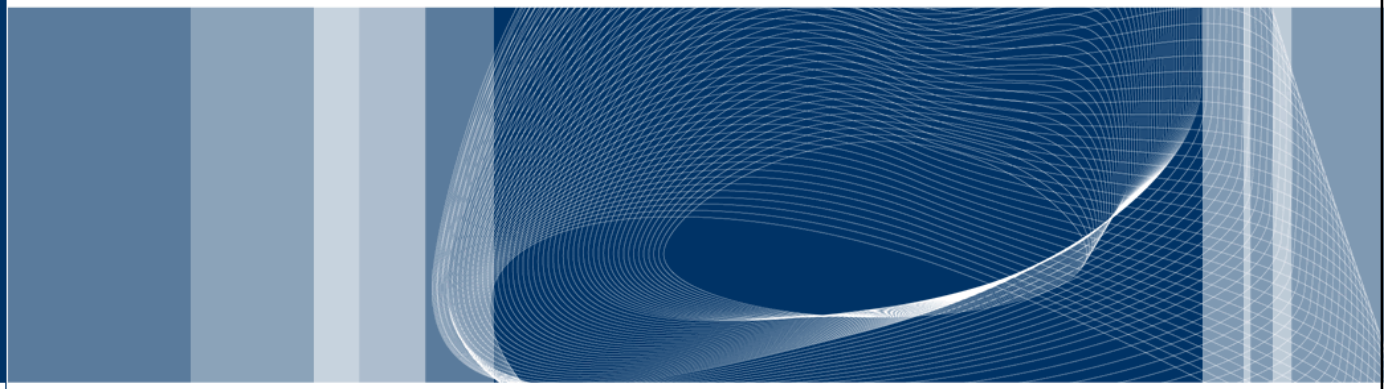
POLITECNICO DI MILANO



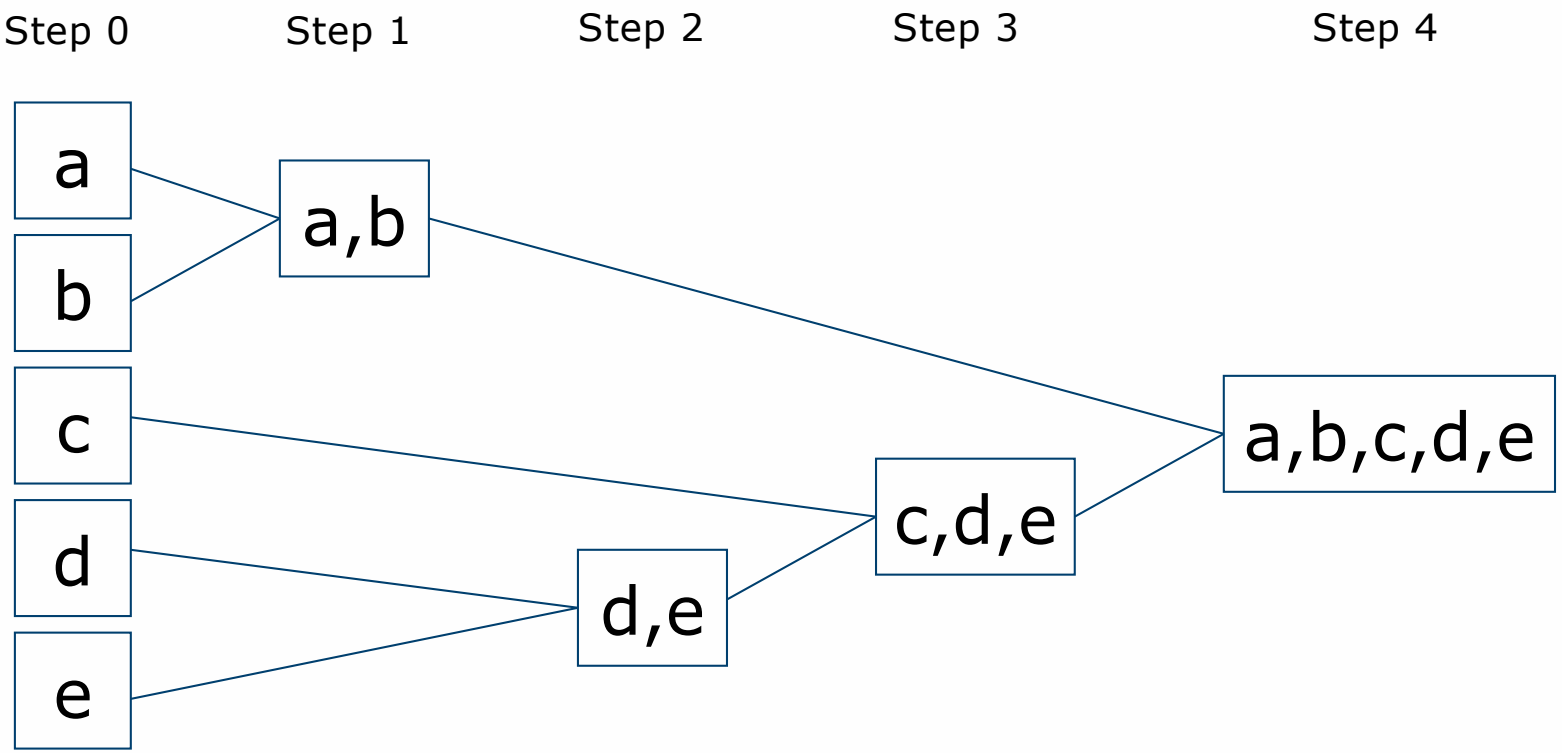
Hierarchical Clustering

Data Mining and Text Mining

Prof. Pierluca Lanzi

Agglomerative 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



Clustering	Partition
C_1	{A} {B} {C} {D} {E}
C_2	{A, B} {C} {D} {E}
C_3	{A, B} {C} {D, E}
C_4	{A, B} {C, D, E}
C_5	{A, B, C, D, E}

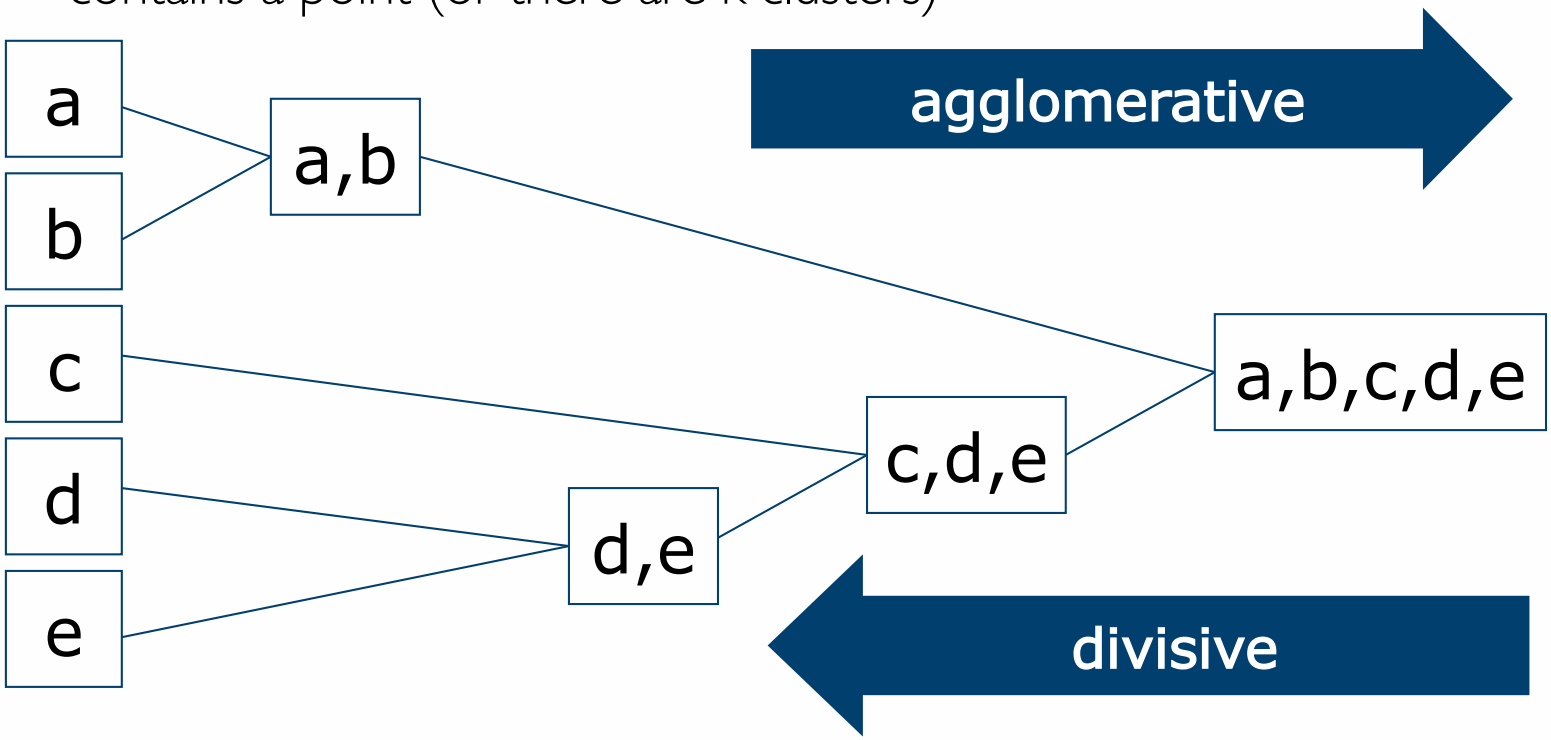
What is Hierarchical Clustering?

By far, one of the most common clustering techniques

Produces a hierarchy (dendrogram) of nested clusters that can be analyzed and visualized

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)

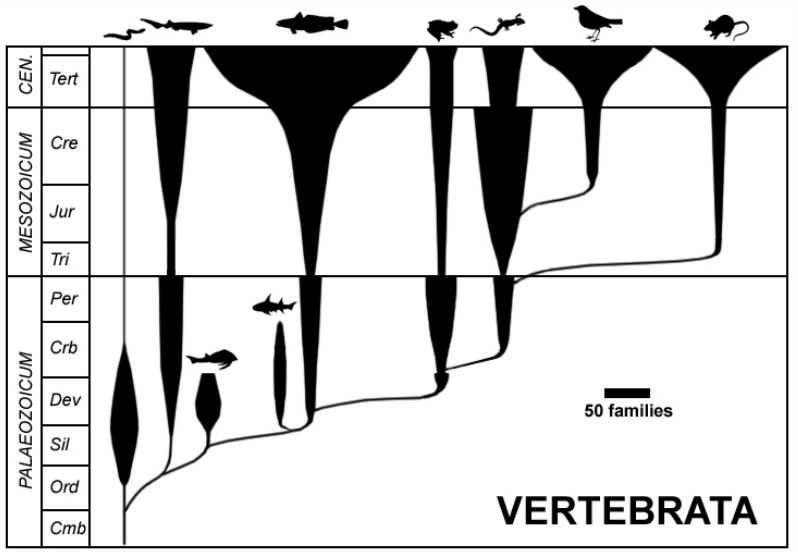


Strengths of Hierarchical Clustering

- 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



Spindle diagram by Petter Bøckman

Agglomerative Clustering Algorithm

1. Compute proximity matrix of pairwise distances between all points
2. Let each data point be a cluster
3. Repeat until a single cluster remains
   - Merge two closest clusters
   - Update proximity matrix

- Most popular hierarchical clustering technique
- Key operation is the computation of the proximity of two clusters
- Different approaches to defining the distance between clusters distinguish the different algorithms

ALGORITHM 14.1. Agglomerative Hierarchical Clustering Algorithm

```
AGGLOMERATIVECLUSTERING(D, k):
1 C ← {C_i = {x_i} | x_i ∈ D} // Each point in separate cluster
2 Δ ← {δ(x_i, x_j): x_i, x_j ∈ D} // Compute distance matrix
3 repeat
4     Find the closest pair of clusters C_i, C_j ∈ C
5     C_ij ← C_i ∪ C_j // Merge the clusters
6     C ← (C \ {C_i, C_j}) ∪ {C_ij} // Update the clustering
7     Update distance matrix Δ to reflect new clustering
8 until |C| = k
```

Hierarchical Clustering: Time and Space Requirements

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

- There are N steps and at each step the size, N^2, proximity matrix must be updated and searched. $O(N^3)$ time in many cases

- Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

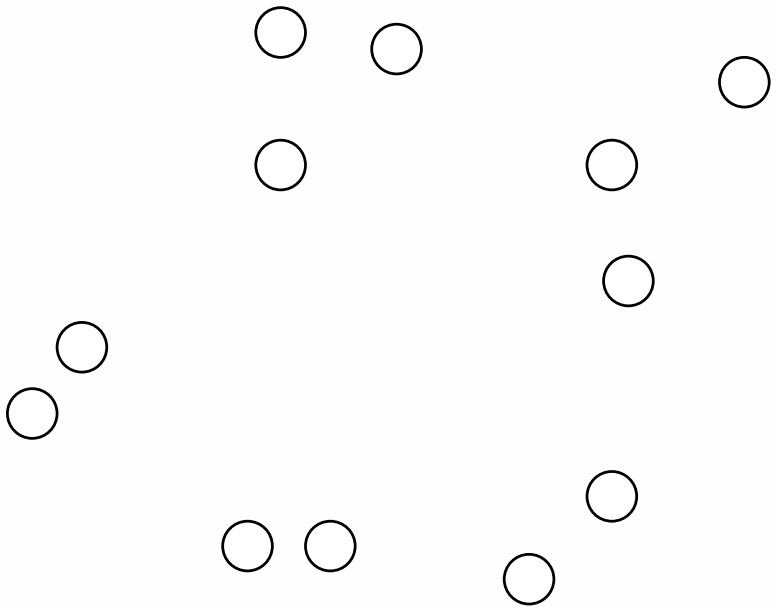
Efficient Implementation

- 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(N log 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

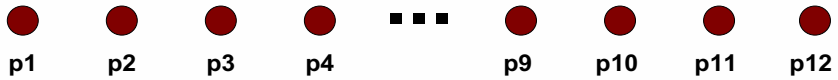
Initial Configuration

- Start with clusters of individual points and the distance matrix



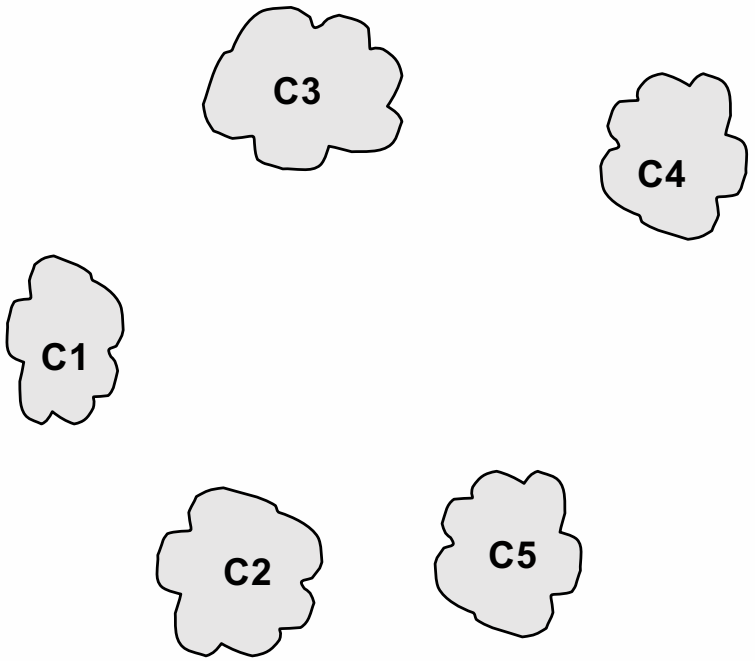
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Distance Matrix



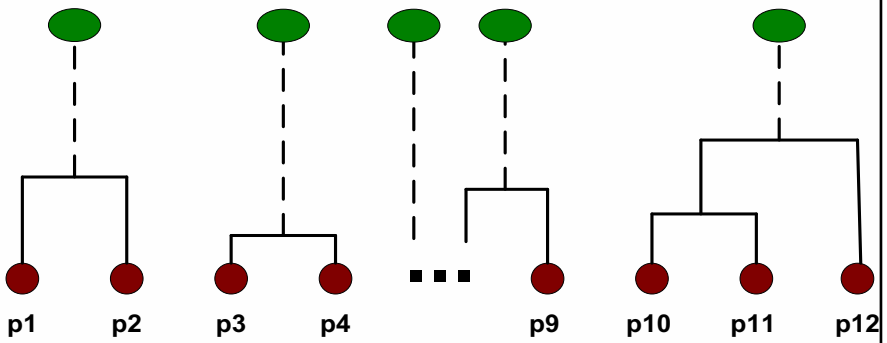
Intermediate Situation

- After some merging steps, we have some clusters



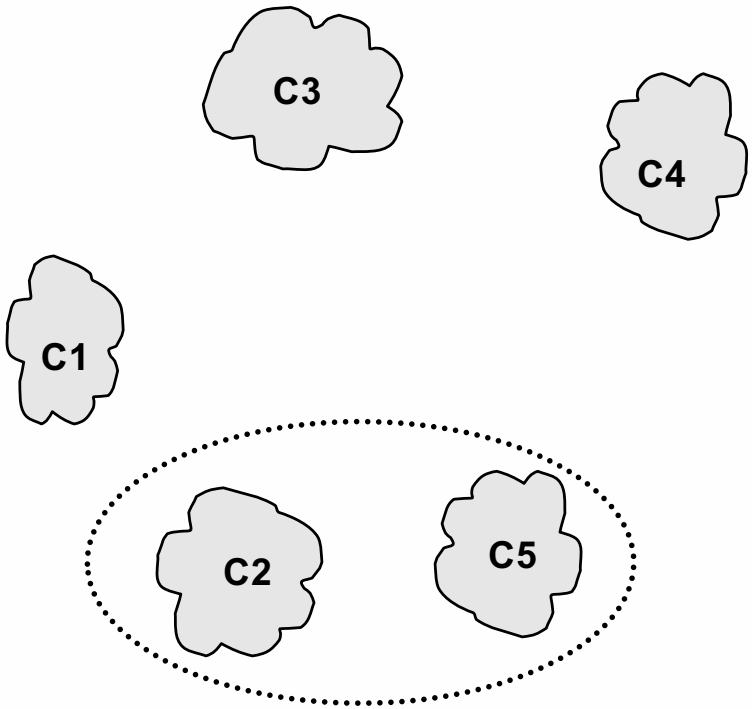
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance Matrix



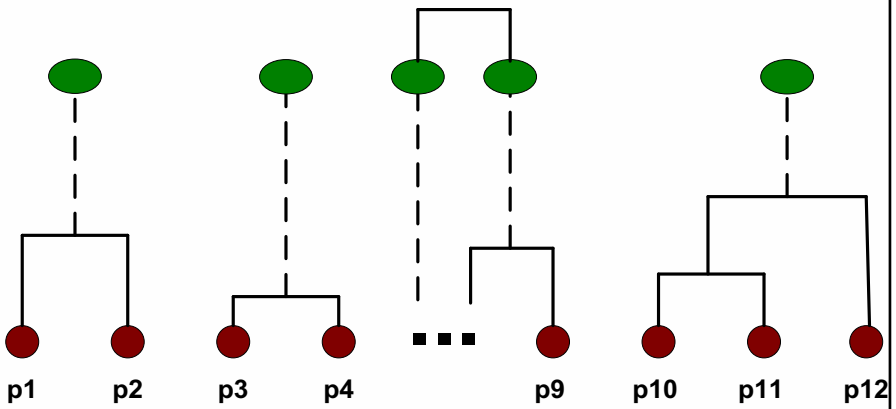
Intermediate Situation

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



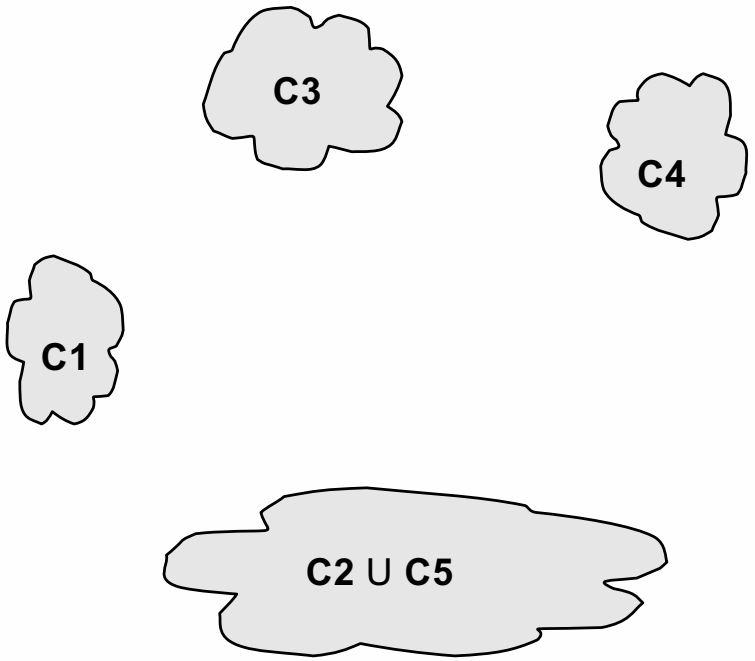
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance Matrix



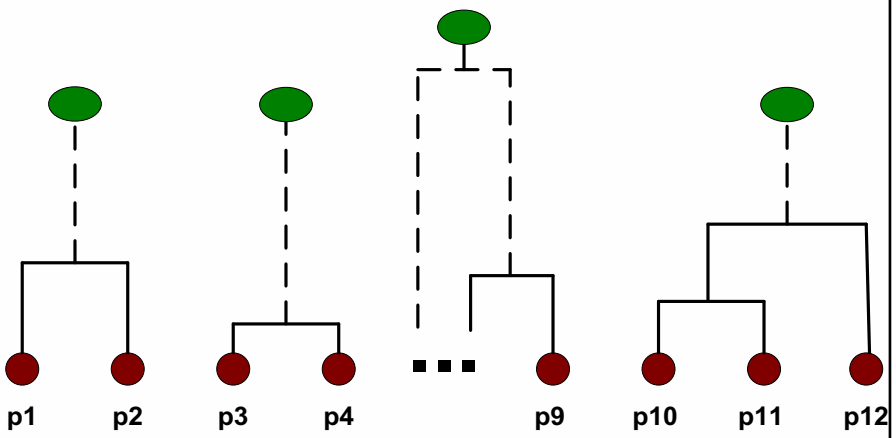
After Merging

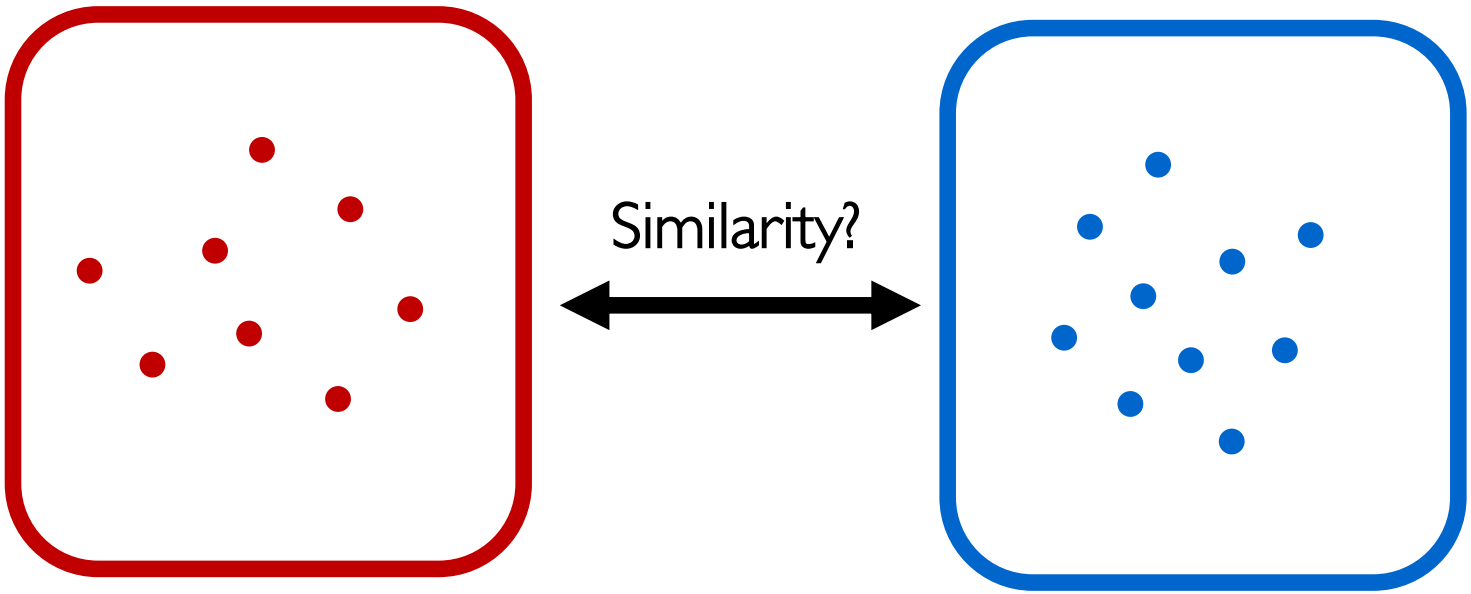
- The question is "How do we update the proximity matrix?"

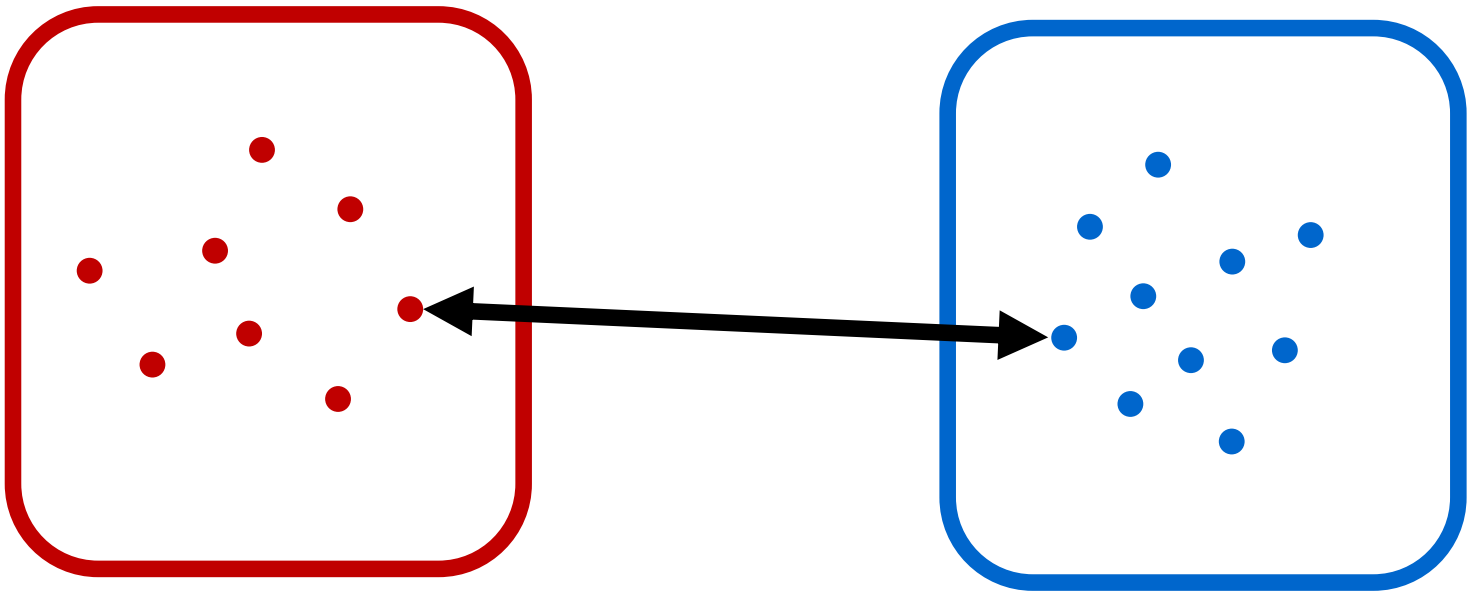


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

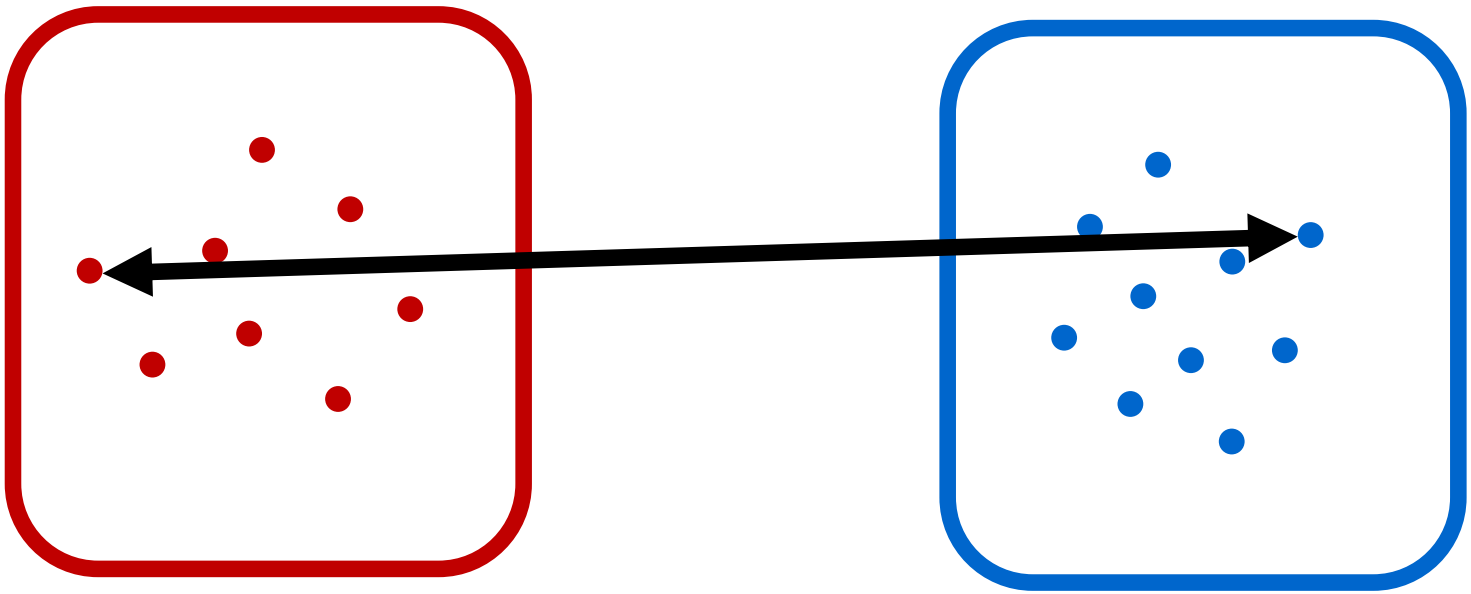
Distance Matrix



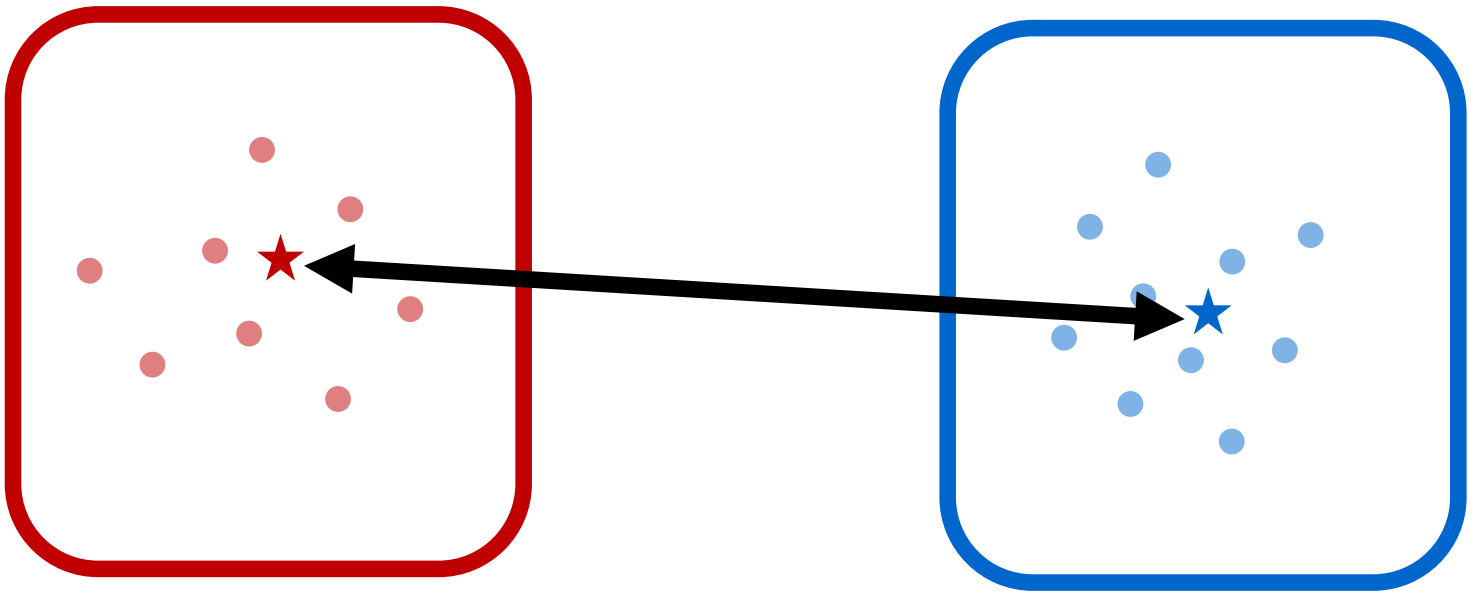

Similarity?



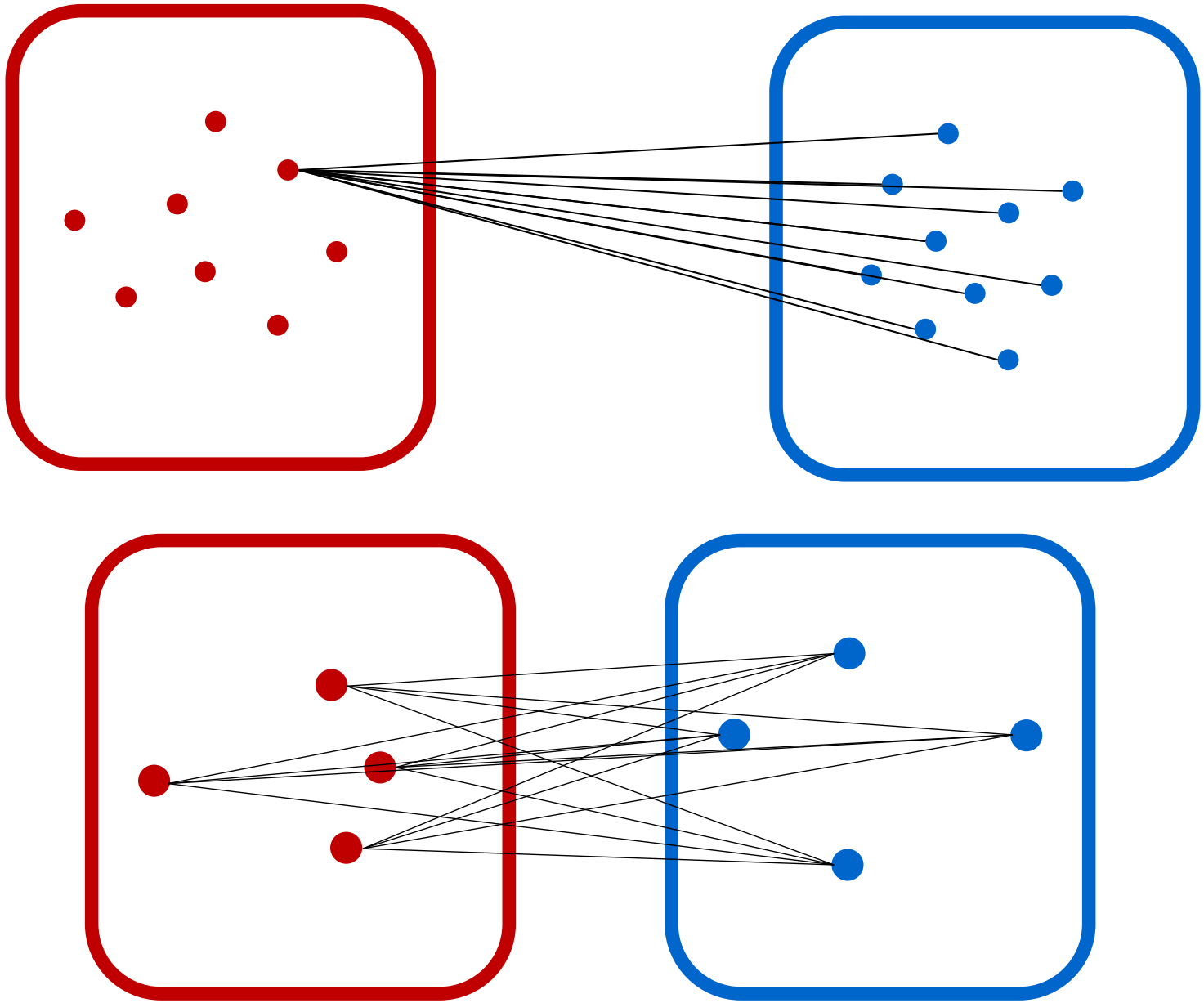
Single Linkage (MIN)



Complete Linkage (MAX)



Mean or Centroid Distance



Group Average

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., $d(C_i, C_j) = \max(t_{i,p}, t_{j,q})$
- Mean Distance
  - 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 cluster means (or centroids)
- Group average
  - average distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \text{avg}(d(t_{i,p}, t_{j,q}))$
- …

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

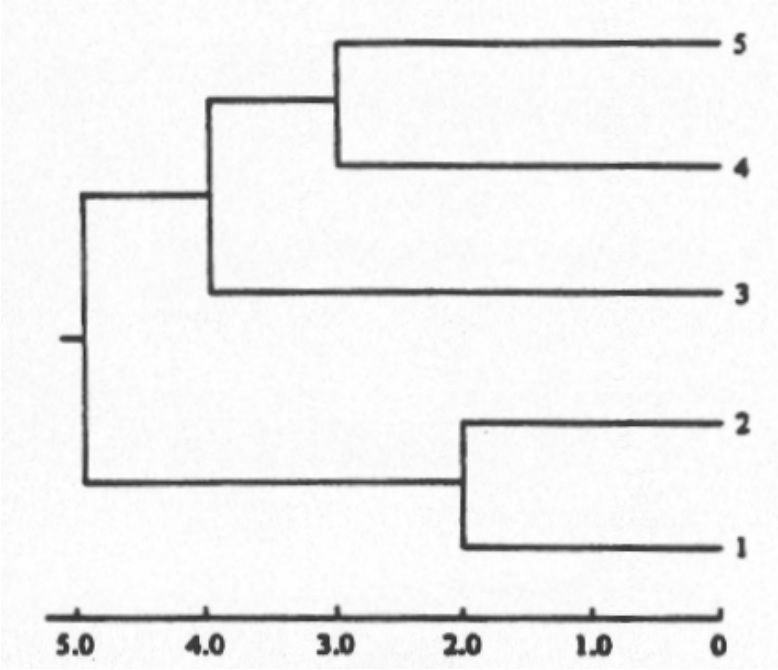
- Group the two instances that are closer
- In this case, a and b are the closest items (D2,1=2)
- Compute again the distance matrix, and start again.
- Suppose we apply single-linkage (MIN), we need to compute the distance between the new cluster {1,2} and the others
  - d(12)3 = min[d13,d23] = d23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

Ideally a good clustering should
data 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.

How can we evaluate
the quality of a clustering solution?

Clustering Quality Measures

- Internal Validation Measures
  - Employ criteria that are derived from the data itself
  - For instance, intracluster and intercluster distances to measure cluster cohesion and separation
  - Cohesion evaluates how similar are the points in the same cluster
  - Separation, how far apart are the points in different clusters
- External Validation Measures
  - Use prior or expert-specified knowledge about the clusters
  - For example, we cluster the Iris data using the four input variables, then we evaluate the clustering using known class labels
  - Employs criteria that are not inherent to the dataset
- Relative Validation Measures
  - Aim to directly compare different solutions, usually those obtained via different parameter settings for the same algorithm.

Internal Measures: Cohesion and Separation

- Cohesion measures how closely related are objects in a cluster
  - Within-cluster sum of squares

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

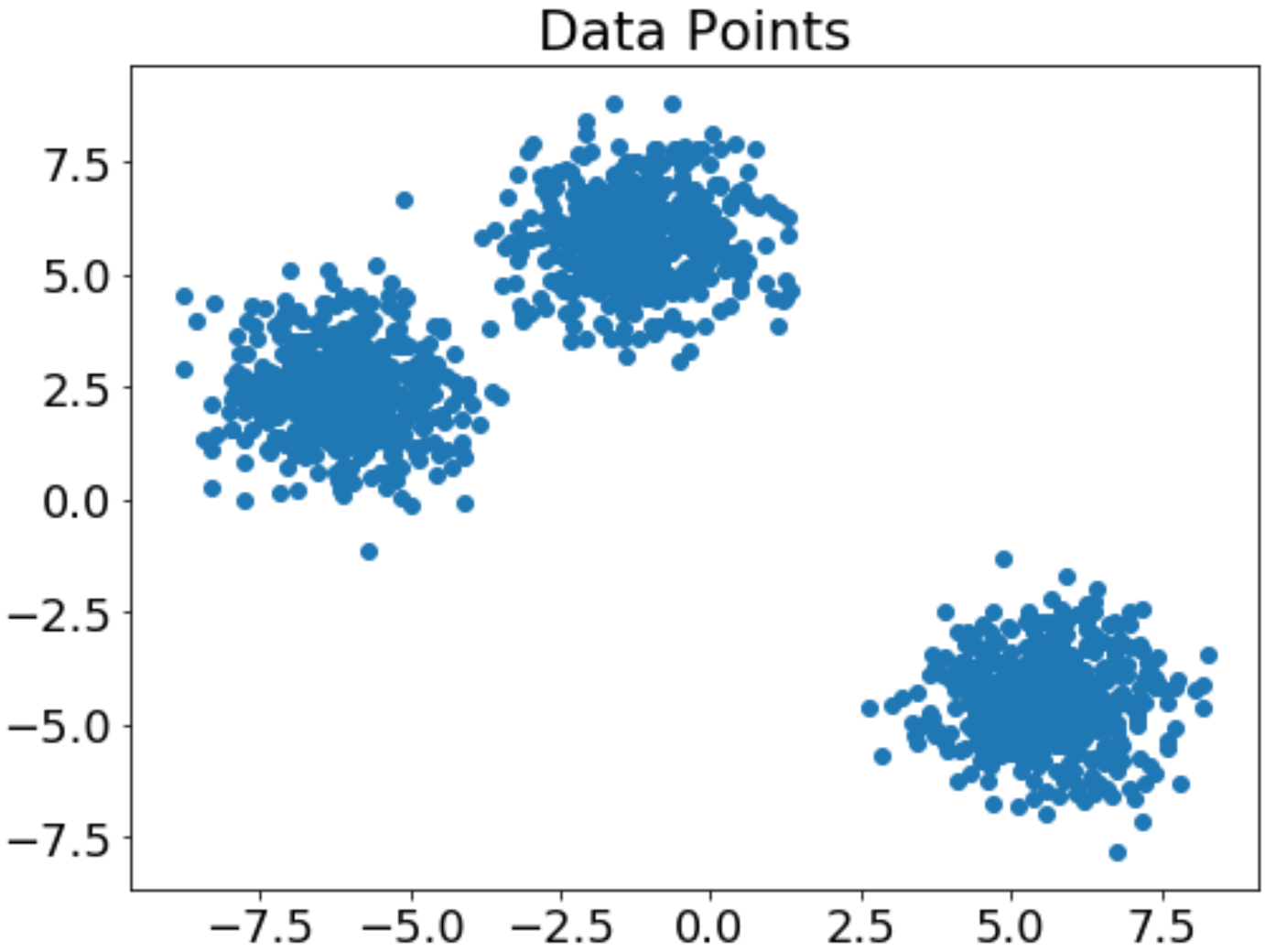
  - where μ_i is the centroid of cluster C_i (in case of Euclidean spaces)
- Separation measures wells separated a cluster is from other clusters
  - Between-cluster sum of squares

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

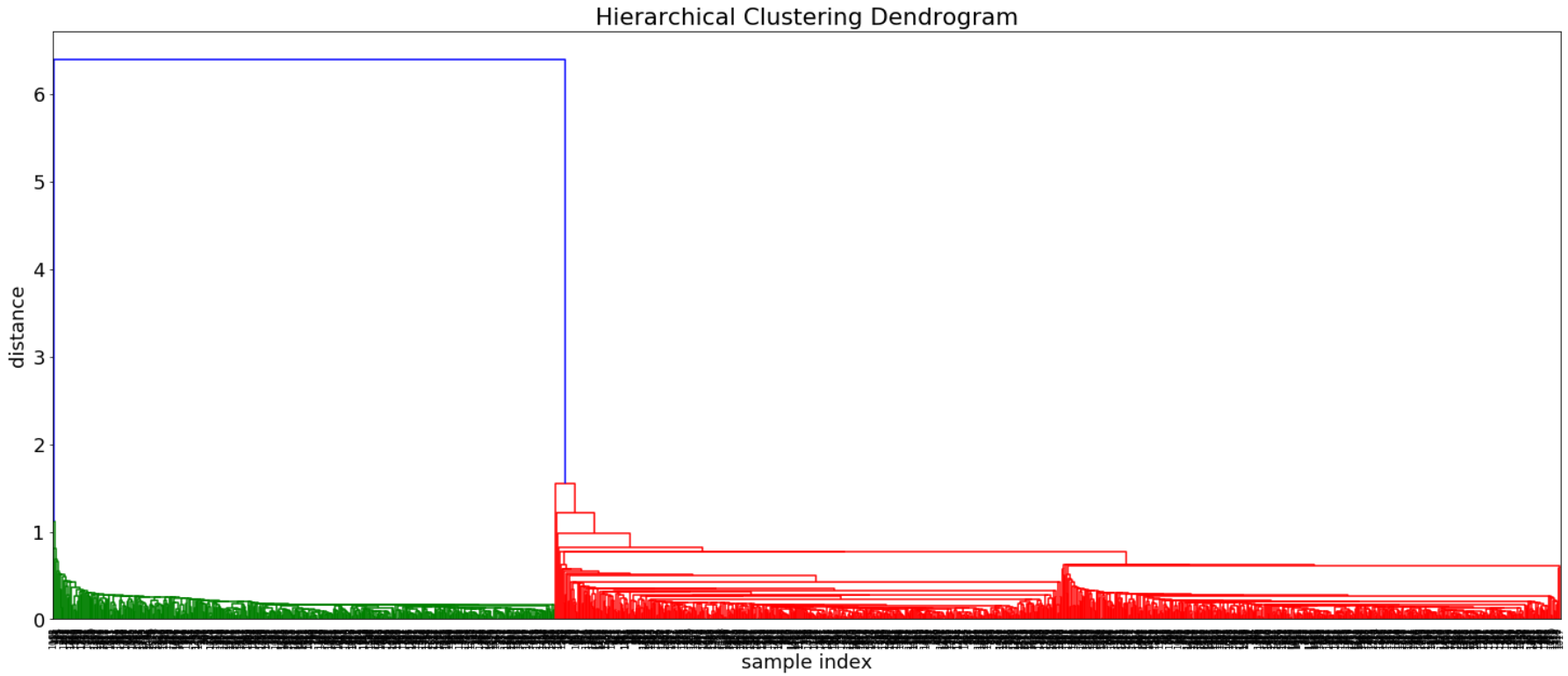
where μ is the centroid of the whole dataset

Evaluation of Hierarchical Clustering using Knee/Elbow Analysis

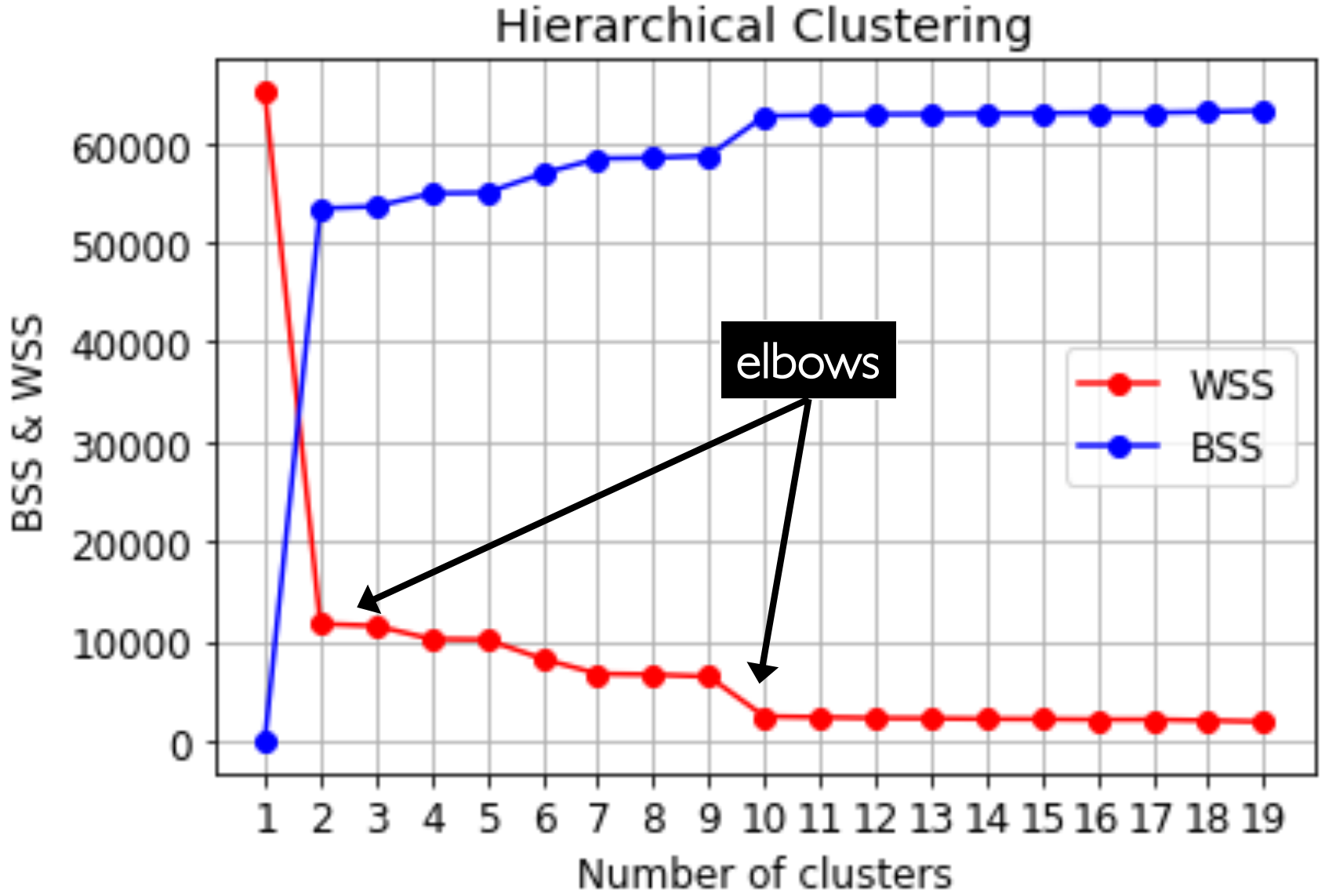
plot the WSS and BSS for every clustering and look for a knee in the plot that show a significant modification in the evaluation metrics



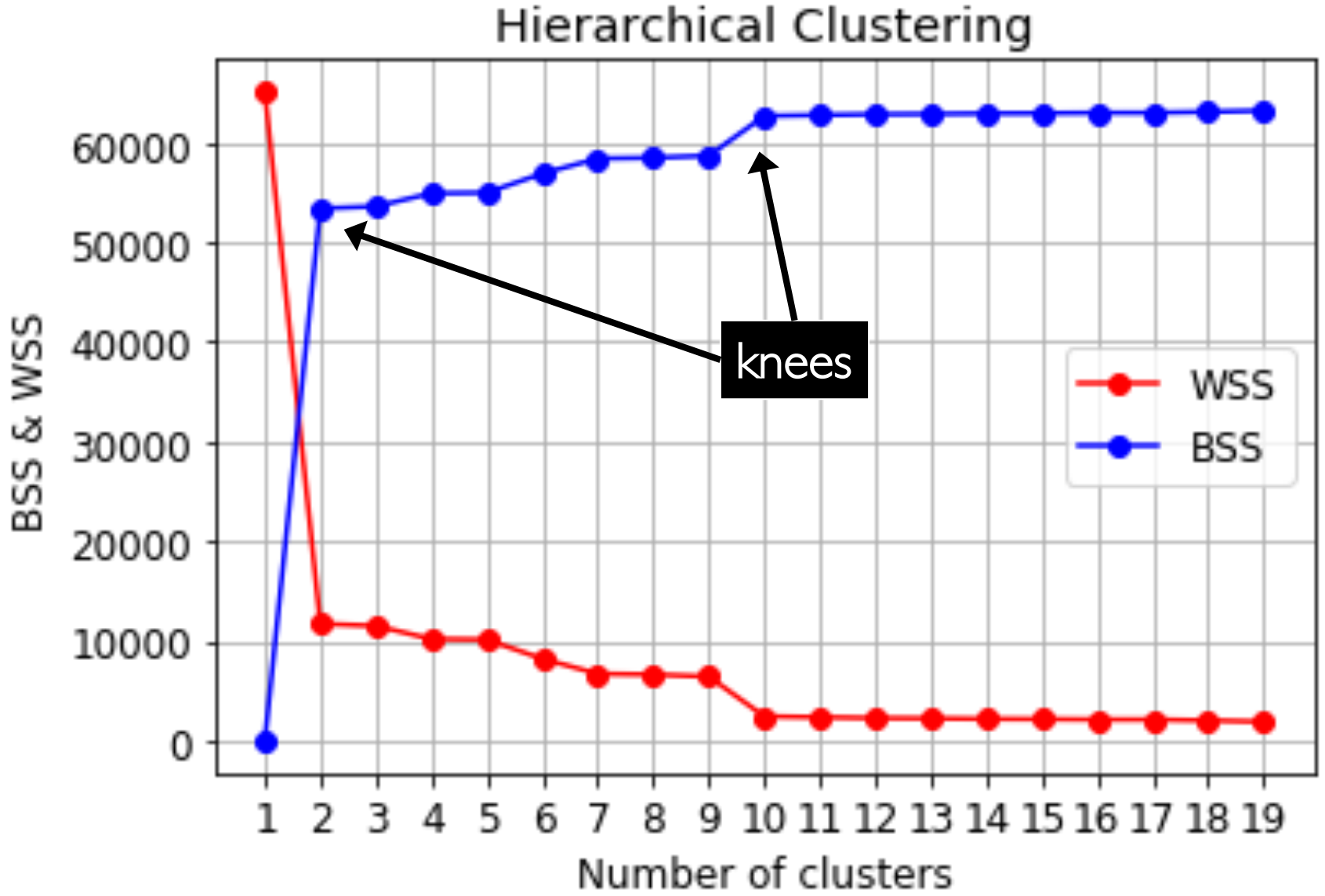
Example data generated using the make_blob function of Scikit-Learn



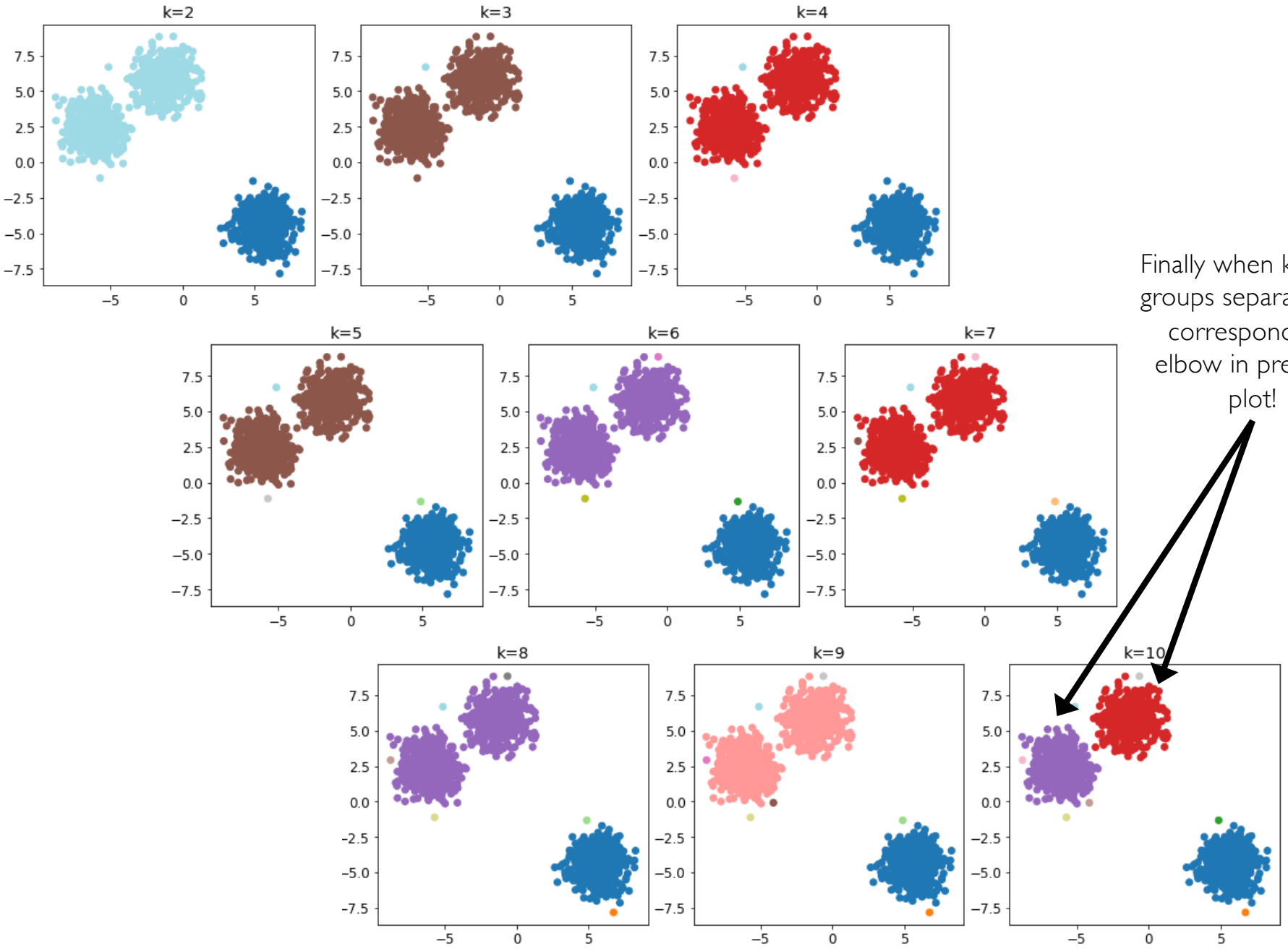
Dendrogram computed using single linkage.



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



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



Clusters produced for #clusters (k) ranging from 2 to 10

Knee/elbow plots might
not tell the whole story!

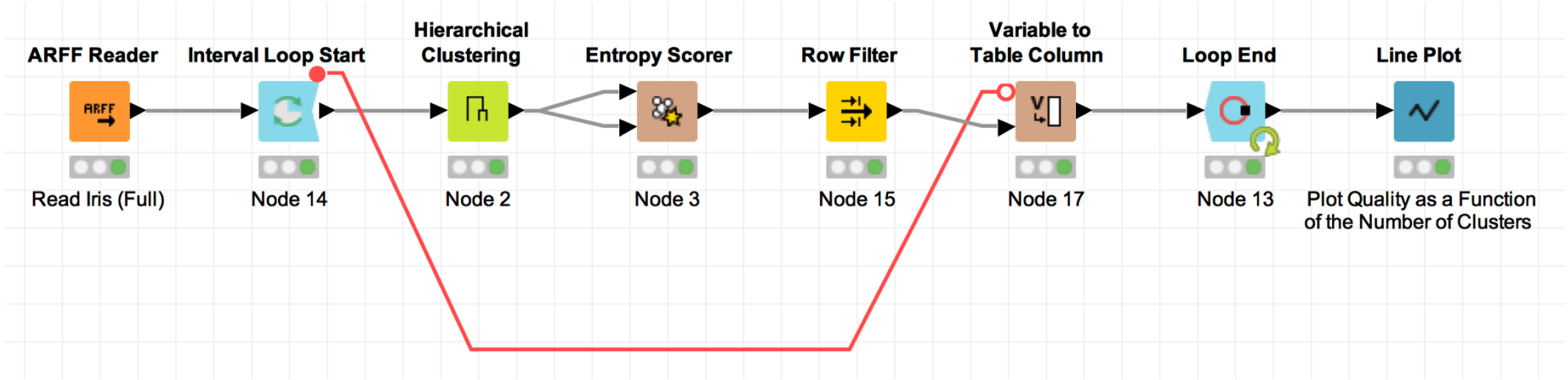
In the example, the plots suggested
several choices of clustering

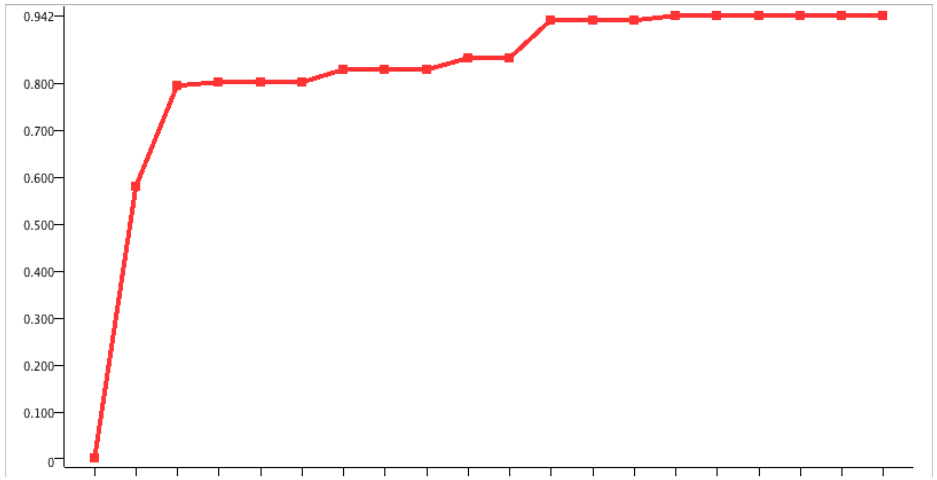
We should use the elbow as a guide but test
multiple solutions around that value

We should analyze different solutions around the knees/elbow

We should also analyze the merges to eliminate the ones due to noise

Examples using KNIME





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

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)
  - If we cannot compute a centroid, we can introduce medoid
- Medoid
  - Existing data point that we take as a cluster representative
  - Point that minimizes the sum of distances to all other points in cluster
- Alternatives to Medoid
  - Choose a point that minimize the maximum distance to another point or the sum of the squares of the distances to the other points in the cluster

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

Study Material

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

- http://www.dataminingbook.info