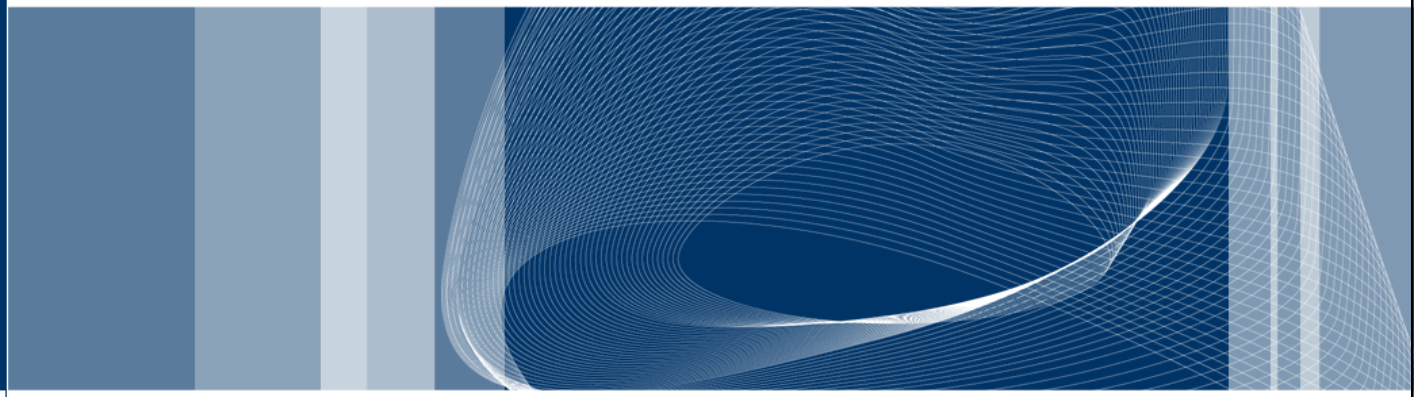
POLITECNICO DI MILANO



Hierarchical Clustering

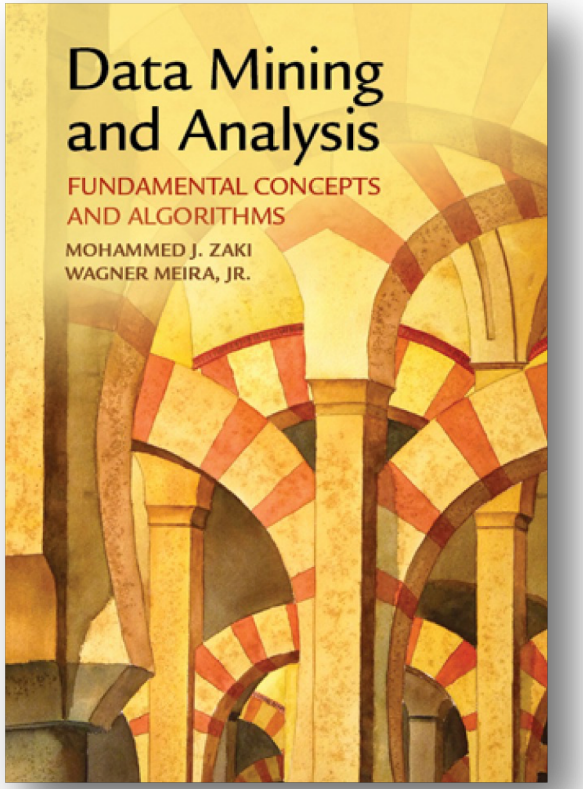
Data Science for Mobility

Prof. Pier Luca Lanzi

Readings

- "Data Mining and Analysis" by Zaki & Meira
  - Chapter 14

- http://www.dataminingbook.info





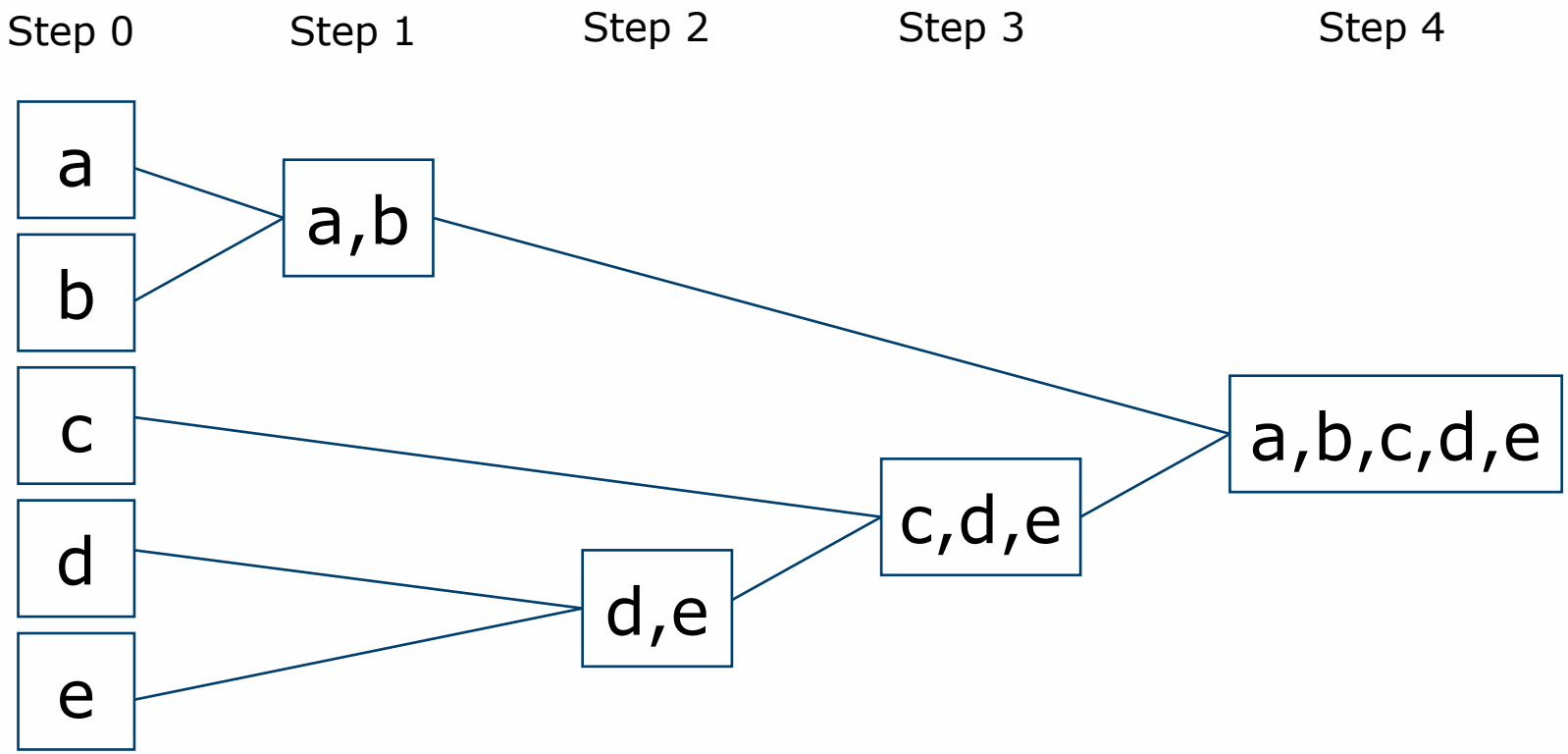






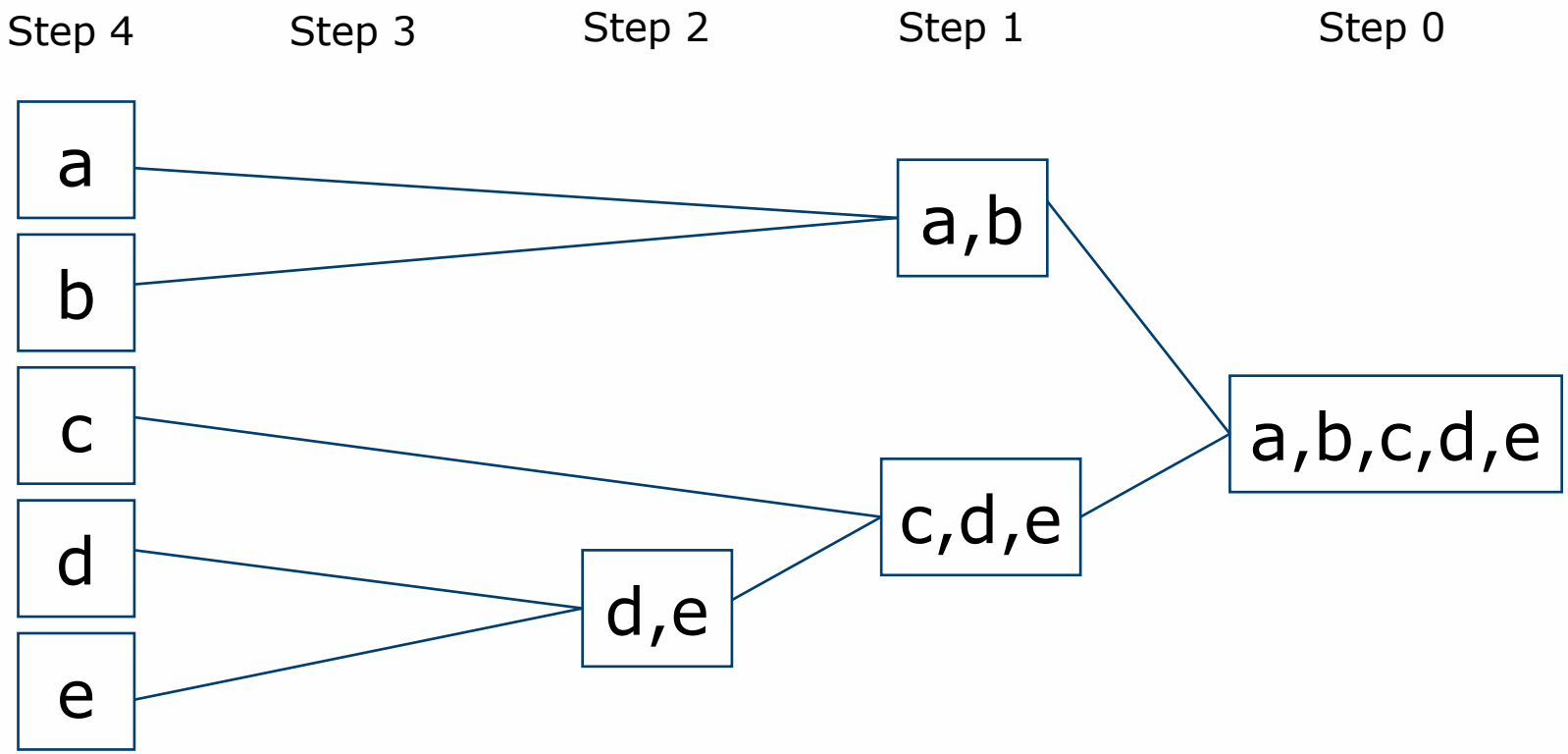

What is Hierarchical Clustering?

- Suppose we have five items, a, b, c, d, and e.
- Initially, we consider one cluster for each item
- Then, at each step we merge together the most similar clusters, until we generate one cluster



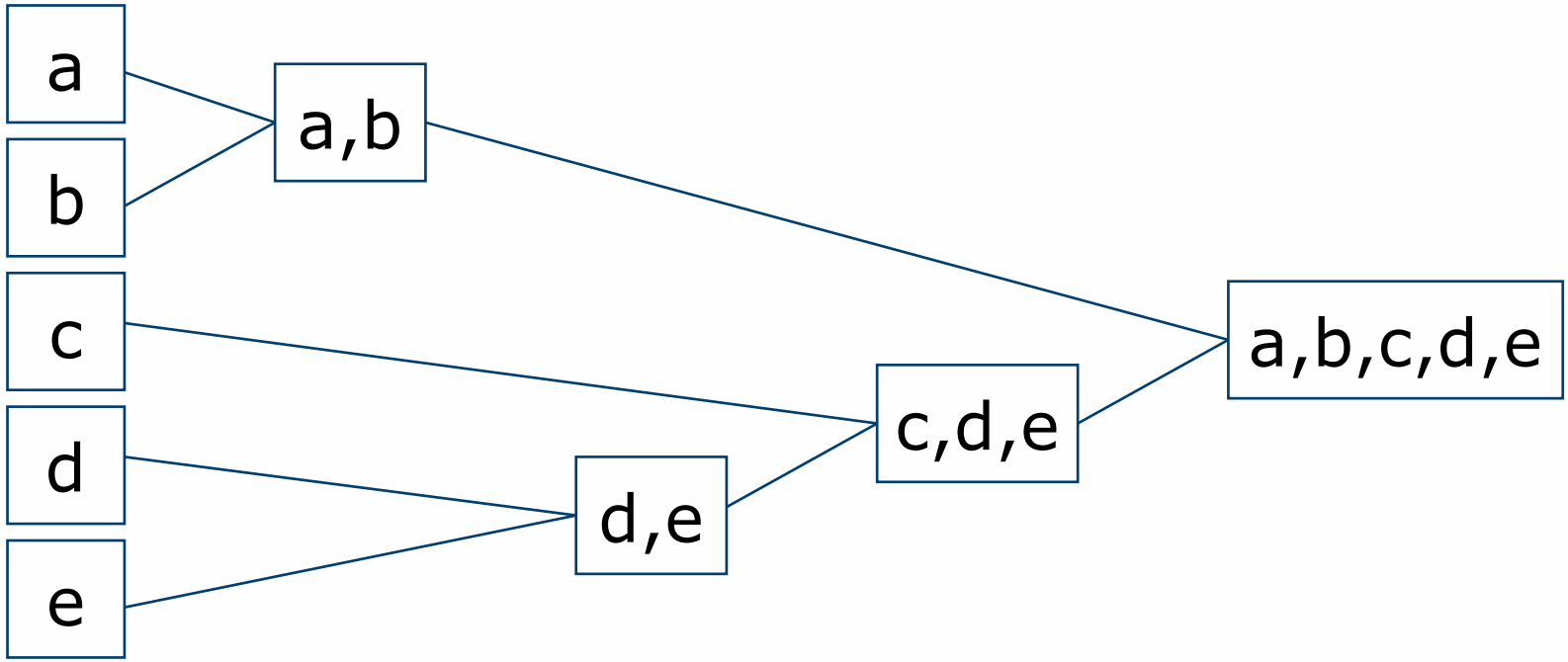
What is Hierarchical Clustering?

- Alternatively, we start from one cluster containing the five elements
- Then, at each step we split one cluster to improve intracluster similarity, until all the elements are contained in one cluster



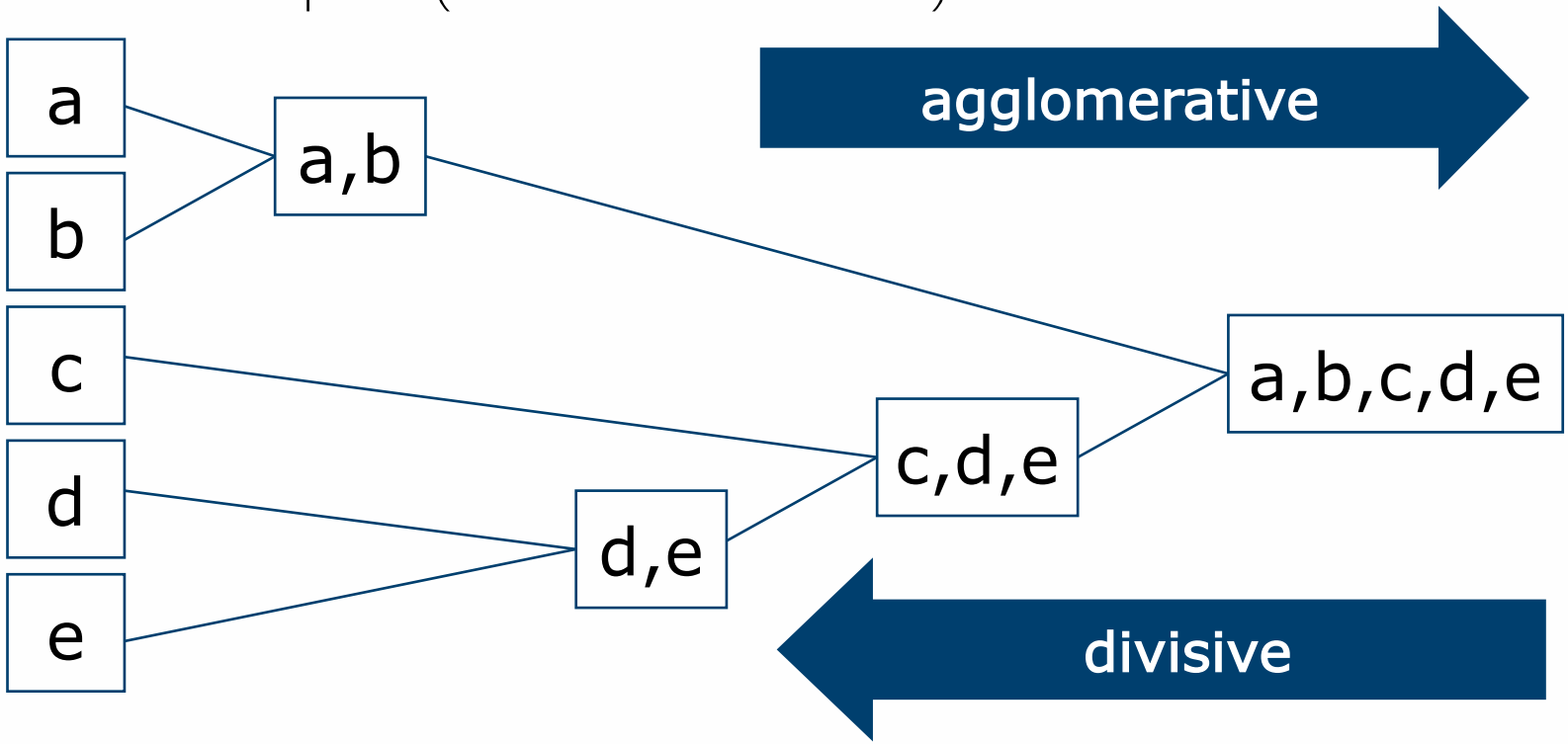
What is Hierarchical Clustering?

- By far, it is the most common clustering technique
- Produces a hierarchy of nested clusters
- The hiearchy be visualized as a dendrogram: a tree like diagram that records the sequences of merges or splits



What Approaches?

- Agglomerative
  - Start individual clusters, at each step, merge the closest pair of clusters until only one cluster (or k clusters) left
- Divisive
  - Start with one cluster, at each step, split a cluster until each cluster contains a point (or there are k clusters)



Strengths of Hierarchical Clustering

- No need to assume any particular number of clusters
- Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level
- They may correspond to meaningful taxonomies
- Example in biological sciences include animal kingdom, phylogeny reconstruction, etc.
- Traditional hierarchical algorithms use a similarity or distance matrix to merge or split one cluster at a time

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique

- Compute the proximity matrix
- Let each data point be a cluster
- Repeat
  - Merge the two closest clusters
  - Update the proximity matrix
- Until only a single cluster remains

- Key operation is the computation of the proximity of two clusters
- Different approaches to defining the distance between clusters distinguish the different algorithms

Hierarchical Clustering: Time and Space Requirements

- $O(N^2)$ space since it uses the proximity matrix.
  - N is the number of points.

- $O(N^3)$ time in many cases
  - There are N steps and at each step the size, N^2, proximity matrix must be updated and searched
  - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

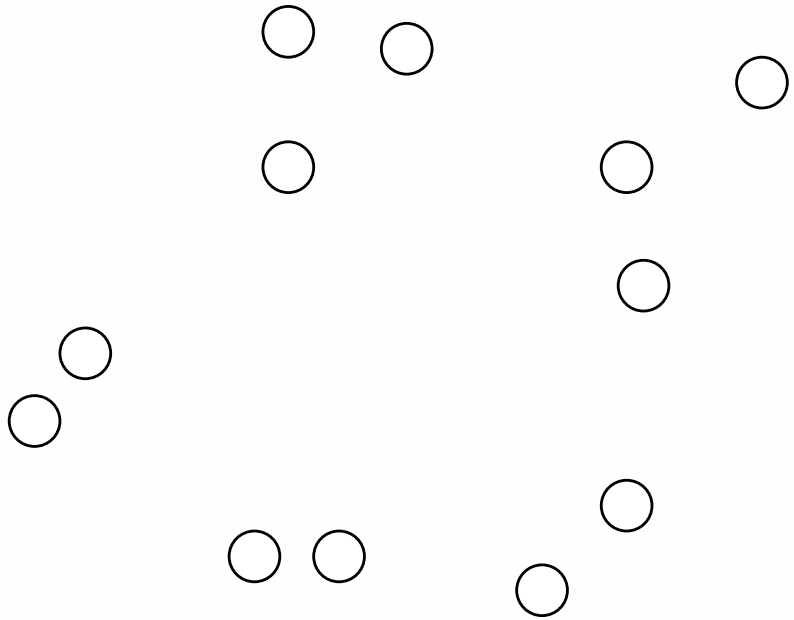
Efficient Implementation

- Compute the distance between all pairs of points [$O(N^2)$]
- Insert the pairs and their distances into a priority queue to fine the min in one step [$O(N^2)$]
- When two clusters are merged, we remove all entries in the priority queue involving one of these two clusters [O(Nlog N)]
- Compute all the distances between the new cluster and the re- maining clusters [O(NlogN)]
- Since the last two steps are executed at most N time, the complexity of the whole algorithms is $O(N^2 \log N)$

Distance Between Clusters

Initial Configuration

- Start with clusters of individual points and the distance matrix

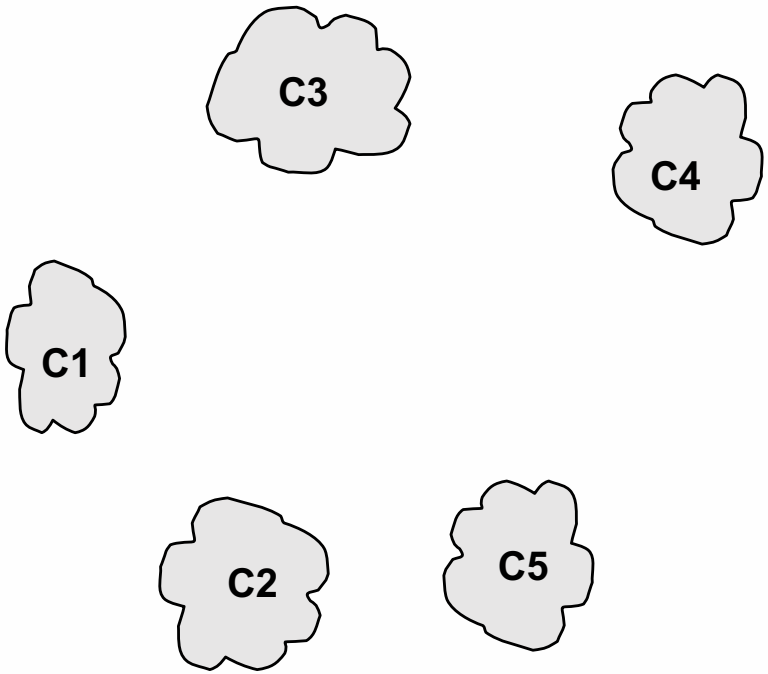


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix

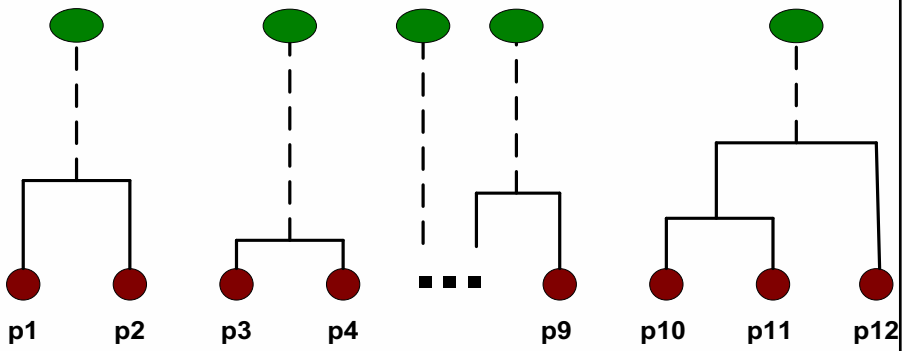
Intermediate Situation

- After some merging steps, we have some clusters



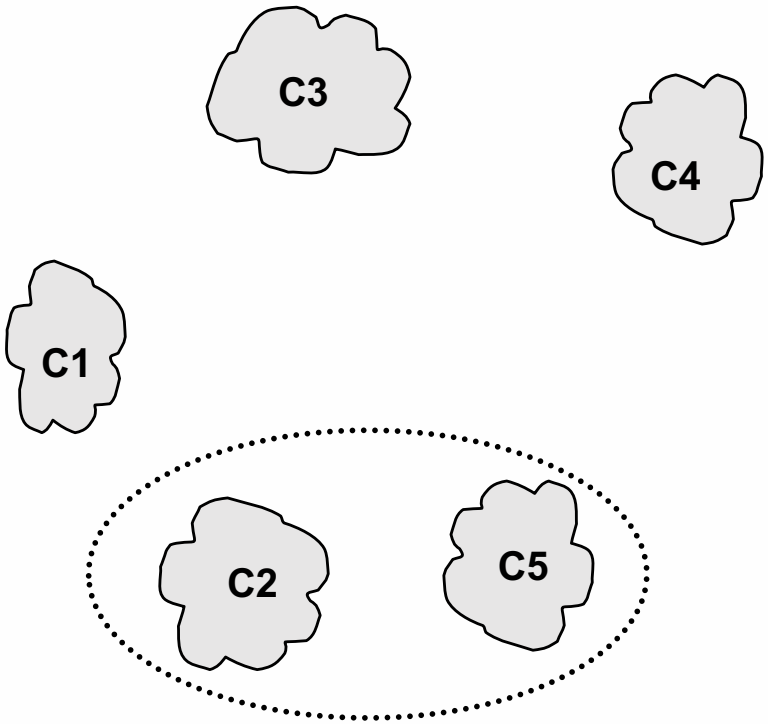
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance Matrix



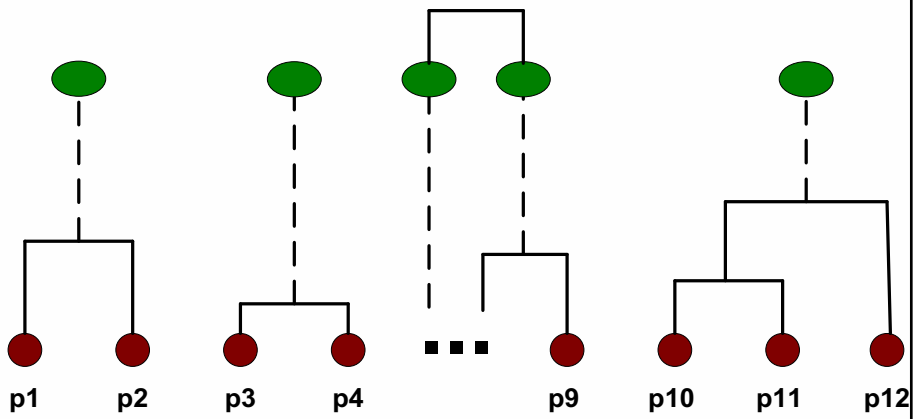
Intermediate Situation

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



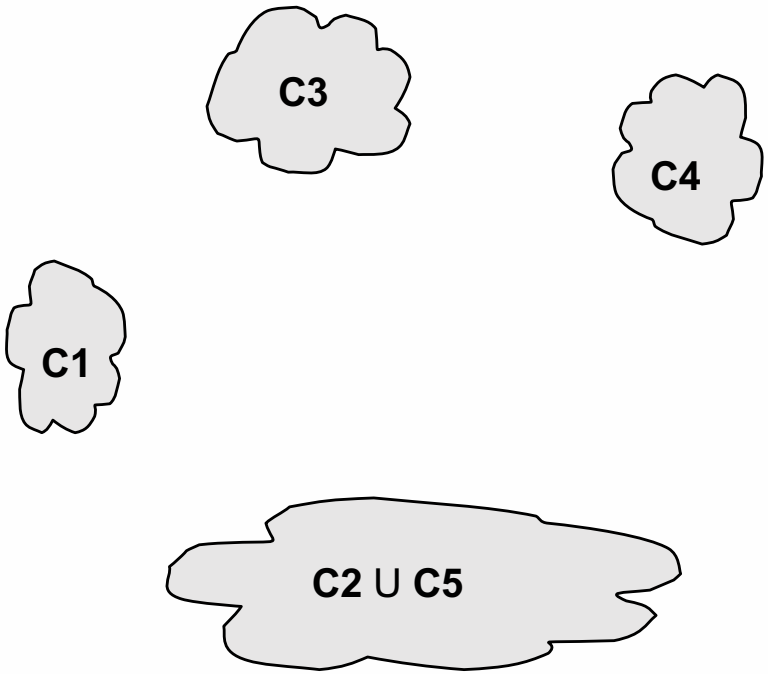
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance Matrix



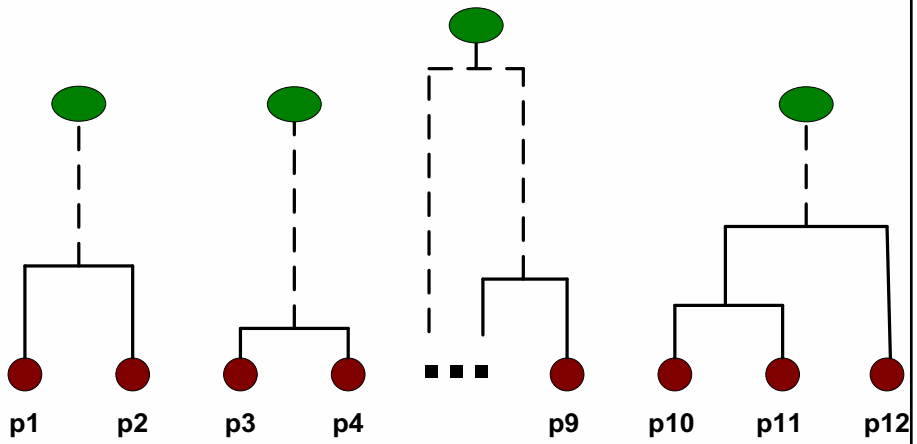
After Merging

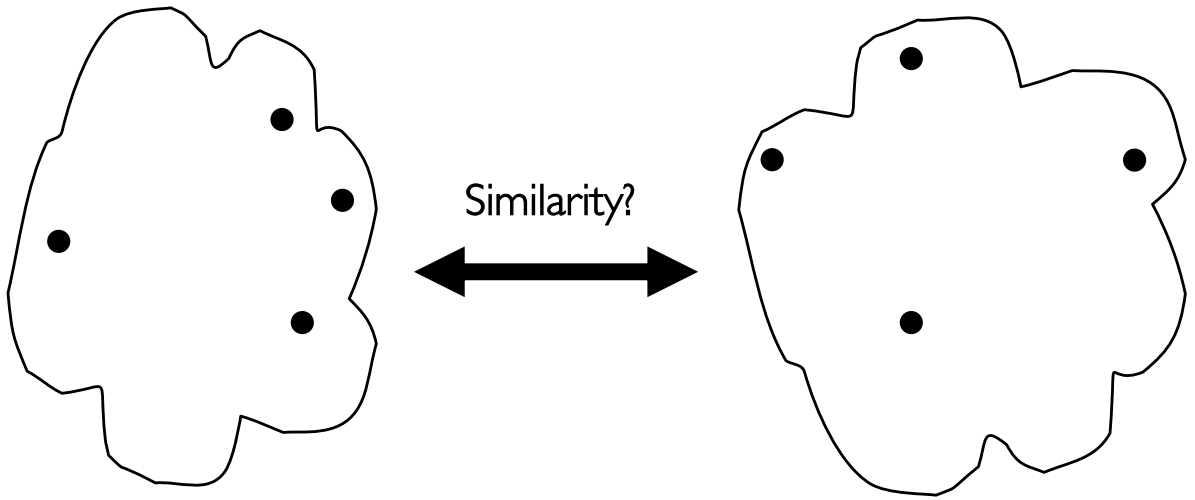
- The question is "How do we update the proximity matrix?"

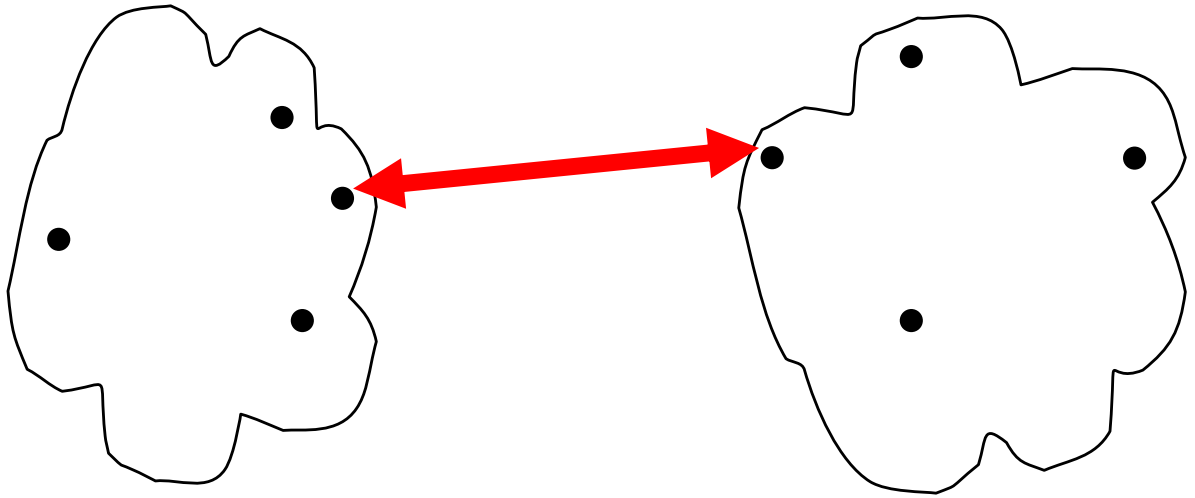


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

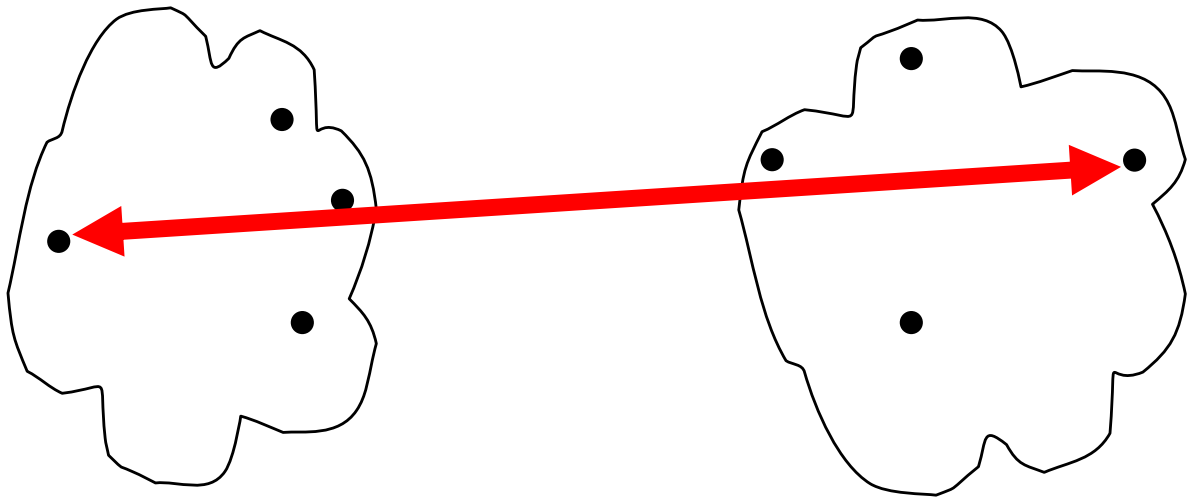
Distance Matrix



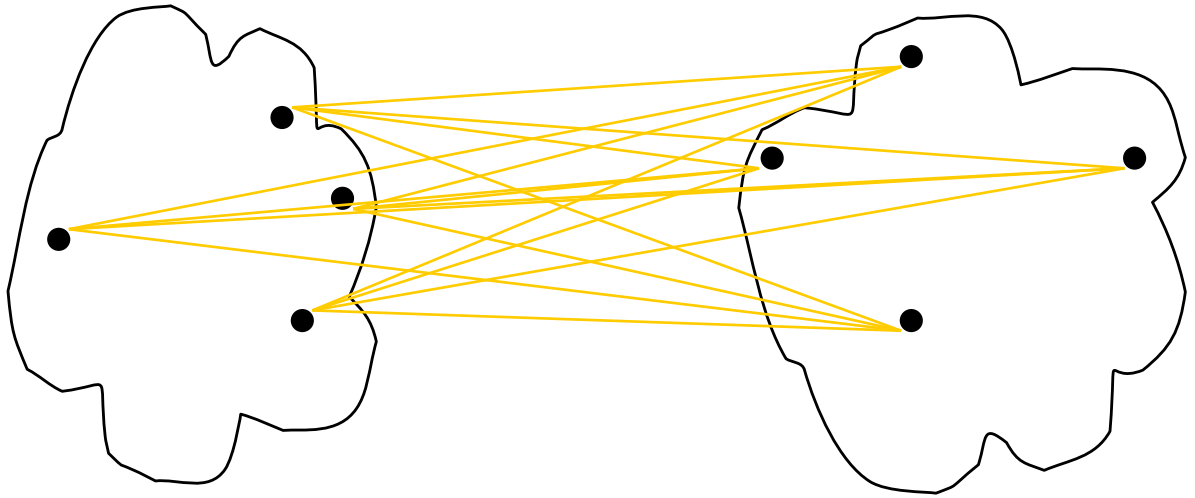

Similarity?



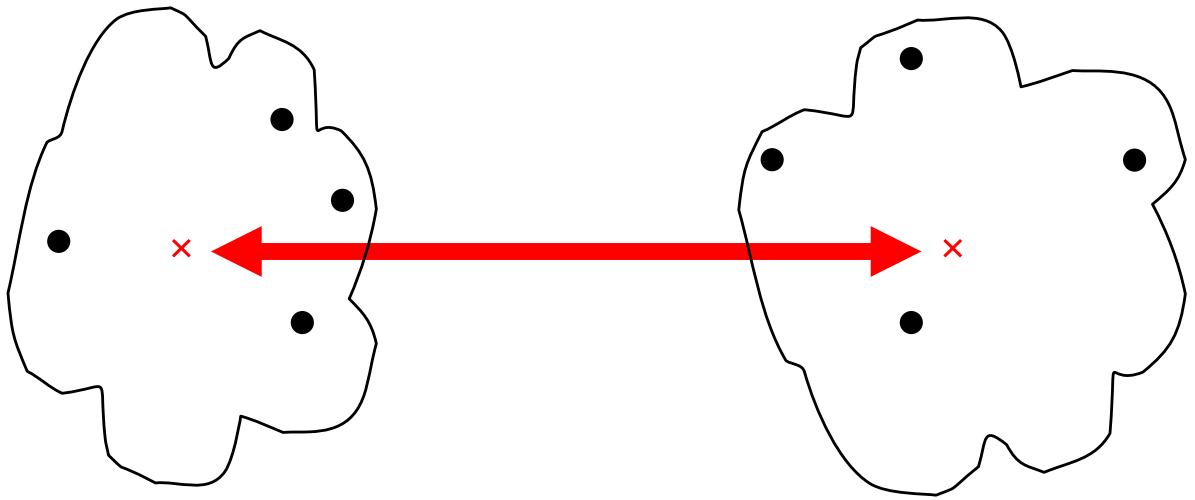
Single Linkage or MIN



Complete Linkage or MAX



Average or Group Average



Distance between Centroids

Typical Alternatives to Calculate the Distance Between Clusters

- Single link (or MIN)
  - smallest distance between an element in one cluster and an element in the other, i.e., $d(Ci, Cj) = \min(t_{i,p}, t_{j,q})$
- Complete link (or MAX)
  - largest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \max(t_{i,p}, t_{j,q})$
- Average (or group average)
  - average distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \text{avg}(d(t_{i,p}, t_{j,q}))$
- Centroid
  - distance between the centroids of two clusters, i.e., $d(C_i, C_j) = d(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)$ where $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_i$ are the centroids
- …

Example

- Suppose we have five items, a, b, c, d, and e.
- We wanto to perform hierarchical clustering on five instances following an agglomerative approach
- First: we compute the distance or similarity matrix
- Dij is the distance between instancce "i" and "j"

$$D = \begin{pmatrix} 0.0 & & & & \\ 2.0 & 0.0 & & & \\ 6.0 & 5.0 & 0.0 & & \\ 10.0 & 9.0 & 4.0 & 0.0 & \\ 9.0 & 8.0 & 5.0 & 3.0 & 0.0 \end{pmatrix}$$

Example

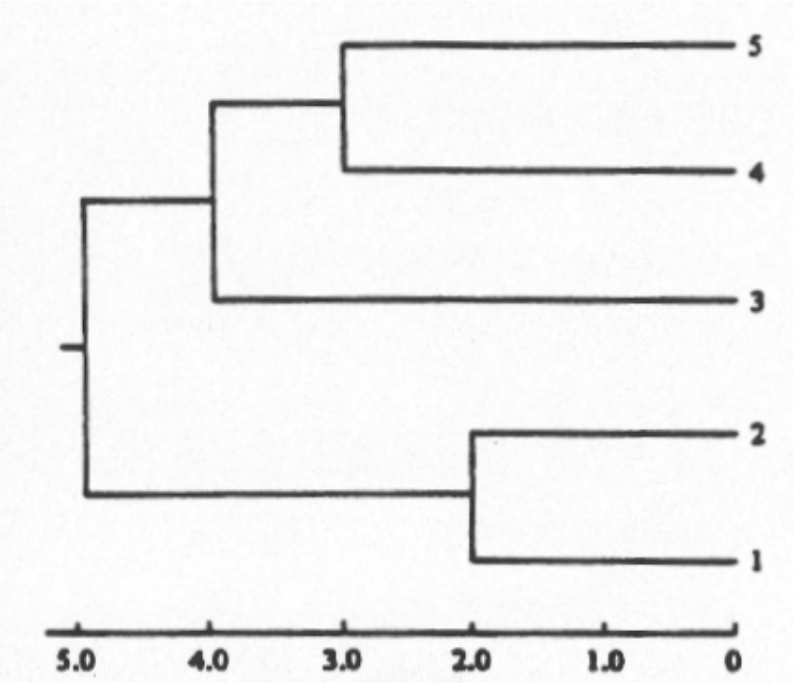
- Group the two instances that are closer
- In this case, a and b are the closest items (D2,1=2)
- Compute again the distance matrix, and start again.
- Suppose we apply single-linkage (MIN), we need to compute the distance between the new cluster {1,2} and the others
  - d(12)3 = min[d13,d23] = d23 = 5.0
  - d(12)4 = min[d14,d24] = d24 = 9.0
  - d(12)5 = min[d15,d25] = d25 = 8.0

Example

- The new distance matrix is,

$$D = \begin{pmatrix} 0.0 & & & \\ 5.0 & 0.0 & & \\ 9.0 & 4.0 & 0.0 & \\ 8.0 & 5.0 & 3.0 & 0.0 \end{pmatrix}$$

- At the end, we obtain the following dendrogram



Determining the Number of Clusters

hierarchical clustering generates
a set of N possible partitions

which one should I choose?

From the previous lecture we know ideally
a good cluster should partition points so that …

Data points in the same cluster should have
a small distance from one another

Data points in different clusters should be at
a large distance from one another.

Within/Between Clusters Sum of Squares (for distance function d)

- Within-cluster sum of squares

$$\mathrm{WSS}(C) = \sum_{i=1}^{k} \sum_{x_j \in C_i} d(x_j, \mu_i)^2$$

where μ_i is the centroid of cluster C_i (in case of Euclidean spaces)

- Between-cluster sum of squares

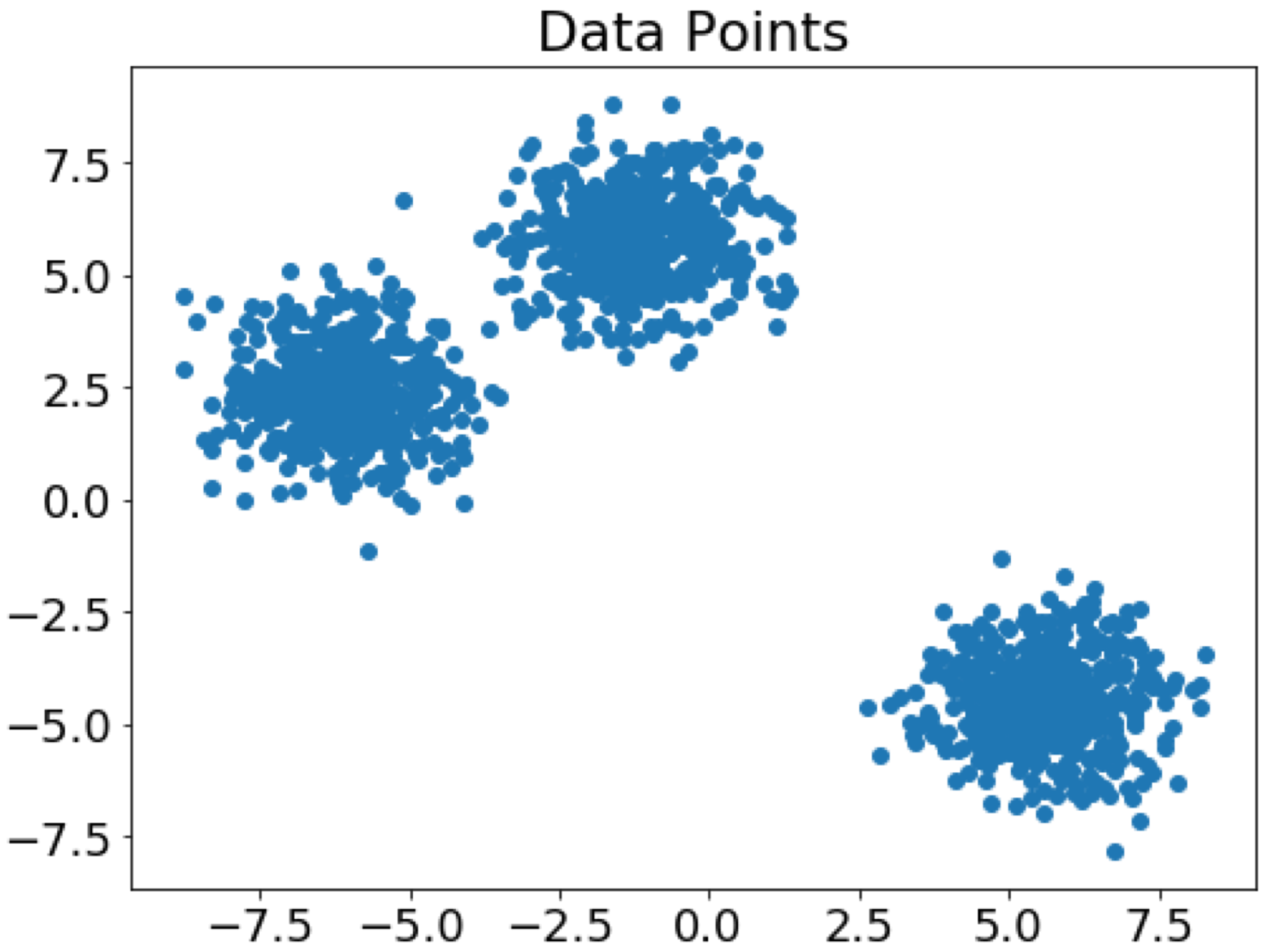
$$\mathrm{BSS}(C) = \sum_{i=1}^{k} |C_i| d(\mu, \mu_i)^2$$

where μ is the centroid of the whole dataset

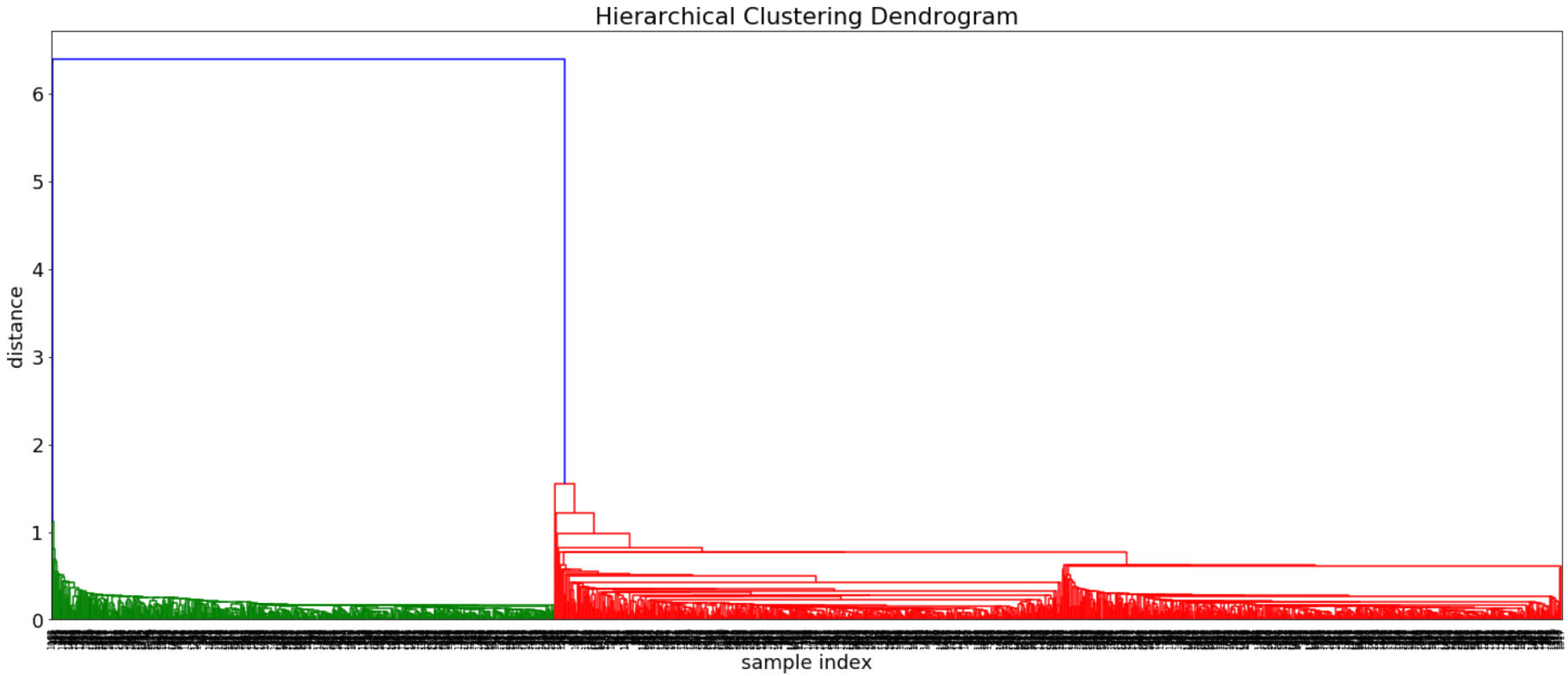
Evaluation of Hierarchical Clustering using Knee/Elbow Analysis

plot the WSS and BSS for every clustering and look for a knee in the plot that show a significant modification in the evaluation metrics

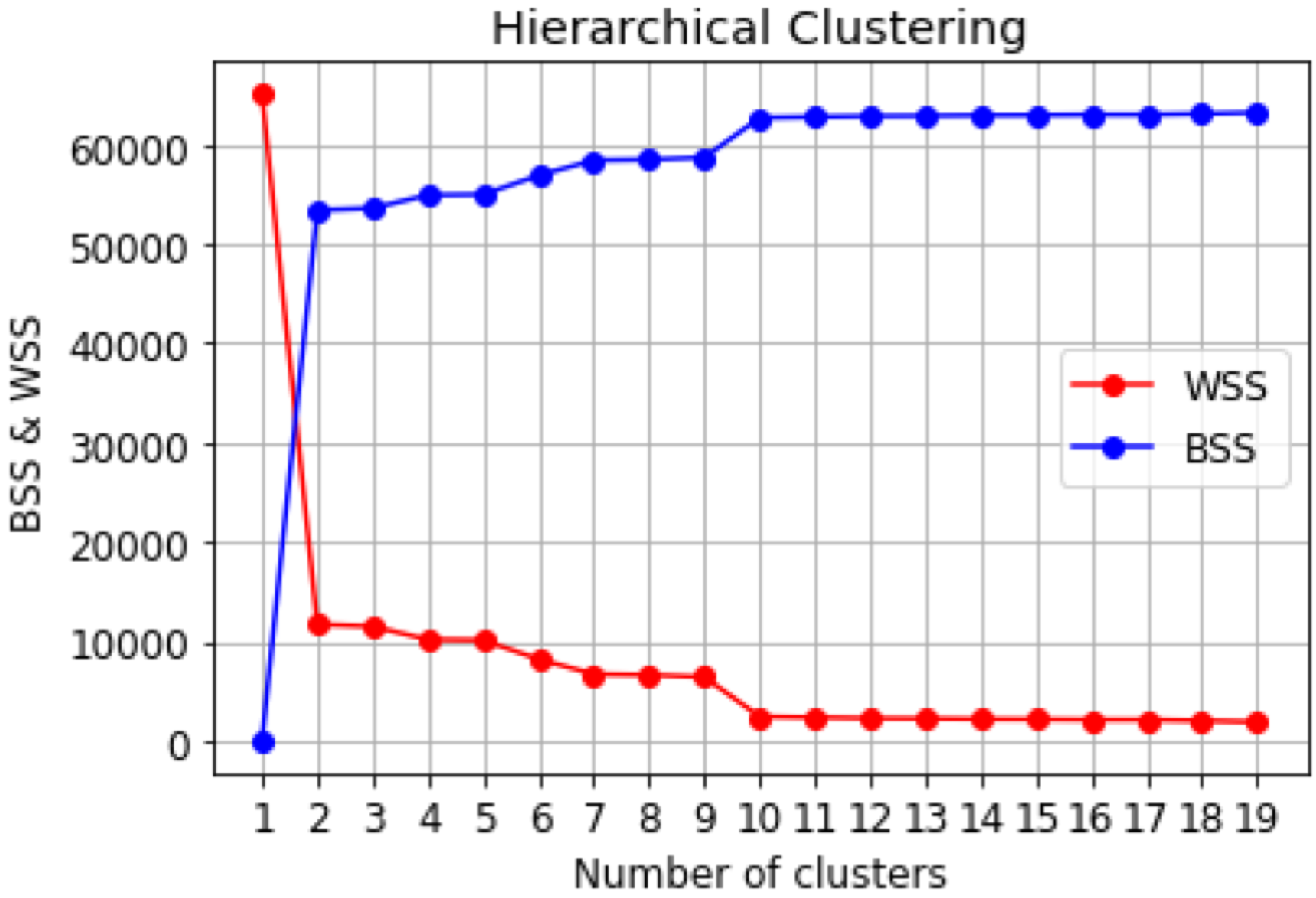
Run the Python notebook for hierarchical clustering



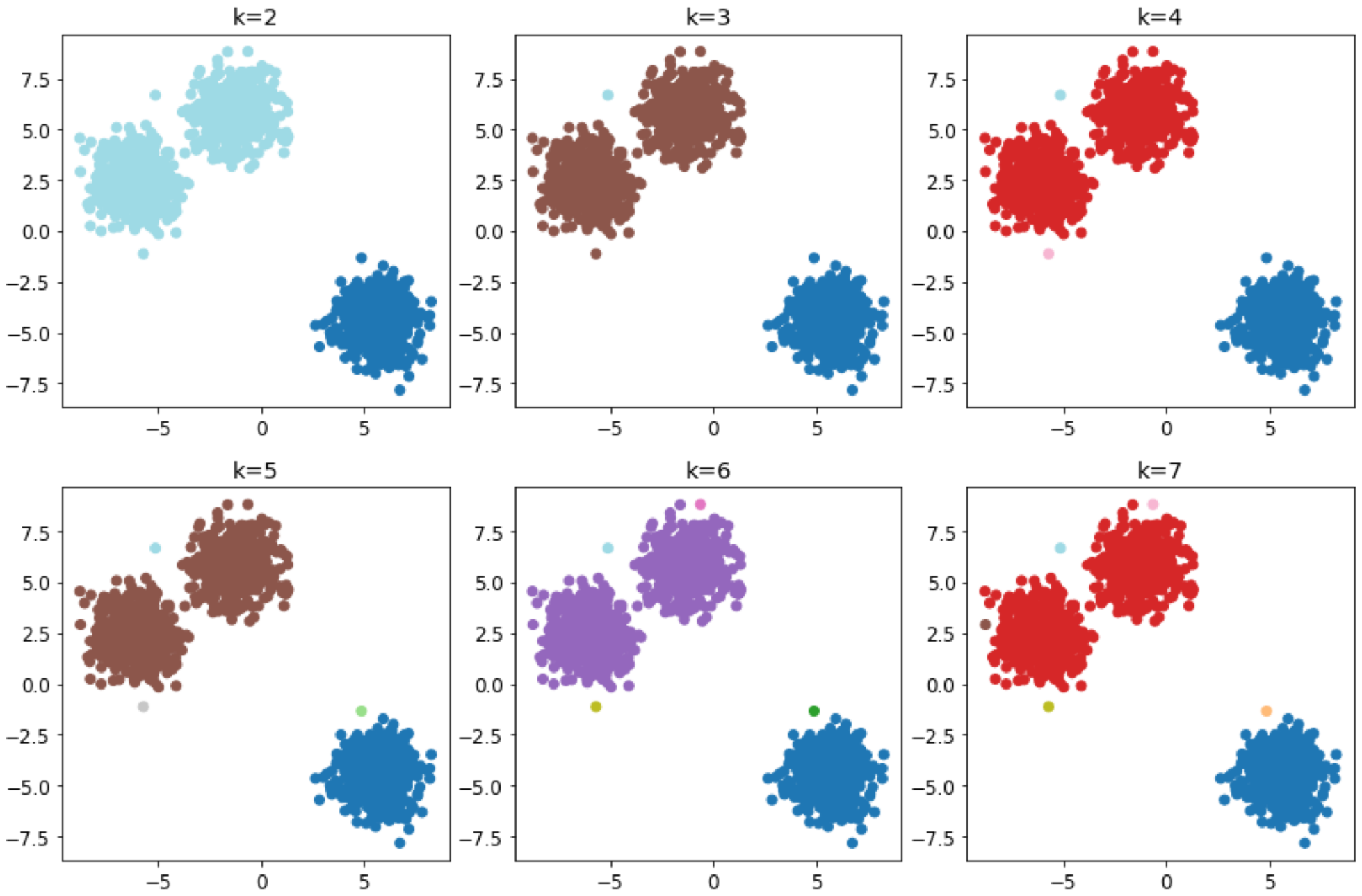
Example data generated using the make_blob function of Scikit-Learn



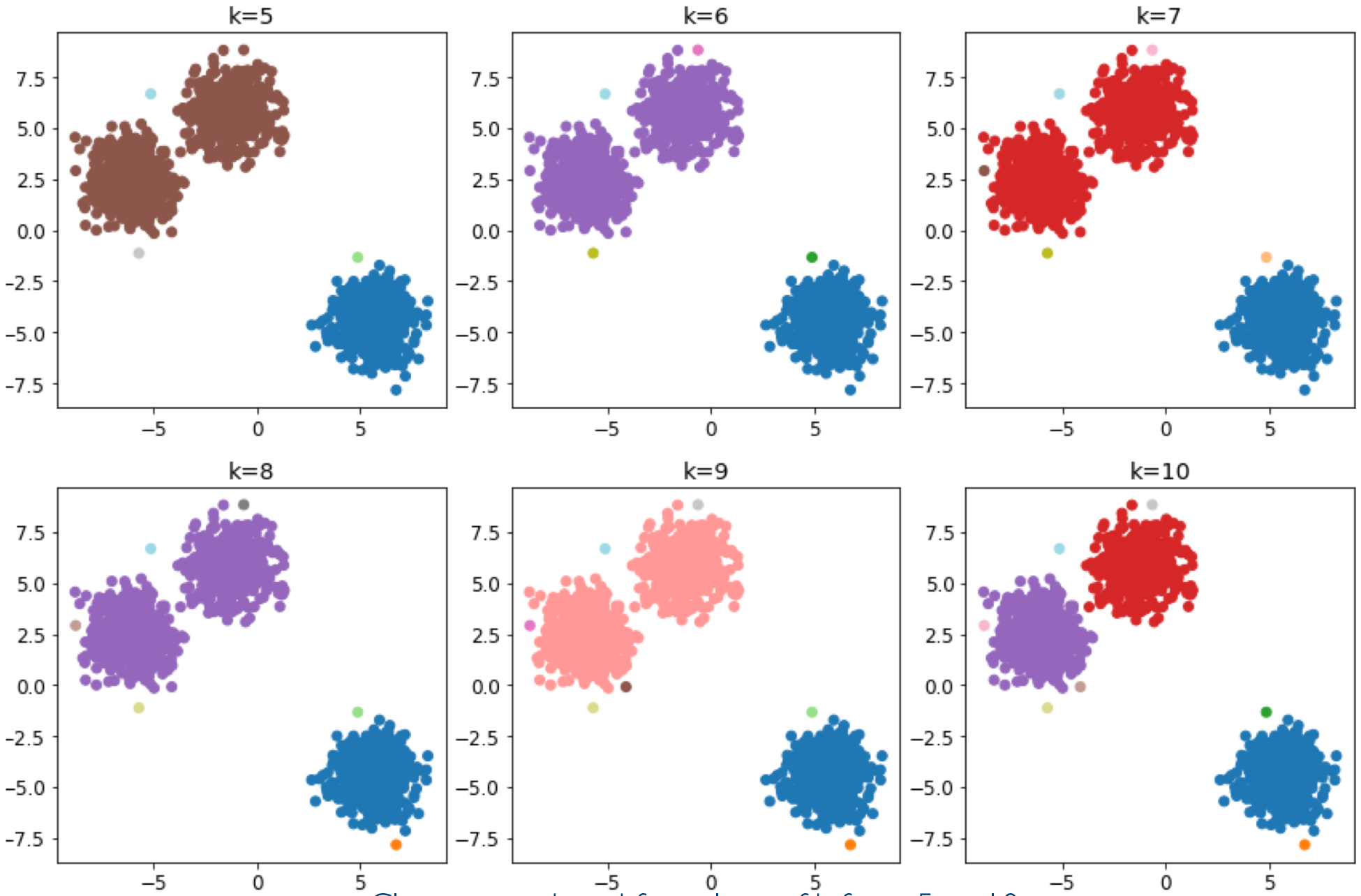
Dendrogram computed using single linkage.



BSS and WSS for values of k from 1 until 19.



Clusters produced for values of k from 2 to 7.



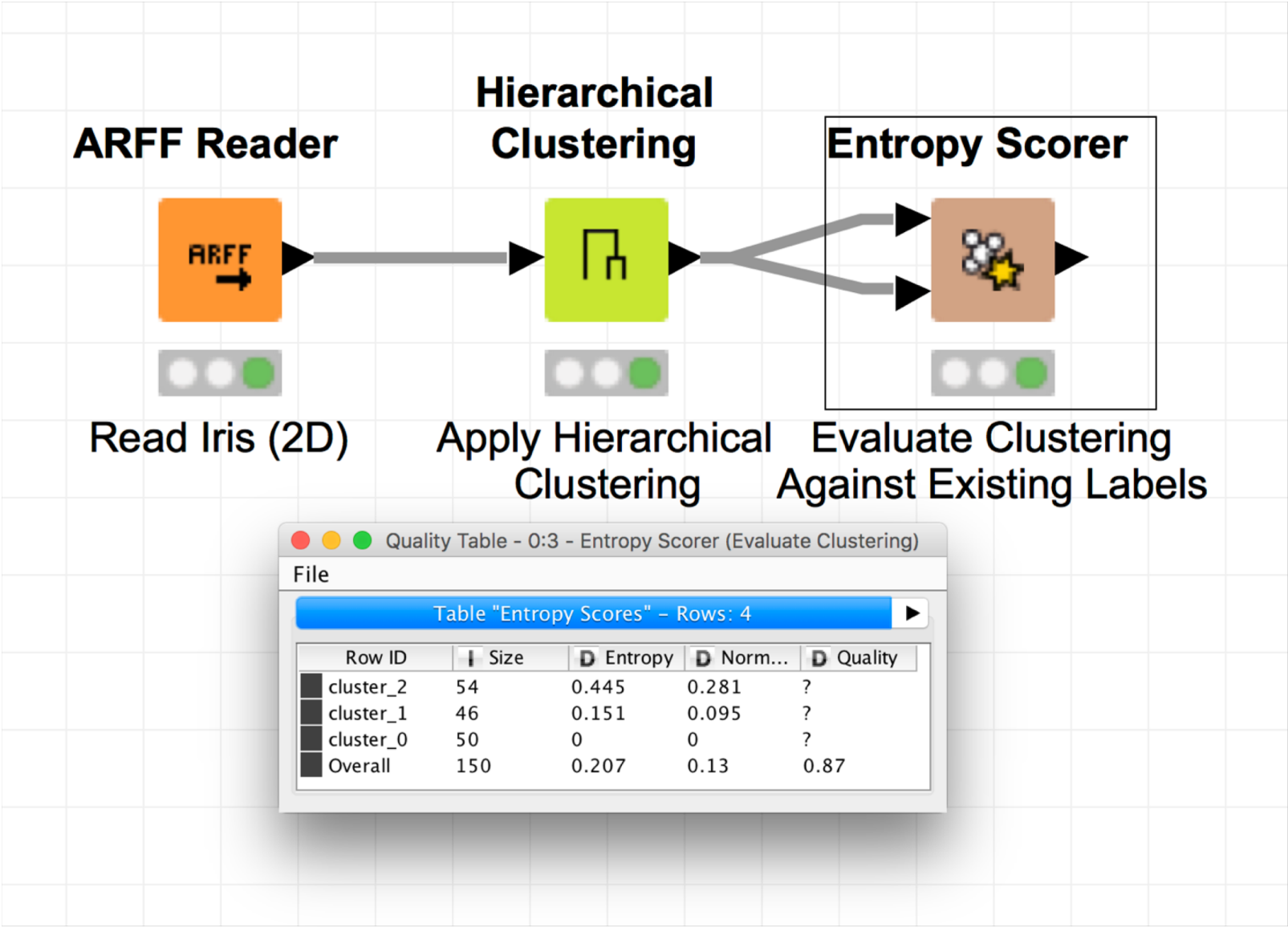
Clusters produced for values of k from 5 to 10.

How can we represent clusters?

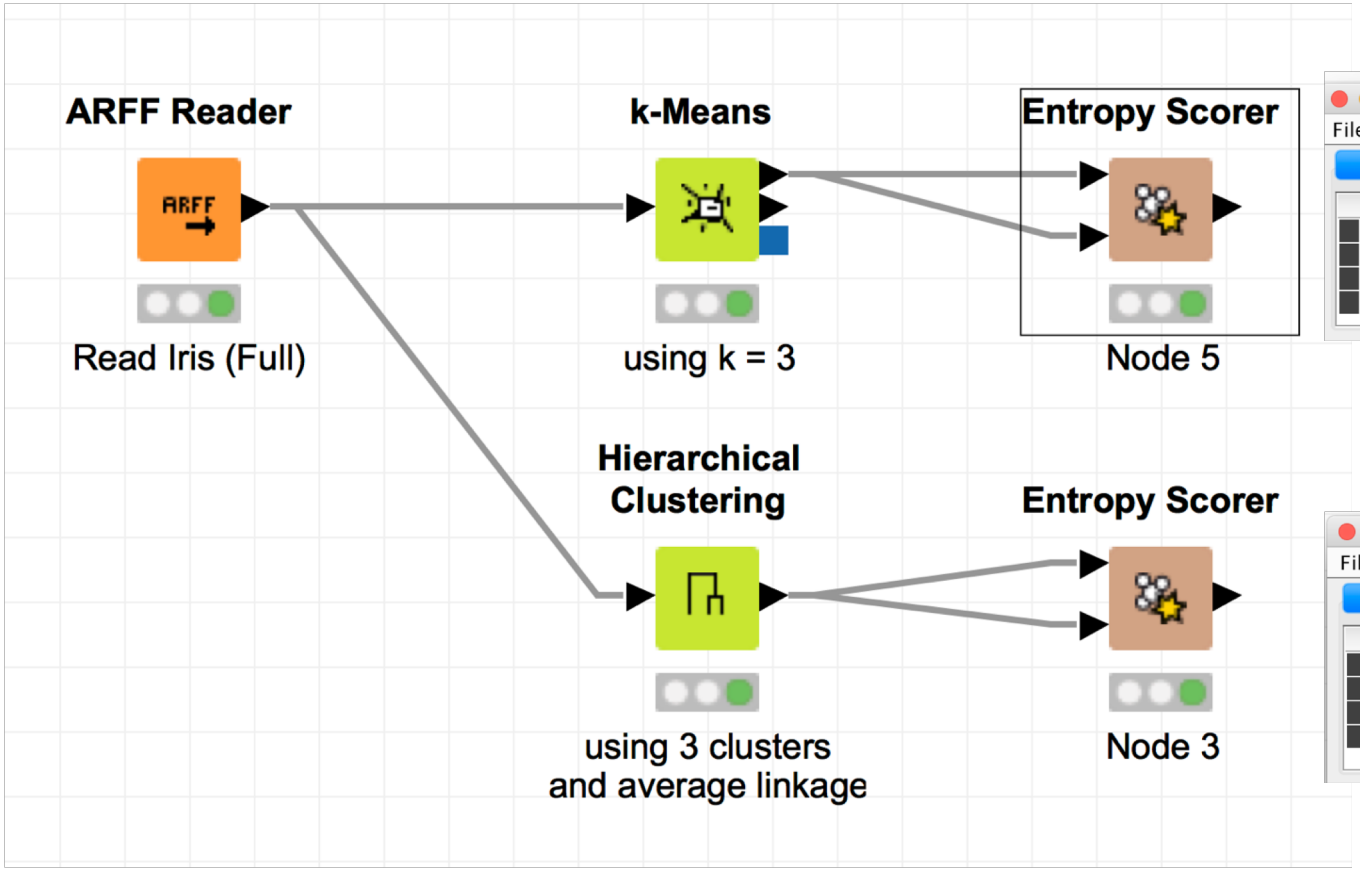
Euclidean vs Non-Euclidean Spaces

- Euclidean Spaces
  - We can identify a cluster using for instance its centroid (e.g. computed as the average among all its data points)
  - Alternatively, we can use its convex hull
- Non-Euclidean Spaces
  - We can define a distance (jaccard, cosine, edit)
  - We cannot compute a centroid and we can introduce the concept of clustroid
- Clustroid
  - An existing data point that we take as a cluster representative
  - It can be the point that minimizes the sum of the distances to the other points in the cluster
  - Or, the one minimizing the maximum distance to another point
  - Or, the sum of the squares of the distances to the other points in the cluster

Examples using KNIME



Evaluation of the result from hierarchical clustering with 3 clusters and average linkage against existing labels



Quality Table - 2:5 - Entropy Scorer

File

Table "Entropy Scores" – Rows: 4

Row ID	Size	Entropy	Norm...	Quality
cluster_2	50	0	0	?
cluster_0	39	0.391	0.247	?
cluster_1	61	0.777	0.49	?
Overall	150	0.418	0.264	0.736

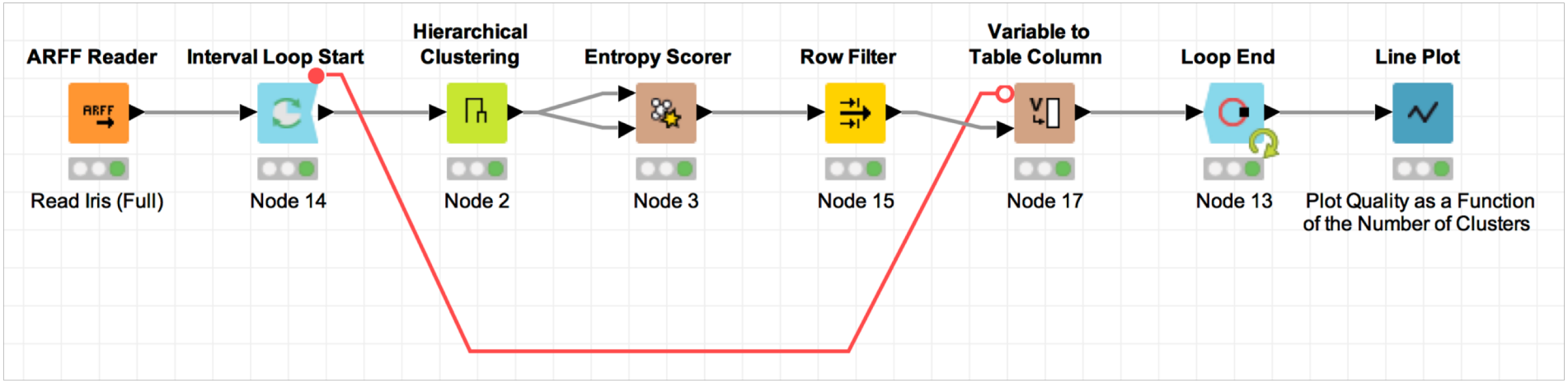
Quality Table - 2:3 - Entropy Scorer

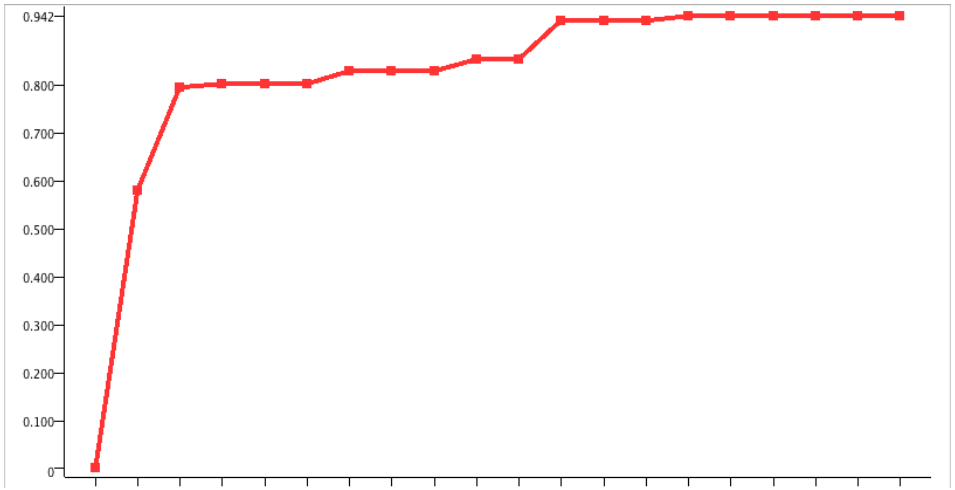
File

Table "Entropy Scores" – Rows: 4

Row ID	Size	Entropy	Norm...	Quality
cluster_0	50	0	0	?
cluster_1	36	0	0	?
cluster_2	64	0.758	0.478	?
Overall	150	0.323	0.204	0.796

Comparison of hierarchical clustering with 3 clusters and average linkage against k-Means with k=3





Computing cluster quality from one to 20 clusters using the entropy scorer

Summary

Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters
- Major weakness of agglomerative clustering methods
  - They do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - They can never undo what was done previously

Run the python notebooks and the KNIME workflows for this lecture