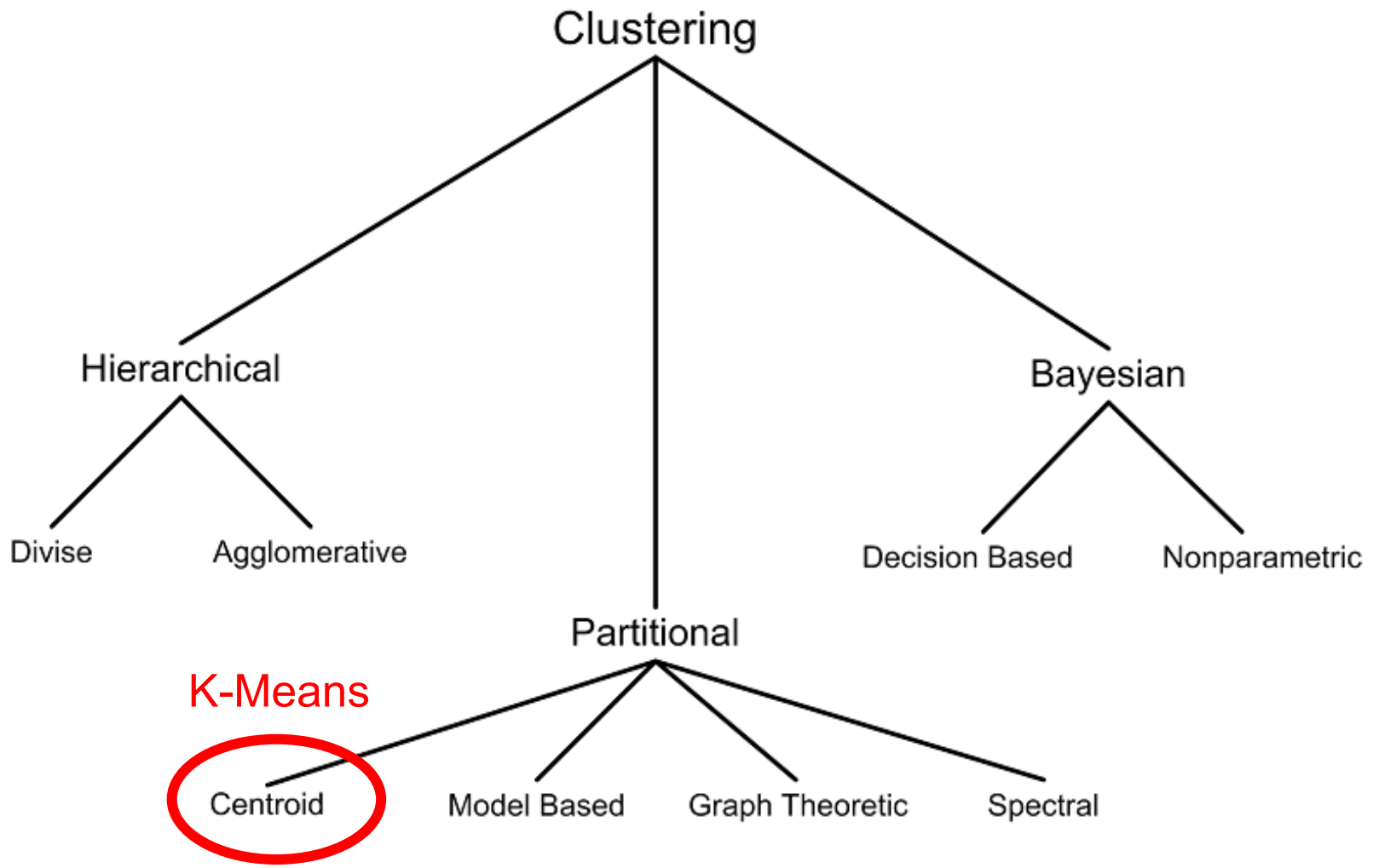
$$L_2^2(x,y) = (2-1)^2 + (2-4)^2 + (3-4)^2 + (-1-0)^2$$
$$= 1^2 + (-2)^2 + (-1)^2 + (-1)^2$$
$$= 1 + 4 + 1 + 1 = 7$$

Clustering techniques



Stockholms universitet

Clustering techniques



Stockholms universitet

K-Means clustering

- K-means is a partitional clustering algorithm
- Given a set X of n points in a d-dimensional space and an integer k
- The K-Means algorithm partitions the given data into k clusters:
  - Each cluster has a cluster center, called centroid.
- **Task:** choose a set of k points $\{c_1, c_2, \dots, c_k\}$ in the d-dimensional space to form clusters $C = \{C_1, C_2, \dots, C_k\}$ such that

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} L_2^2(x, c_i) = \sum_{i=1}^{k} \sum_{x \in C_i} (x - c_i)^2$$

is minimized

Algorithmic properties of K-Means

- NP-hard if the dimensionality of the data is at least 2 ($\boldsymbol{d >= 2}$)
- Finding the best solution in polynomial time is infeasible
- For $\boldsymbol{d = 1}$ the problem is solvable in polynomial time
- A simple iterative algorithm works quite well in practice

The K-Means algorithm

- Randomly pick k cluster centers $\{c_1, \ldots, c_k\}$
- For each i, set the cluster C_i to be the set of points in X that are closer to c_i than they are to c_j for all $i \neq j$
- For each i let c_i be the center of cluster C_i (mean of the vectors in C_i)
- Repeat until convergence

Mean of a vector = mean of all values for each dimension (attribute)

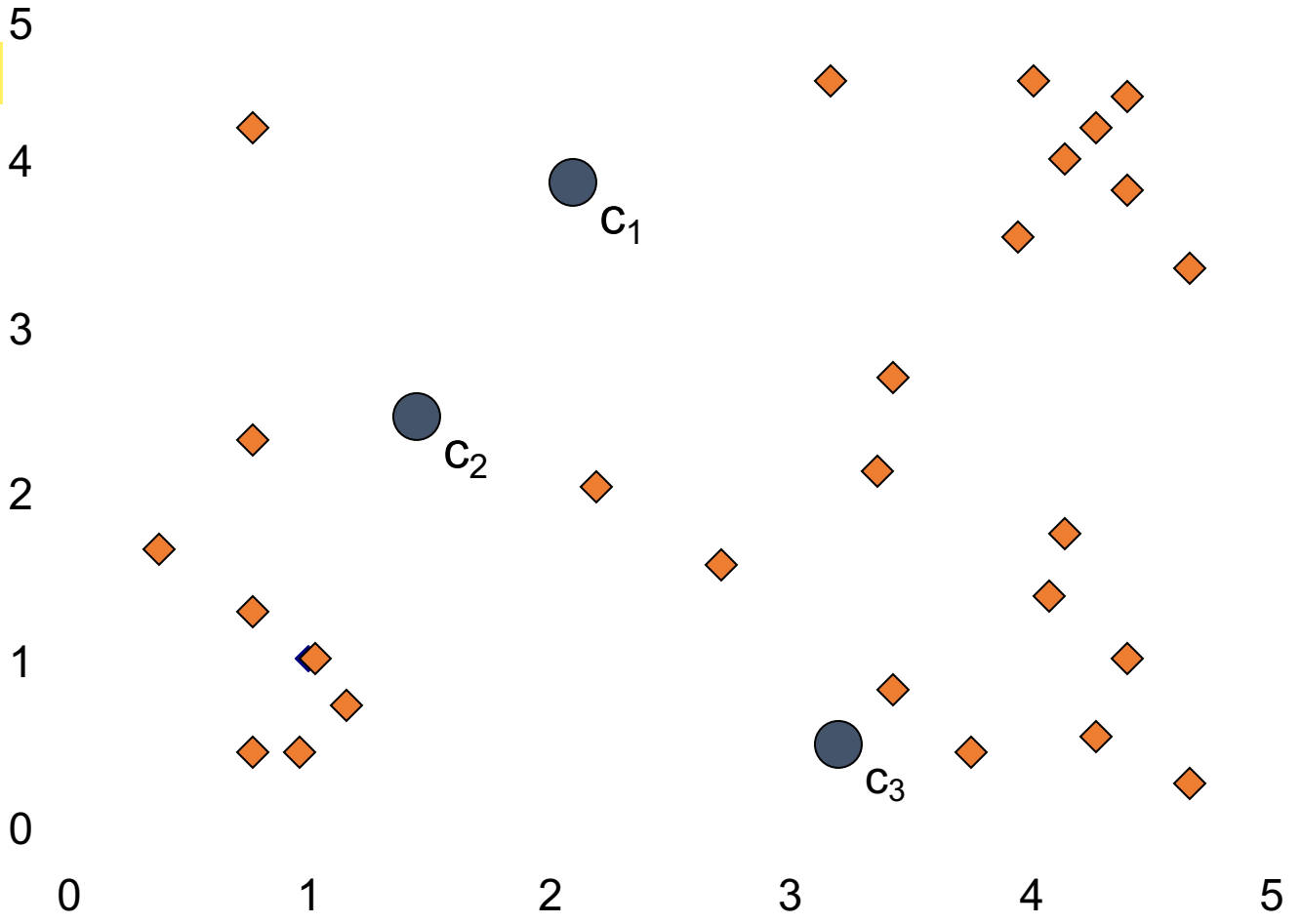
K-Means convergence (stopping) criterion

- No (or minimum) re-assignments of data points to different clusters
- No (or minimum) change of centroids
- Minimum decrease in the Cost function

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} L_2^2 \left(x, c_i \right) = \sum_{i=1}^{k} \sum_{x \in C_i} \left(x - c_i \right)^2$$

K-Means Clustering: Step 1

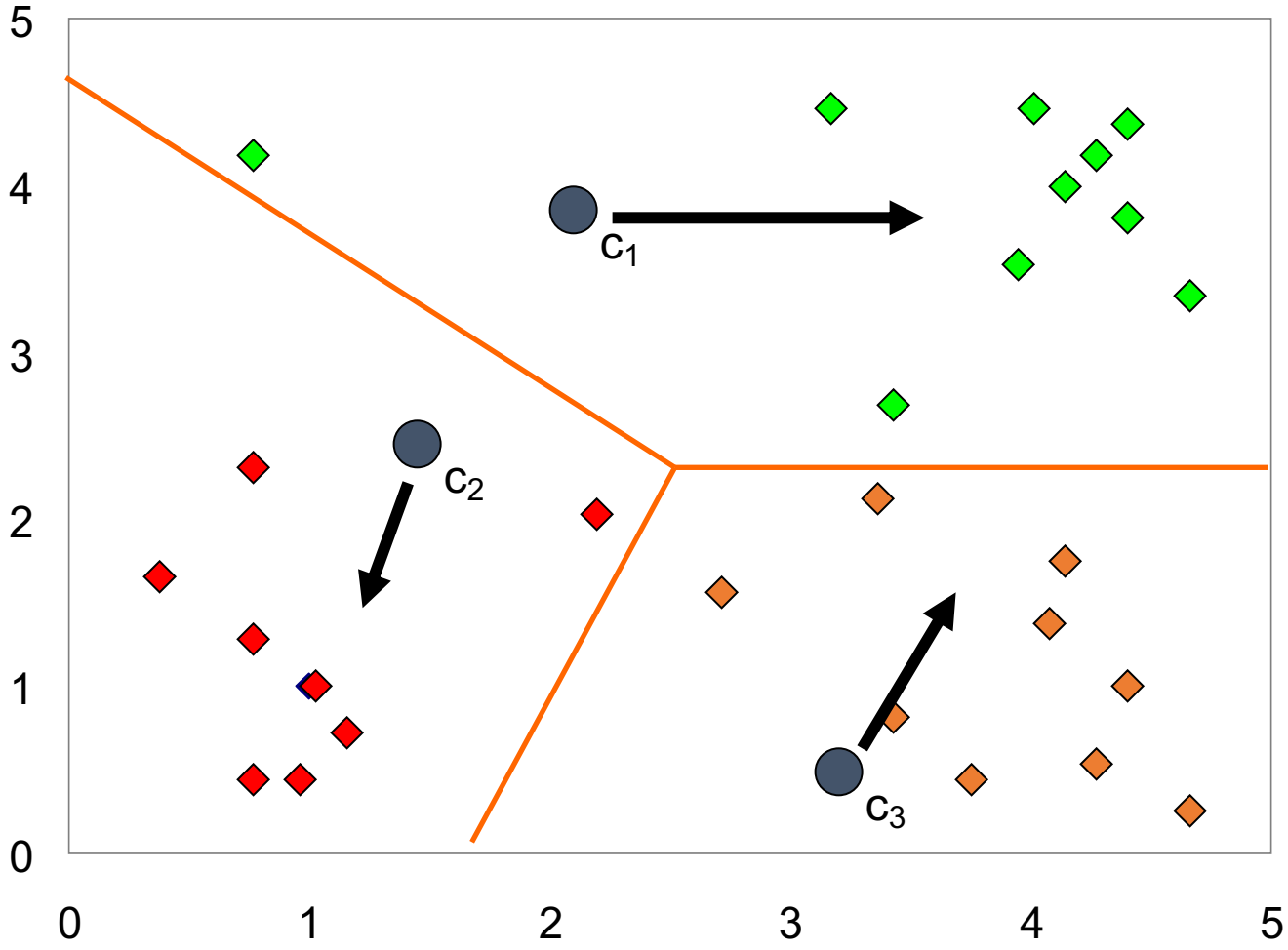
- Randomly initialize the cluster centers
- Distance Metric: Euclidean Distance



Stockholms universitet

K-Means Clustering: Step 2

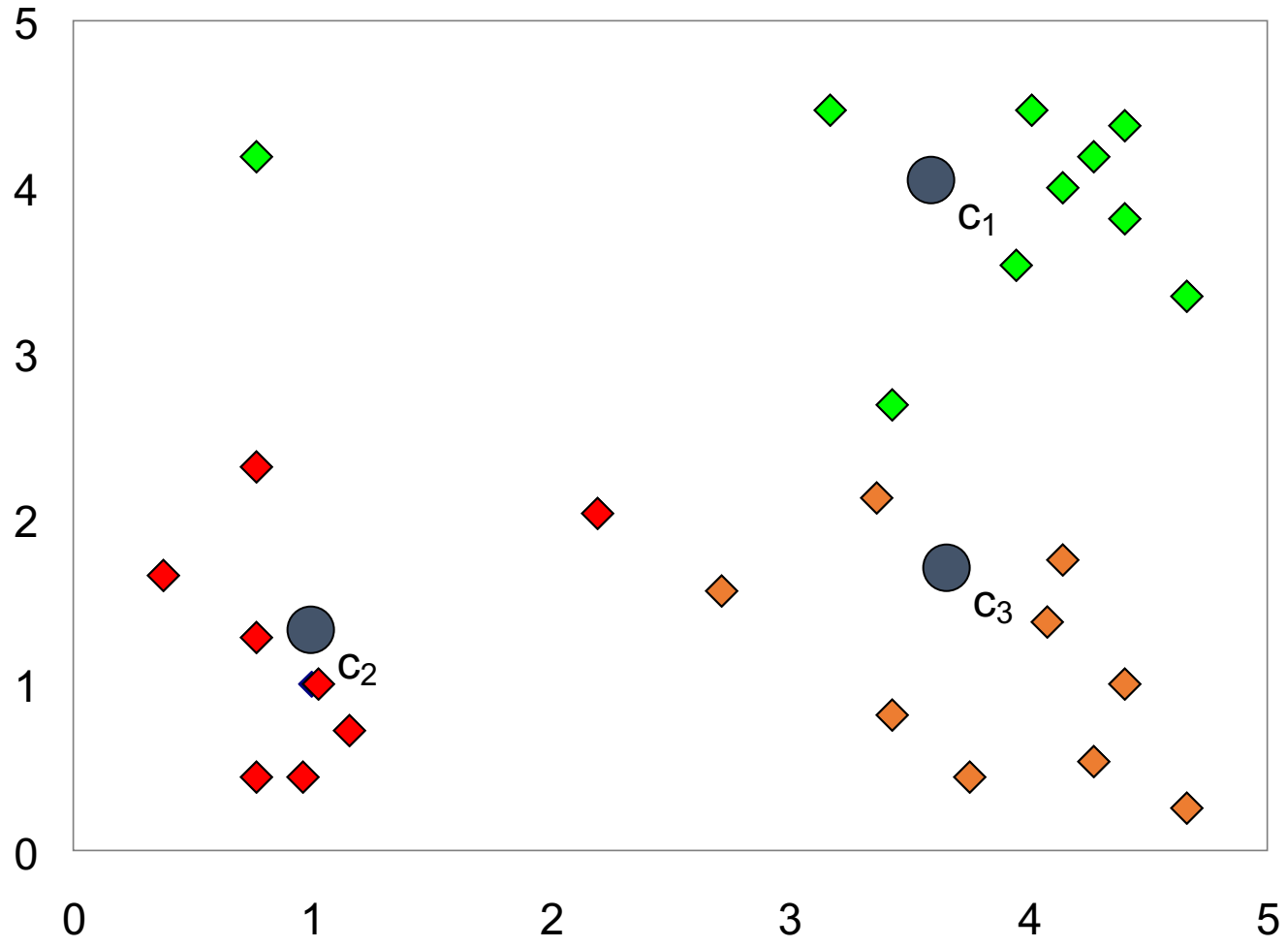
- Determine cluster membership for each point
- Re-estimate cluster centres



Stockholms universitet

K-Means Clustering: Step 3

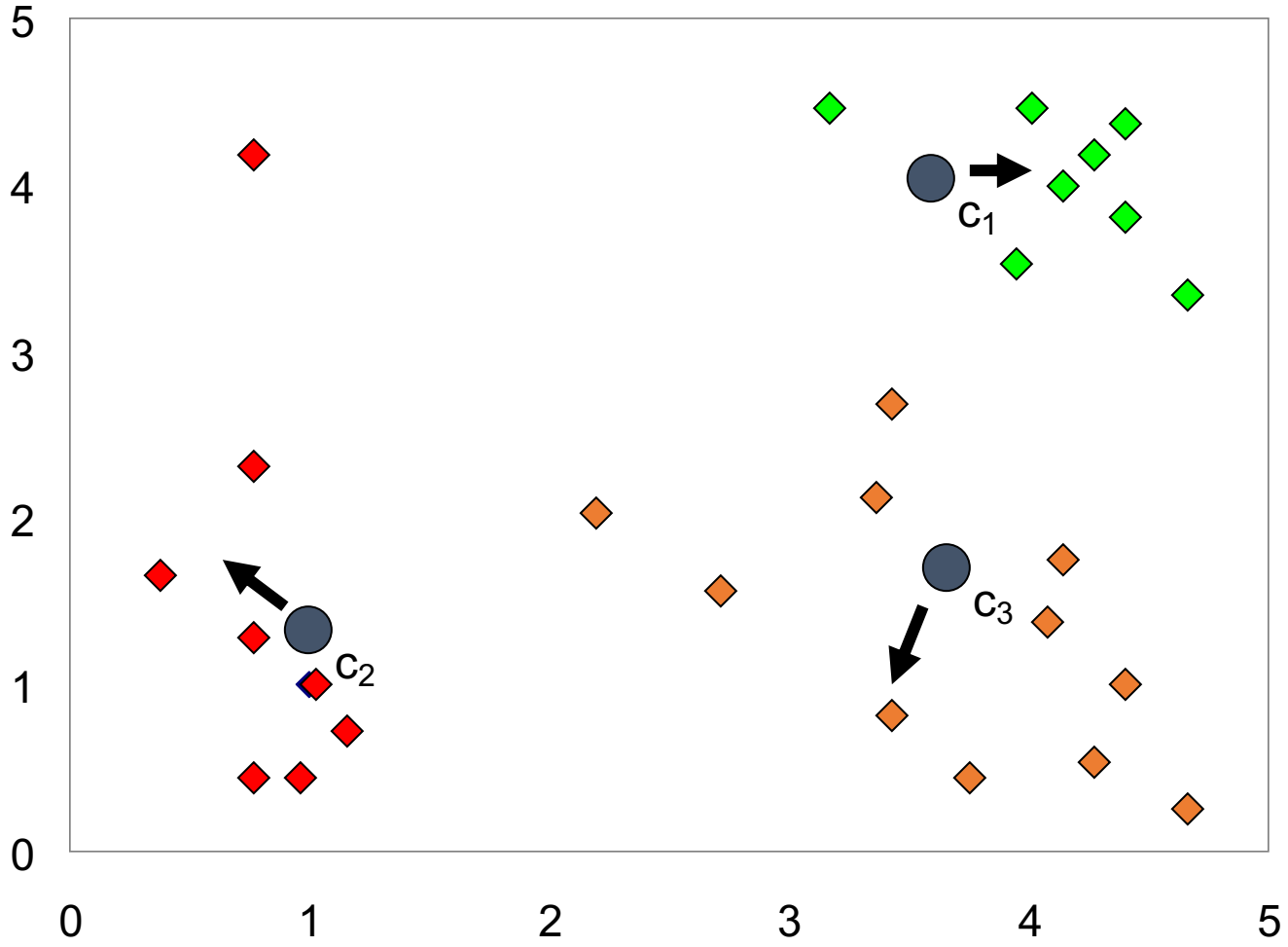
- Results of first iteration



Stockholms universitet

K-Means Clustering: Step 4

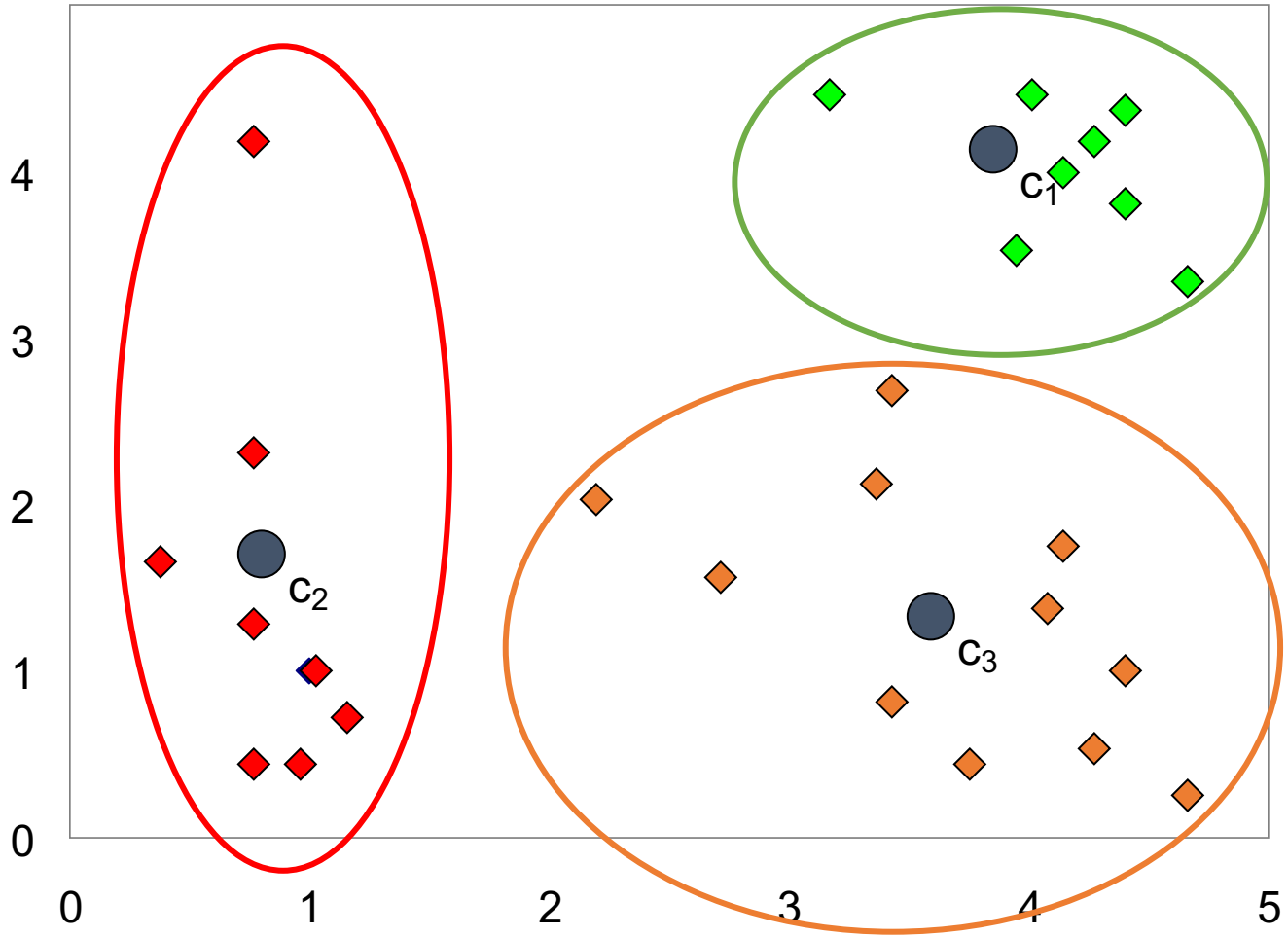
- Second iteration



Stockholms universitet

K-Means Clustering: Step 5

- Result of second iteration



Stockholms universitet

Why use K-Means?

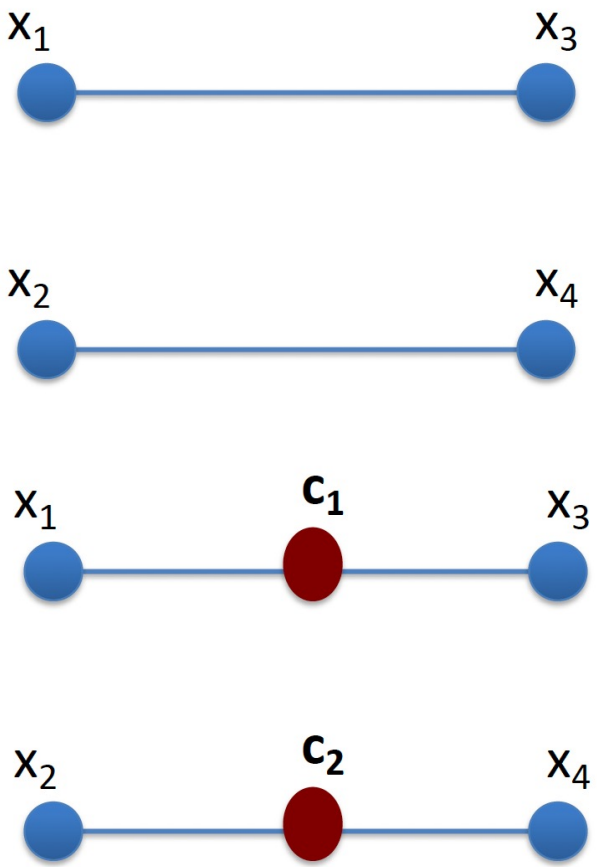
- Finds a **local** optimum
- Converges **often** quickly (but not always)
- Relatively efficient: $O(tKNM)$
  - N: #objects
  - M: #dimensions
  - K: #clusters
  - t: #iterations
- Since both K and t are small, K-Means is considered a linear algorithm.
- It is guaranteed to converge after at most K^N iterations
- The choice of initial points can have large influence in the result

Weaknesses of K-Means

- The algorithm is only applicable if the mean is defined.
  - For categorical data, (K-Mode) the centroid is represented by most frequent values.
    - Gender (Male, Female)
    - Brand of soaps (Dove, Olay…)
    - Hair color (Blonde, Brunette, Brown, Red, etc.)
    - Survey on a topic “Do you have children?” (Yes or No)
- The user needs to specify k.
- The algorithm is sensitive to outliers

Discussion of the k-means algorithm

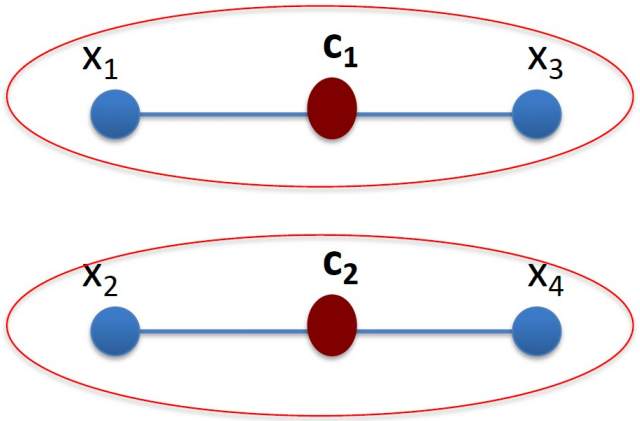
$K = 2$



Stockholms universitet

Discussion of the k-means algorithm

$K = 2$

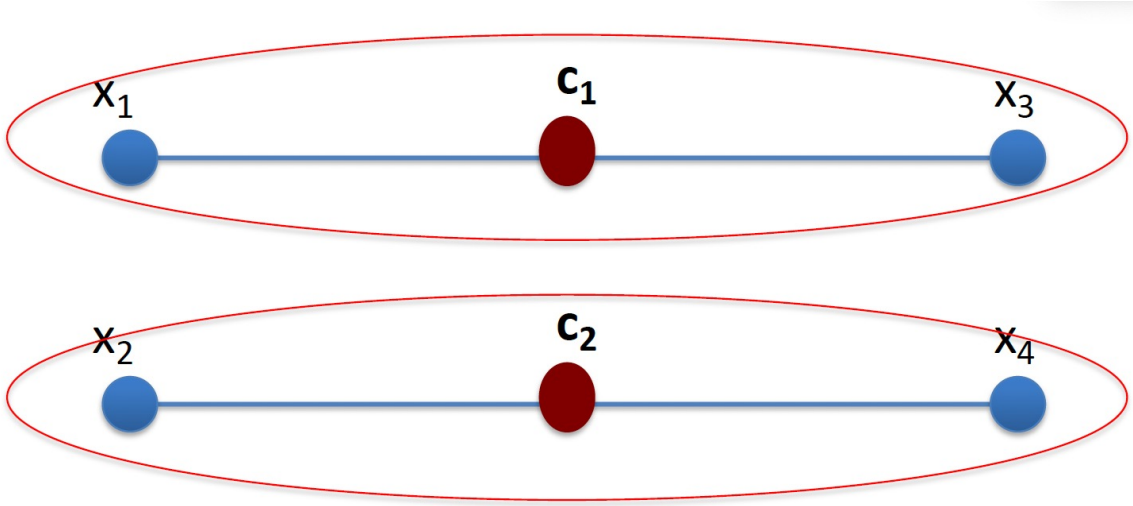


- K-means converges immediately!
- Bad result

Stockholms universitet

Discussion of the k-means algorithm

$K = 2$



- K-means converges immediately!
- Even worse as the horizontal lines stretch

Stockholms universitet

K-means++

1. Choose the first center uniformly at random from X
2. For each data point x, compute $d(x)$
   - distance between x and the nearest center that has already been chosen

3. Choose "the farthest" data point as a new center
   - using a weighted probability distribution where a point x is chosen with probability proportional to $d(x)^2$

- Repeat Steps 2 and 3 until k centers have been chosen

K-Means summary

- Despite weaknesses, K-Means is still the most popular algorithm due to its simplicity and efficiency
- No clear evidence that any other clustering algorithm performs better in general
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

The k-median problem

- Given a set X of n points in a d-dimensional space and an integer k

- **Task:** choose a set of k points $\{c_1, c_2, \ldots, c_k\}$ in the d-dimensional space to form clusters $C = \{C_1, C_2, \ldots, C_k\}$ such that

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} L_1(x, c_i)$$

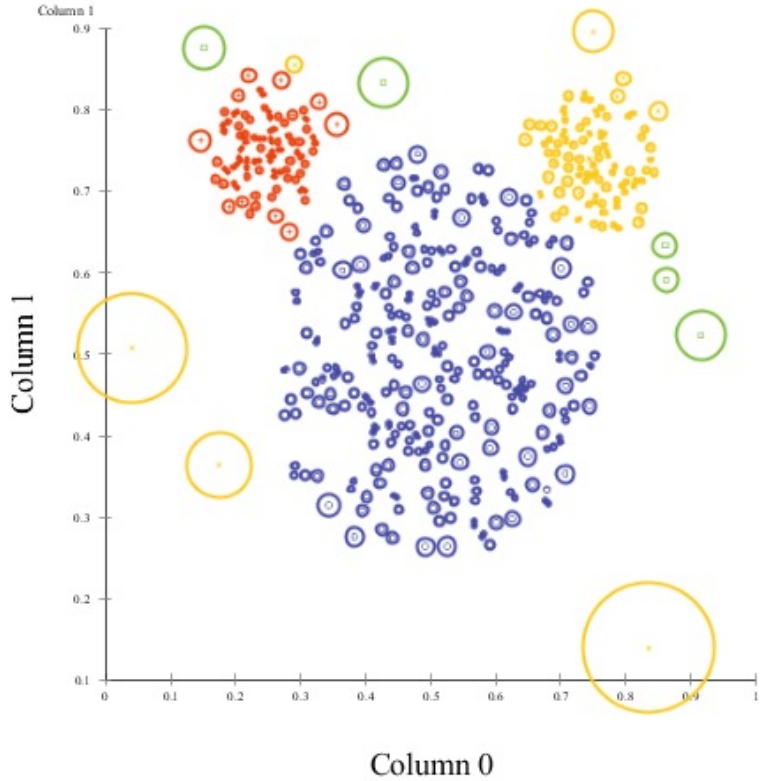
is minimized

- Same as the K-means…but for each dimension we identify the median
- Hence: centroid attributes will come from the dataset
- This makes the algorithm more reliable for discrete-valued or binary datasets

Dealing with outliers

- Remove some data points that are much further away from the centroids than other data points
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.

- Perform random sampling: by choosing a small subset of the data points, the chance of selecting an outlier is much smaller
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification



Stockholms universitet

The K-Medoids algorithm

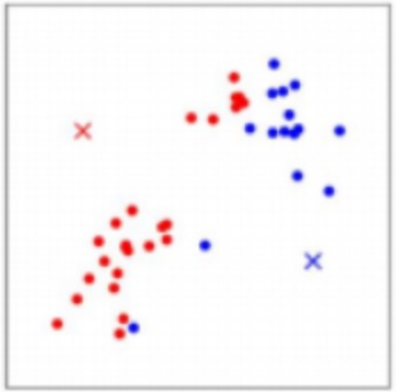
- The K-Medoids algorithm is a clustering algorithm, similar to K-Means
- K-Medoids *Or PAM* (Partitioning Around Medoids, 1987)
  - Choose randomly k medoids from the original dataset X
  - Assign each of the $n - k$ remaining points in X to their closest medoid
  - Iteratively replace one of the medoids by one of the non-medoids if it improves the total clustering cost

K-Means VS. K-Medoids

K-Means

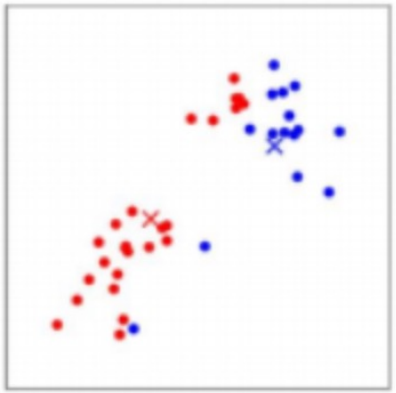
(a)

(b)

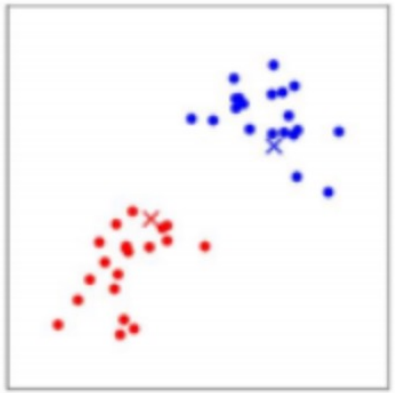


(c)

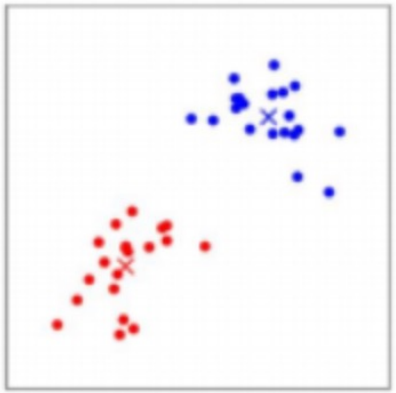
K-Medoids



(d)



(e)



(f)

K-Means VS. K-Medoids

- K-Medoids is more flexible:
  - Can be used with any distance/similarity measure
  - K-Means works only with measures that are consistent with the *mean* (e.g., Euclidean Distance)
- K-Medoids is more robust to outliers:
  - Example: [1 2 3 4 100000]
  - mean = 20002 vs medoid = 3
- K-Medoids is more expensive:
  - All pair-wise distances are computed at each iteration

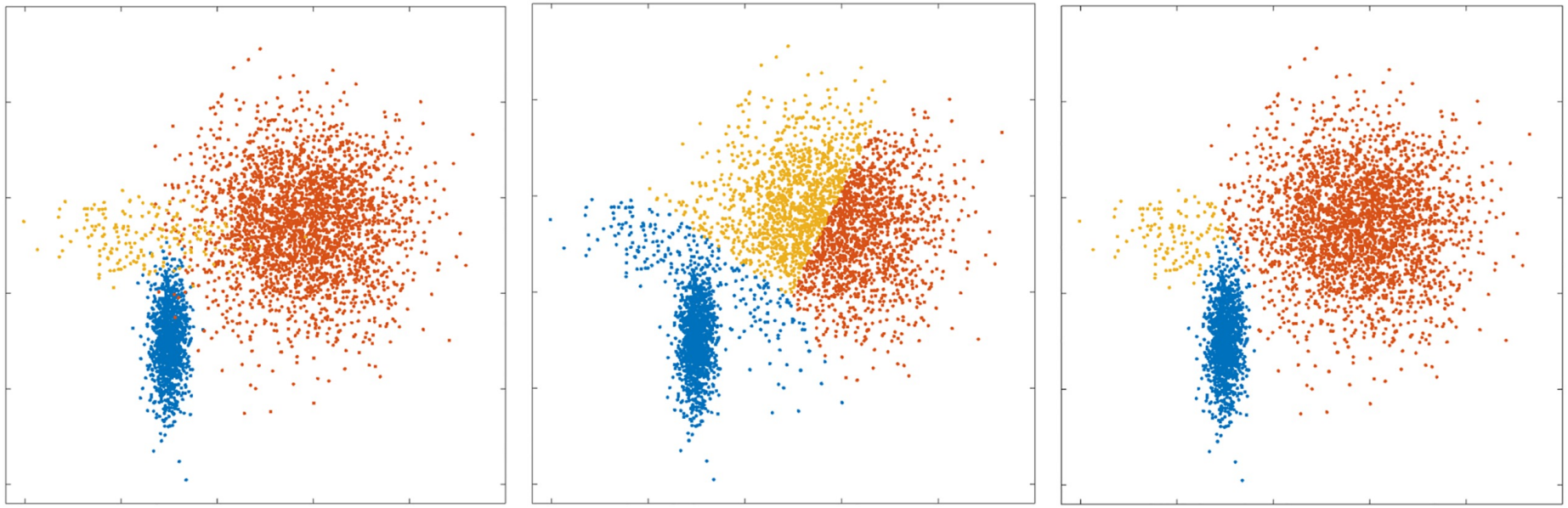
CLARA (Clustering Large Applications)

- It draws multiple samples of the data set, applies PAM on each sample, and gives the best clustering as the output

- Strength
  - deals with larger data sets than PAM

- Weakness*:*
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

How many clusters?



Clustering 1

Clustering 2

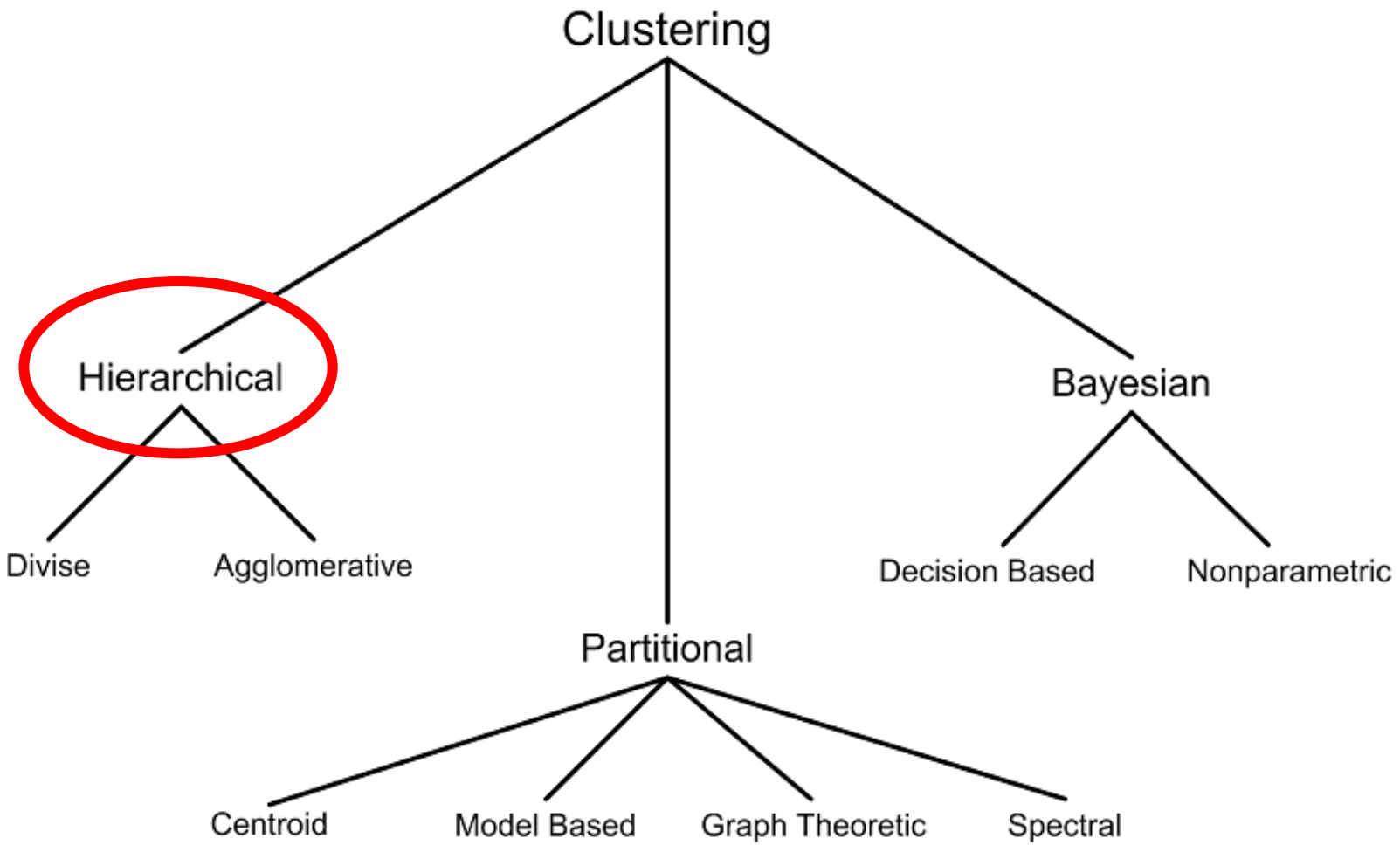
Clustering 3

Stockholms universitet

What is the right number of clusters?

- …or who sets the value of k?
- For n points to be clustered consider the case where $k = n$. What is the value of the error function?
- What happens when $k = 1$?
- X-means:
  - based on the Bayesian Information Criterion
  - Detects the optimal number of clusters

Clustering techniques

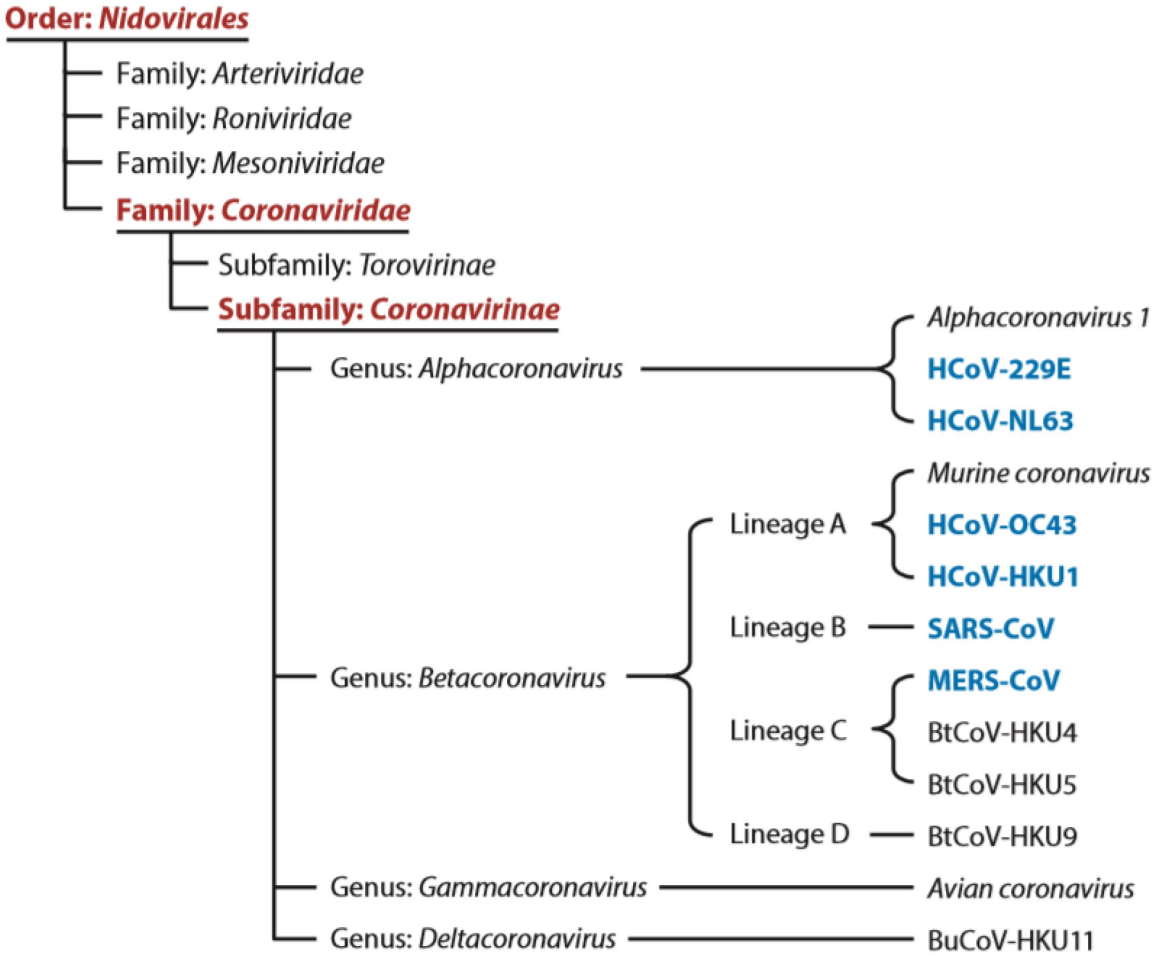


Stockholms universitet

Hierarchical Clustering

- Produces a set of ***nested clusters*** organized as a hierarchical tree

- Can be visualized as a **dendrogram**
  - A tree-like diagram that records the sequences of merges or splits



Strengths of Hierarchical Clustering

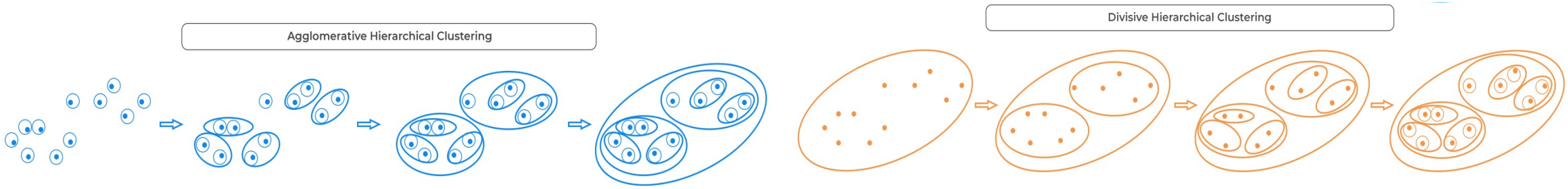
- No assumptions on the number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level

- Hierarchical clusterings may correspond to meaningful taxonomies
  - Example in web (e.g., product catalogs), etc…

Hierarchical Clustering

- Two main types of hierarchical clustering
  - **Agglomerative** (*Bottom-up)*:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster left
  - **Divisive** (*Top-down*):
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point
- Traditional hierarchical algorithms use a similarity or distance matrix
- Merge or split one cluster at a time

Agglomerative & Divisive Clustering



Pros:

- Intuitive and easy to understand the hierarchy of clusters.
- Works well with a wide range of data types and distance metrics.
- It doesn't require specifying the number of clusters beforehand.

Cons:

- Computationally expensive, especially with large datasets.
- Sensitive to the choice of linkage criterion.

Pros:

- Computationally less expensive than agglomerative clustering since it starts with one large cluster.
- It may be more suitable for cases where you have prior knowledge or a good starting point for clustering.

Cons:

- sensitive to the choice of splitting criterion and the order in which clusters are split.
- The hierarchy can be harder to interpret than in agglomerative clustering.

Agglomerative clustering algorithm

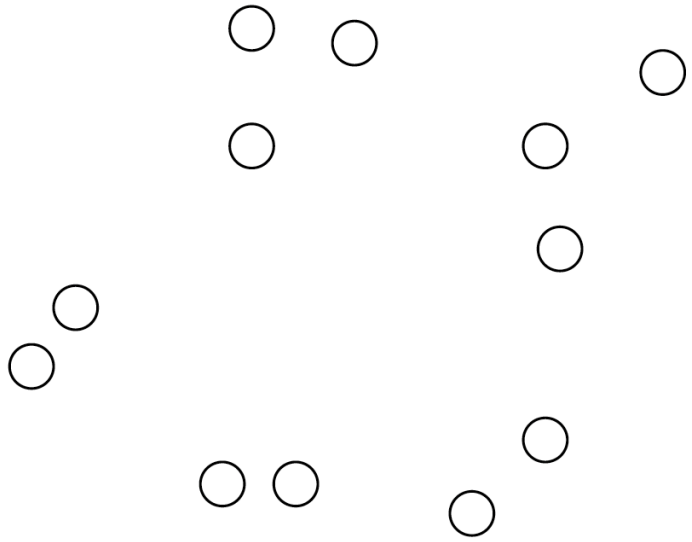
- Most popular hierarchical clustering technique

- Basic algorithm
  1. Compute the distance matrix between the input data points
  2. Let each data point be a cluster
  3. Repeat
  4. Merge the two closest clusters
  5. Update the distance matrix
  6. Until only a single cluster remains

- Key operation is the computation of the distance between two clusters
  - Different definitions of the distance between clusters lead to different algorithms

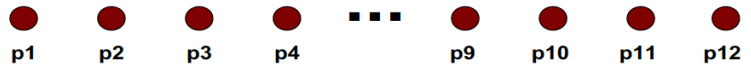
Input/ Initial setting

- Start with clusters of individual points and a distance/proximity matrix



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

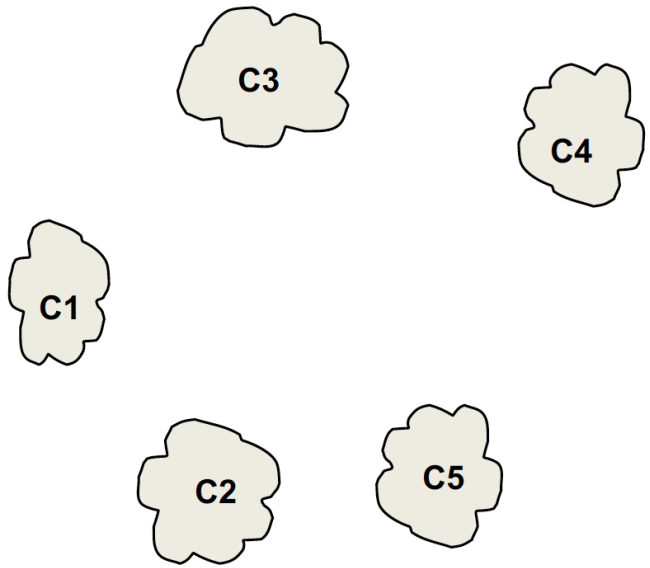
Distance/Proximity Matrix



Stockholms universitet

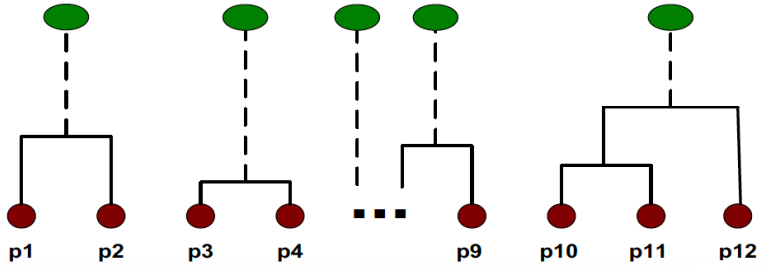
Intermediate State

- After some merging steps, we have some clusters



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

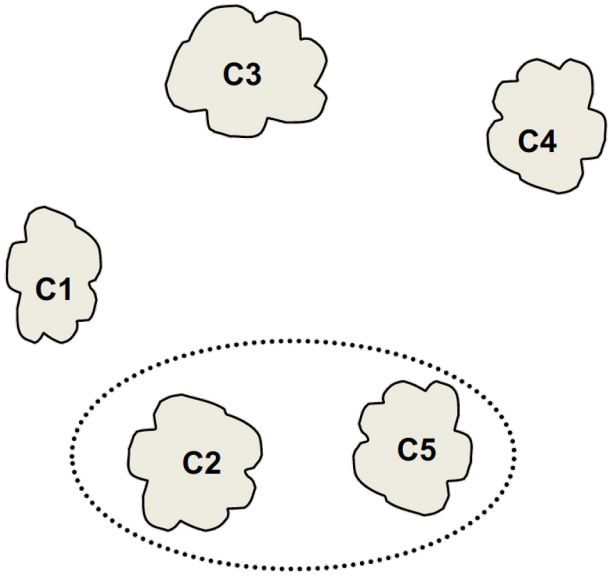
Distance/Proximity Matrix



Stockholms universitet

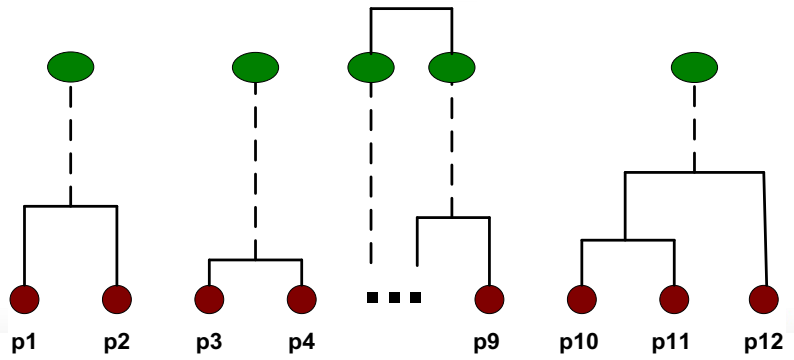
Intermediate State

- Merge the two closest clusters (C2 and C5) and update the distance matrix



	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

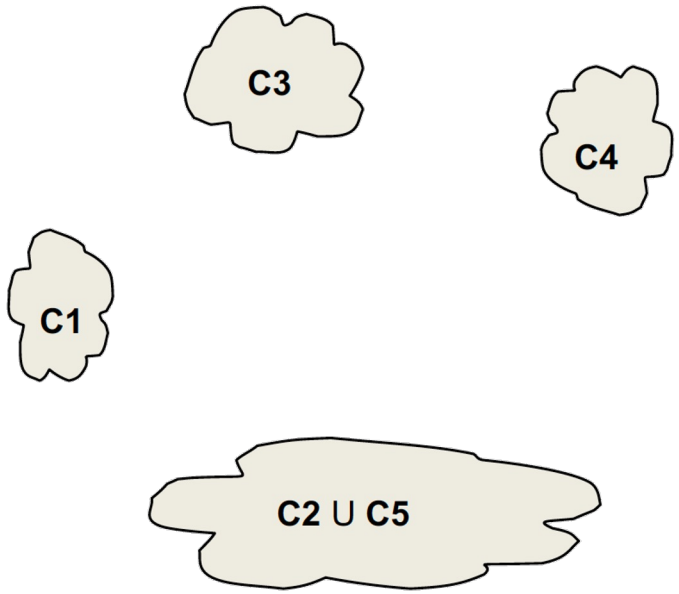
Distance/Proximity Matrix



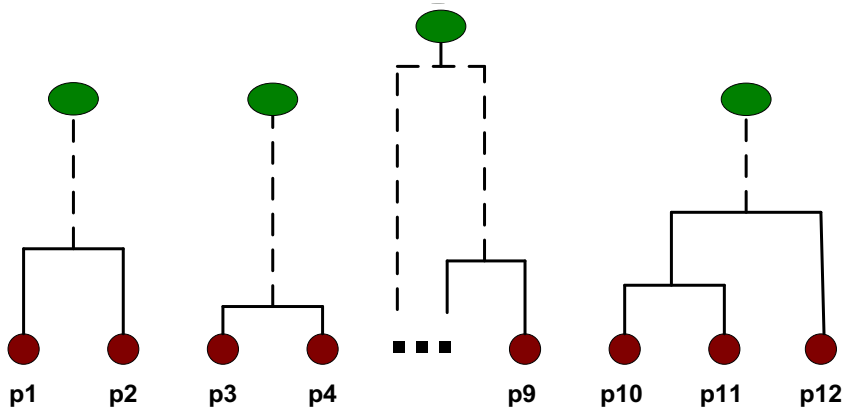
Stockholms universitet

After Merging

- “How do we update the distance matrix?”



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



Stockholms universitet

Agglomerative Clustering

- Let $\{x_1, \ldots, x_k\}$ shows the observations from cluster 1
- Let $\{y_1, \ldots, y_l\}$ shows the observations from cluster 2
- $d(x, y)$ shows distance between a subject with observation vector x and a subject with observation vector y

- Here are four different methods for this approach:
  1. Single Linkage
  2. Complete Linkage
  3. Average Linkage
  4. Centroid Method

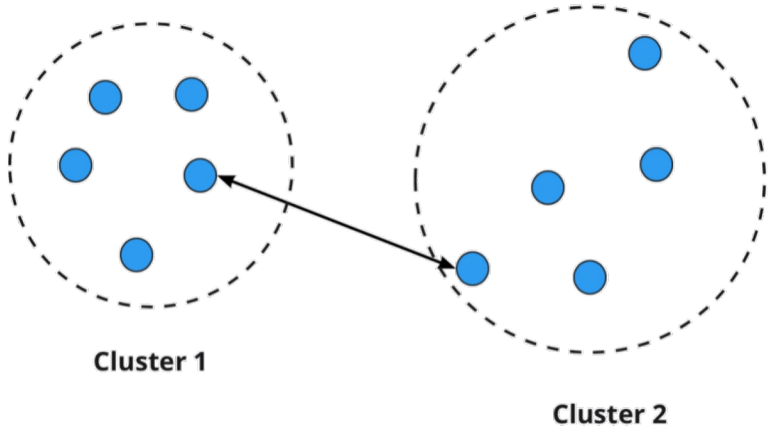
Single (Simple) linkage Clustering

- Define the distance between two clusters as the minimum distance between any single data point in the first cluster and any single data point in the second cluster.

- At each stage of the process we combine the two clusters with the smallest single linkage distance.

- This is the distance between the closest members of the two clusters.

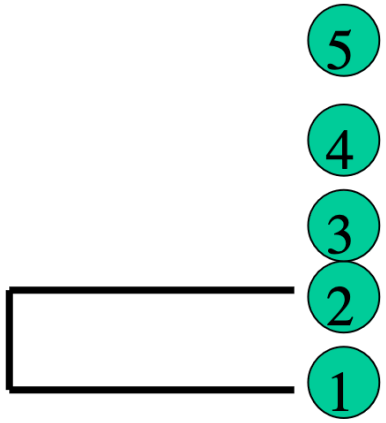
$$d_{12} = \min_{i,j} d(X_i, Y_j)$$



Stockholms universitet

Agglomerative Clustering: Example: single link

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix}
0 & & & & \\
2 & 0 & & & \\
6 & 3 & 0 & & \\
10 & 9 & 7 & 0 & \\
9 & 8 & 5 & 4 & 0
\end{bmatrix}
\end{array}
$$



Stockholms universitet

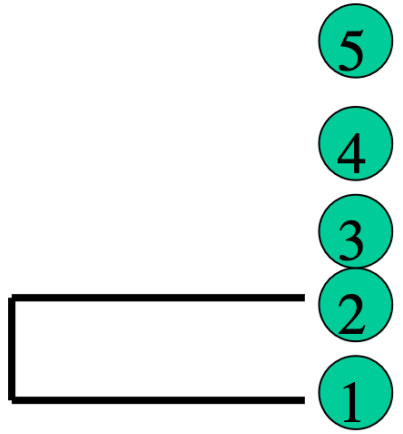
Agglomerative Clustering: Example: single link

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 9 & 7 & 0 & \\ 8 & 5 & 4 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6,3\} = 3$$

$$d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10,9\} = 9$$

$$d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9,8\} = 8$$



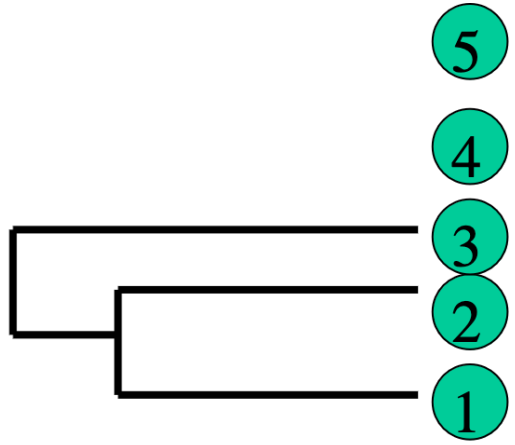
Stockholms universitet

Agglomerative Clustering: Example: single link

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 9 & 7 & 0 & \\ 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2,3) \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccc} (1,2,3) & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & \\ 7 & 0 & \\ 5 & 4 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2,3),4} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9,7\} = 7$$

$$d_{(1,2,3),5} = \min\{d_{(1,2),5}, d_{3,5}\} = \min\{8,5\} = 5$$

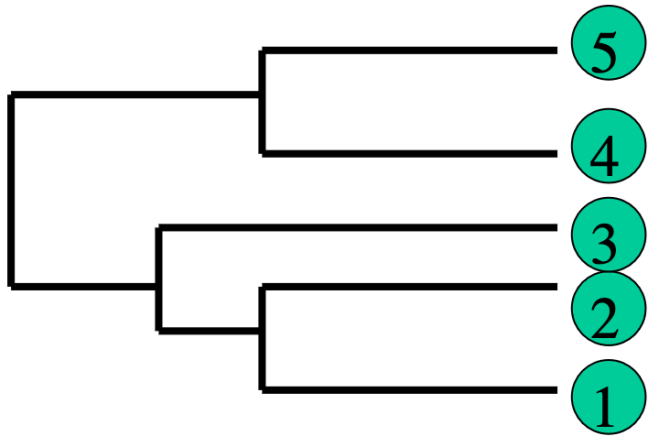


Stockholms universitet

Agglomerative Clustering: Example: single link

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{cccc} (1,2) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 9 & 7 & 0 & \\ 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \Rightarrow \begin{array}{c} \\ (1,2,3) \\ 4 \\ 5 \end{array}\begin{array}{c} \begin{array}{ccc} (1,2,3) & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & \\ 7 & 0 & \\ 5 & 4 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = 5$$

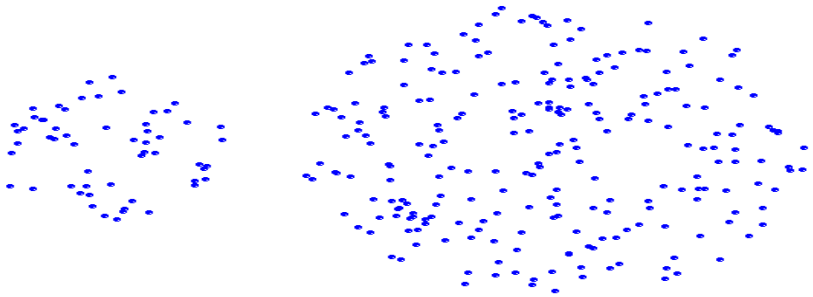


Stockholms universitet

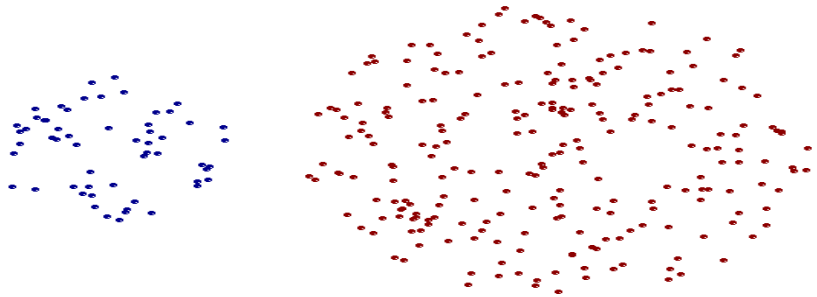
Pros and Cons of Simple Linkage

Pros of Simple Linkage

- Simple Linkage methods can handle non-elliptical shapes.
- Single Linkage algorithms are the best for capturing clusters of different sizes.



Original Points

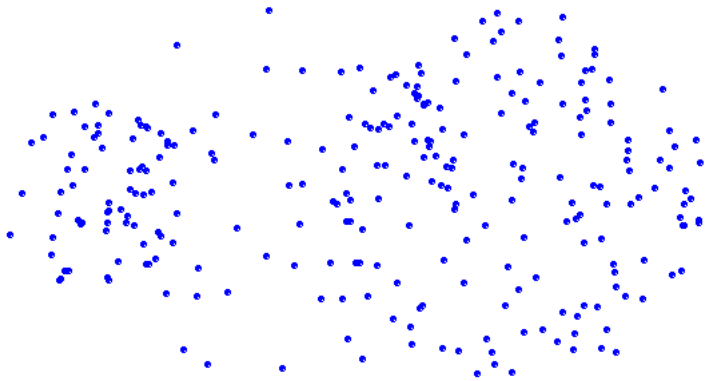


Two Clusters

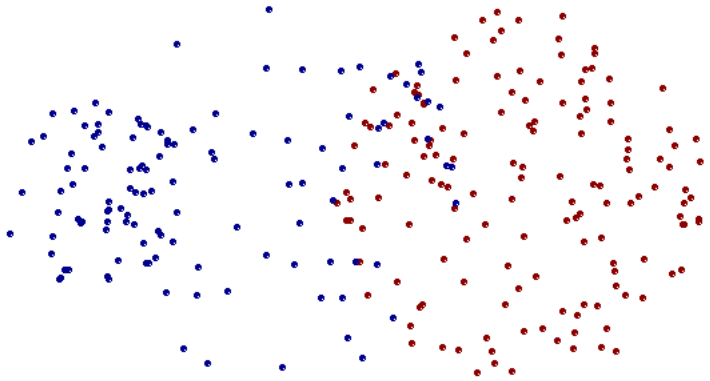
Pros and Cons of Simple Linkage

Cons of Simple Linkage

- Simple Linkage methods are sensitive to noise and outliers.
- That means Simple Linkage methods can not group clusters properly if there is any noise between the clusters.



Original Points



Two Clusters

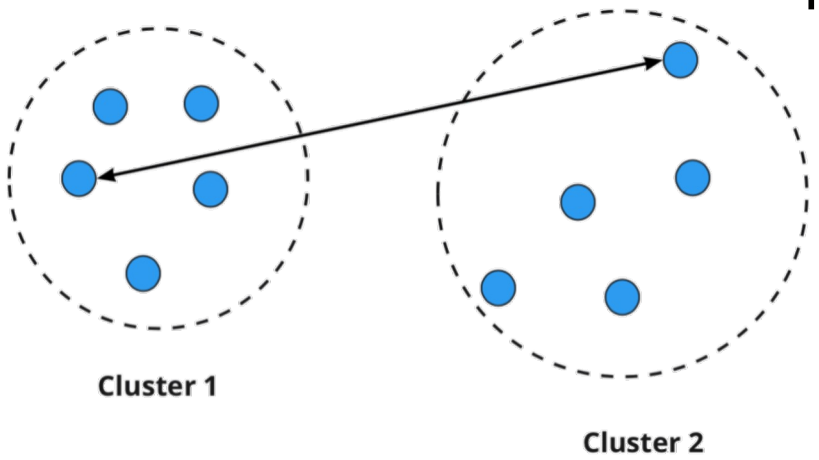
Complete linkage Clustering

- Define the distance between two clusters to be the maximum distance between any single data point in the first cluster and any single data point in the second cluster.

- At each stage of the process we combine the two clusters that have the smallest complete linkage distance.

- This is the distance between the members that are farthest apart (most dissimilar)

$$d_{12} = \max_{i,j} d(X_i, Y_j)$$

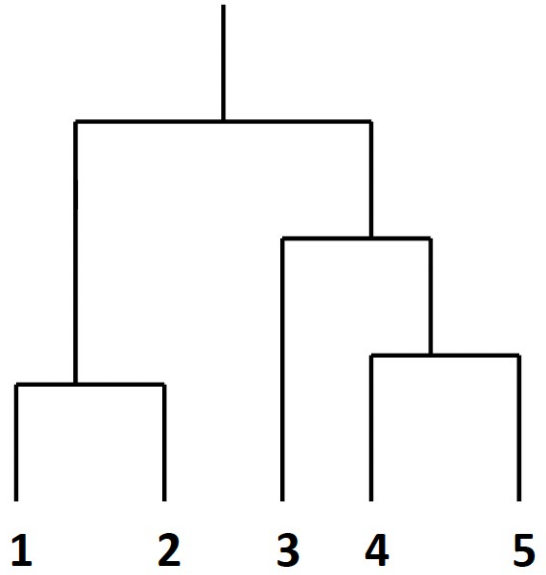


Stockholms universitet

Agglomerative Clustering: Example: Complete link

- Distance between clusters is determined by the two most distant points in the different clusters

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

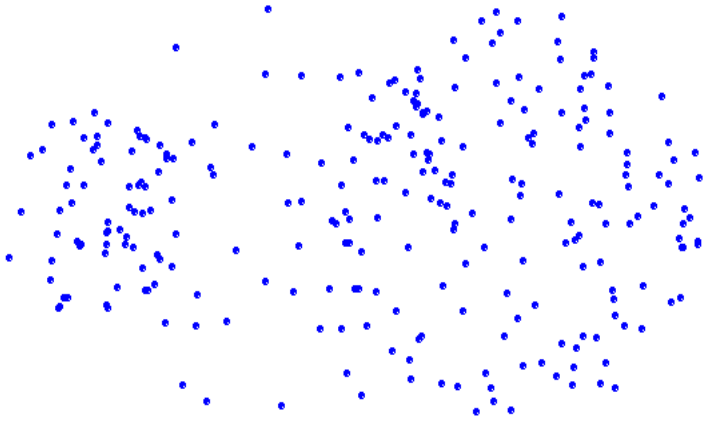


Stockholms universitet

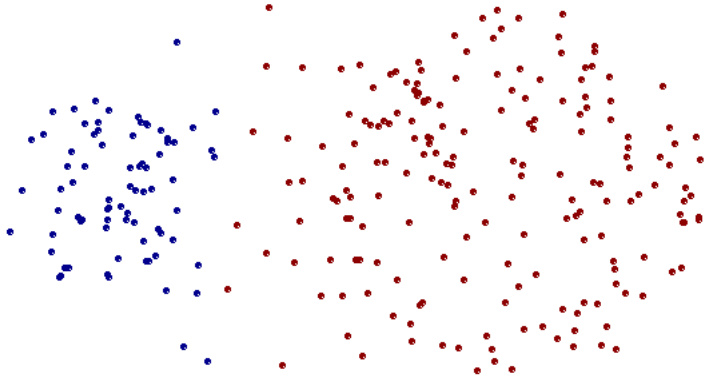
Pros and Cons of Complete Linkage

Pros of Complete Linkage

- Complete Linkage algorithms are less susceptible to noise and outliers.
- That means the Complete Linkage method also does well in separating clusters if there is any noise between the clusters.



Original Points

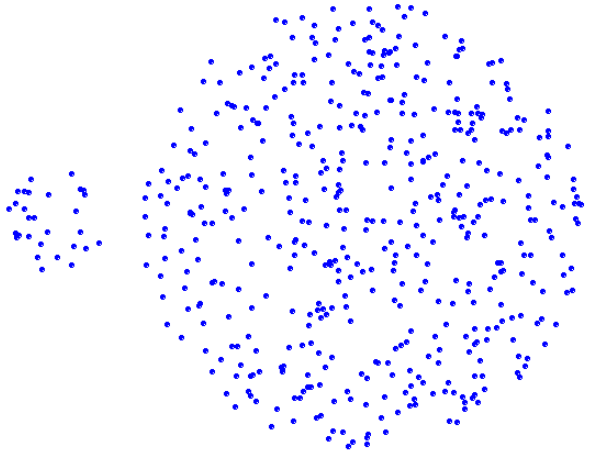


Two Clusters

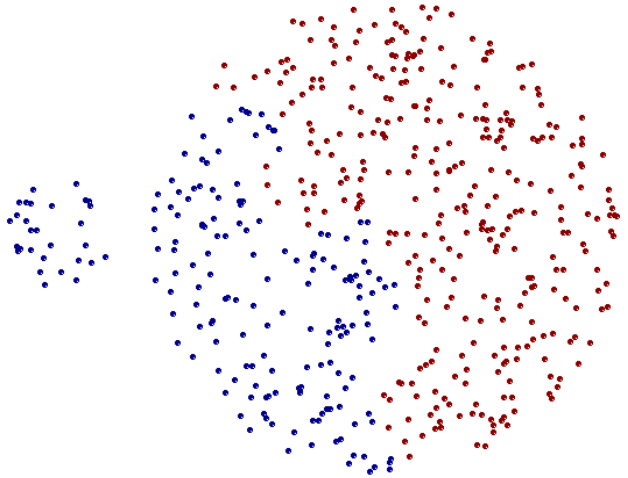
Pros and Cons of Complete Linkage

Cons of Complete Linkage

- Complete linkage methods tend to break large clusters.
- Complete Linkage is biased towards globular clusters.



Original Points



Two Clusters

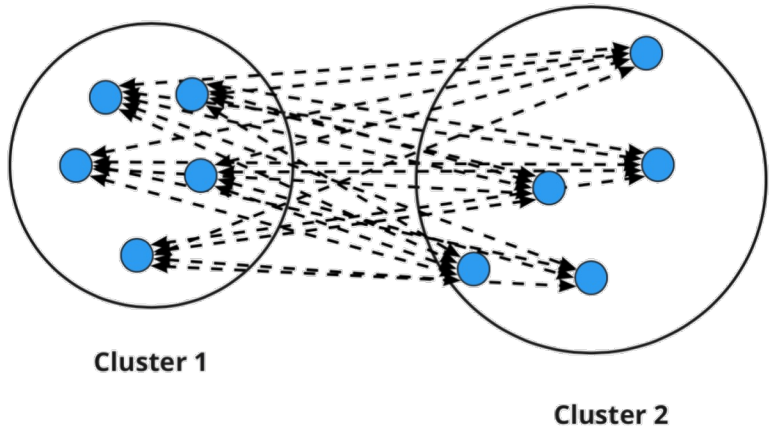
Average linkage Clustering

- Define the distance between two clusters to be the average distance between data points in the first cluster and data points in the second cluster.

- At each stage of the process we combine the two clusters that have the smallest average linkage distance.

- It is also called UPGMA - Unweighted Pair Group Mean Averaging.

$$d_{12} = \frac{1}{kl}\sum_{i=1}^{k}\sum_{j=1}^{l} d(X_i, Y_j)$$

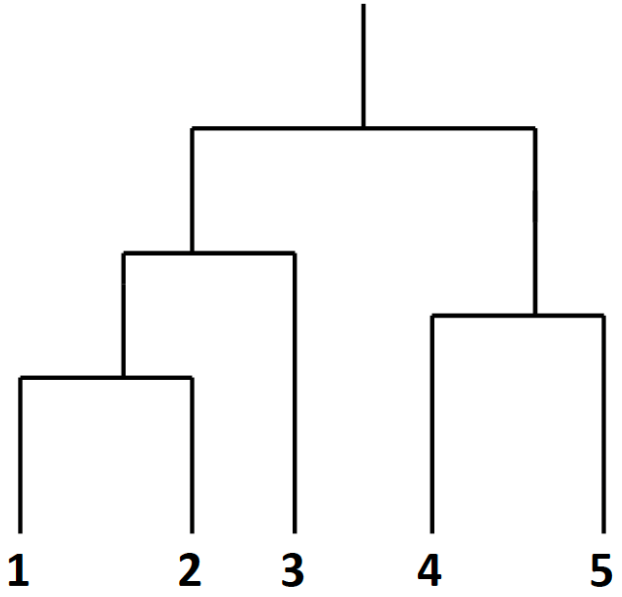


Stockholms universitet

Agglomerative Clustering: Example: Average link

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Stockholms universitet

Pros and Cons of Average Linkage

Pros of Average Linkage	Cons of Average Linkage
✔ The average Linkage method also does well in separating clusters if there is any noise between the clusters.	✖ The average Linkage method is biased towards globular clusters.

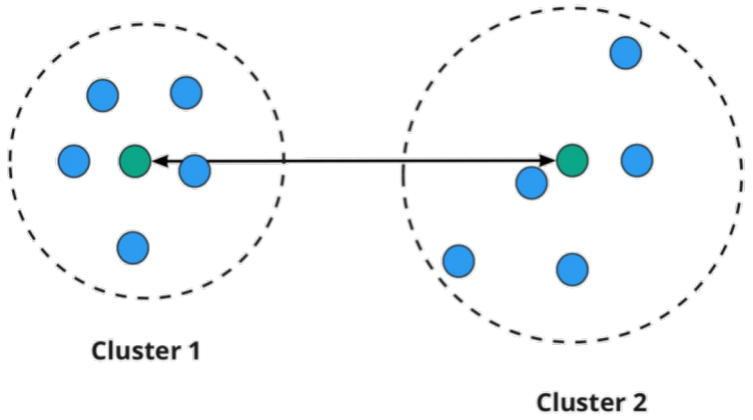
Centroid linkage Clustering

- Define the distance between two clusters as the distance between the two mean vectors of the clusters.

- At each stage of the process we combine the two clusters that have the smallest centroid distance.

- This involves finding the mean vector location for each of the clusters and taking the distance between the two centroids.

$$d_{12} = d(\overline{x}, \overline{y})$$



Stockholms universitet

Pros and Cons of Centroid Linkage

Pros of Centroid Linkage

- The Centroid Linkage method also does well in separating clusters if there is any noise between the clusters.

Cons of Centroid Linkage

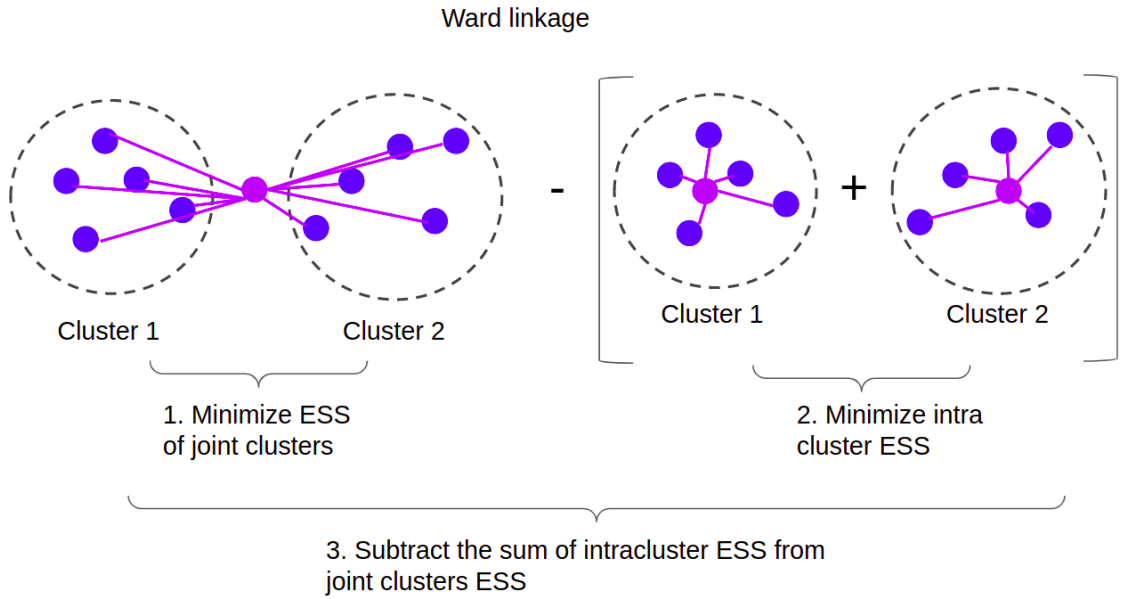
- Similar to Complete Linkage and Average Linkage methods, the Centroid Linkage method is also biased towards globular clusters.

Distance between two clusters

- Ward's distance between clusters C_i and C_j the difference between:
  - the total within cluster sum of squares for the two clusters separately
  - the within cluster sum of squares resulting from merging the two clusters in cluster C_{ij}

$$D_w(C_i, C_j) = \left| \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2 \right|$$

- r_i : centroid (mean) of C_i
- r_j : centroid (mean) of C_j
- r_{ij} : centroid (mean) of C_{ij}



Stockholms universitet

Ward's distance for clusters

- Similar to average-link
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of k-means
  - Can be used to initialize k-means

Hierarchical Clustering: Time and Space requirements

- For a dataset X consisting of n points
- $O(n^2)$ **space**; it requires storing the distance matrix
- $O(n^3)$ **time** in most of the cases
  - There are n steps and at each step the size n^2 distance matrix must be updated and searched
  - Complexity can be reduced to $O(n^2 \log(n))$ time by using appropriate data structures

Today …

What is **Clustering**?

What are the **distance measures**?

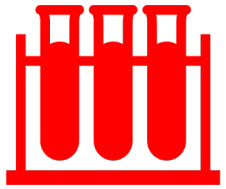
Partitional clustering algorithms

Hierarchical clustering algorithms

TODOs

Reading:

Main course book: chapter 7



Lab 1

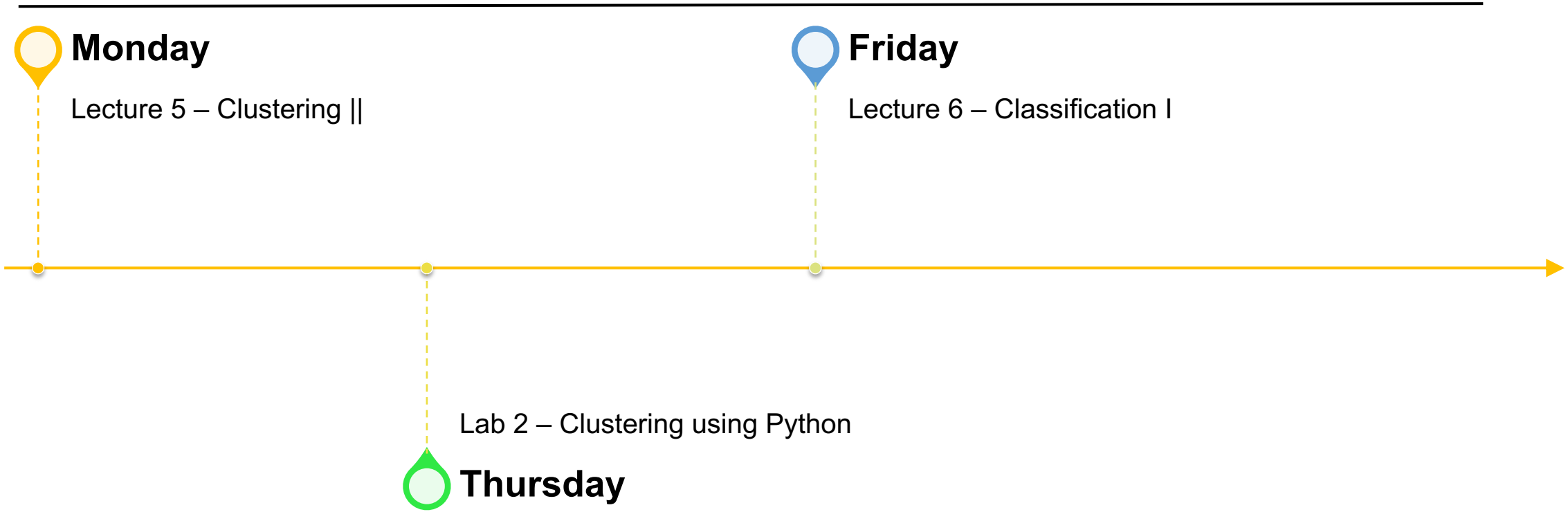
Due: Sep 14, 23:59

Quiz 1

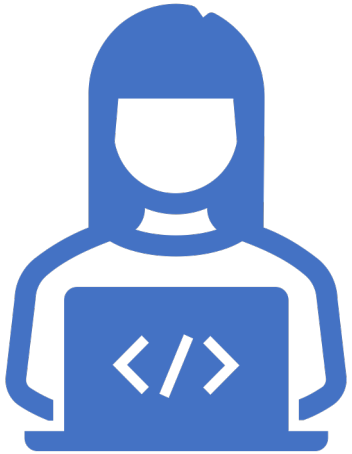
Due: Sep 10, 23:59

Stockholms universitet

Coming up next



Stockholms universitet



Thanks!

golnaz.taheri@dsv.su.se

Stockholms universitet