COMP47460

Clustering - Part 2

Aonghus Lawlor
## Deepak Anjwani

UCD DUBLIN

Overview

- Part 1
    - Supervised v Unsupervised Learning
    - Partitional Clustering
        - k-Means clustering
        - Cluster initialisation
- Part 2
    - Hierarchical Clustering
        - Agglomerative algorithms
        - Cluster metrics
        - Divisive algorithms
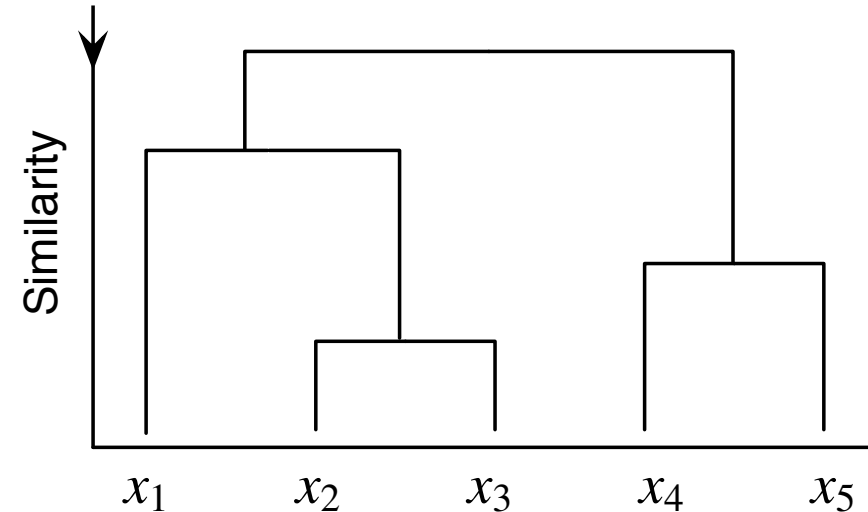   - Cluster Validation

Clustering

- **General goal**: Assign similar items to the same cluster, keep dissimilar items apart.

- Algorithms employ different definitions of similarity/dissimilarity and objective function for determining a "good" cluster.

Partitional Algorithms

Hierarchical Algorithms
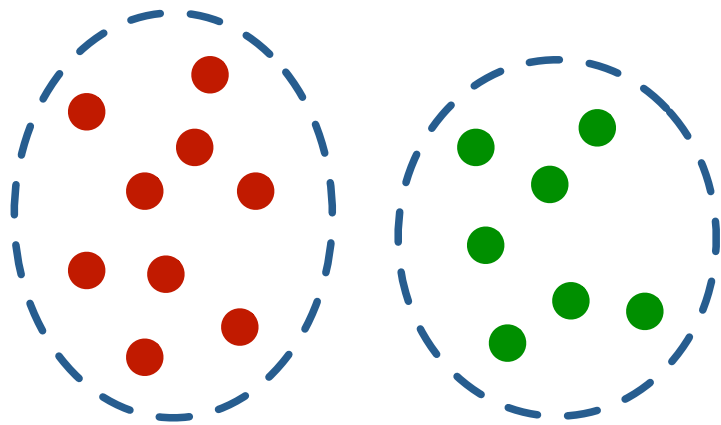
Build a "flat" clustering of the data all at once

Gradually build a nested tree structure of clusters

Hierarchical Clustering

- Instead of generating a flat partition of data, it can be useful to construct a hierarchy of items by producing a set of nested clusters that are arranged to form a tree structure.

- Hierarchical structure allows for multiple levels of granularity…

  News → Sport → Olympics → Athletics

  News → Sport → Rugby → World Cup

- Two distinct categories of hierarchical clustering algorithm:

  1. Agglomerative: Begin with each item assigned to its own cluster. Apply a bottom-up strategy where, at each step, the most similar pair of clusters are merged.

  2. Divisive: Begin with a single cluster containing all items. Apply a top-down strategy where, at each step, a chosen cluster is split into two sub-clusters.
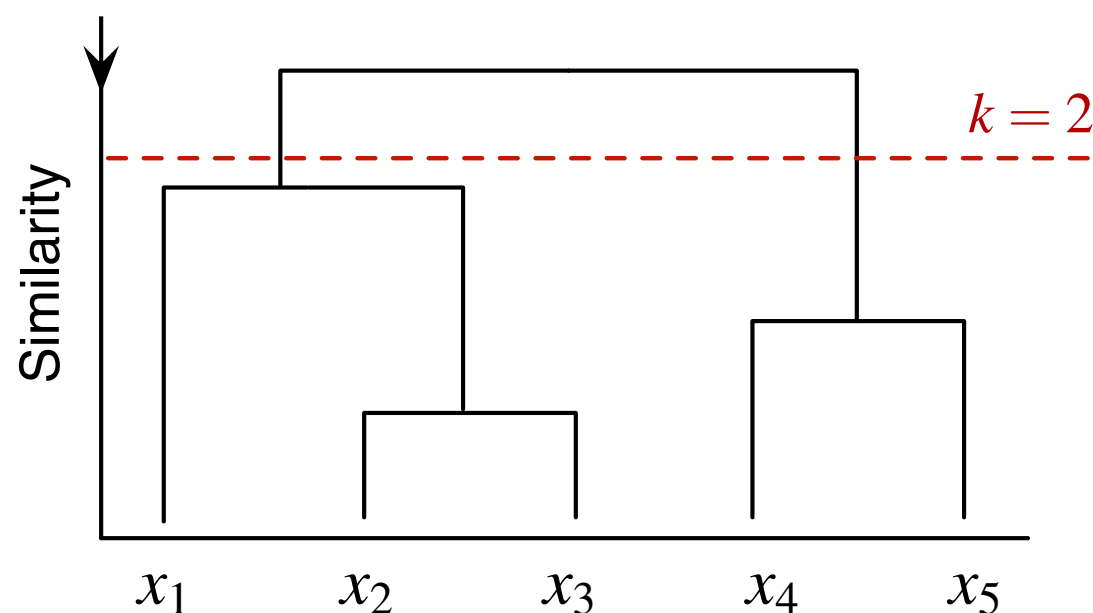
Dendrograms

Dendrogram: A tree diagram, frequently used to illustrate arrangement of clusters produced by a hierarchical clustering algorithm. General groups are near the top of the tree, more granular groups at the bottom.
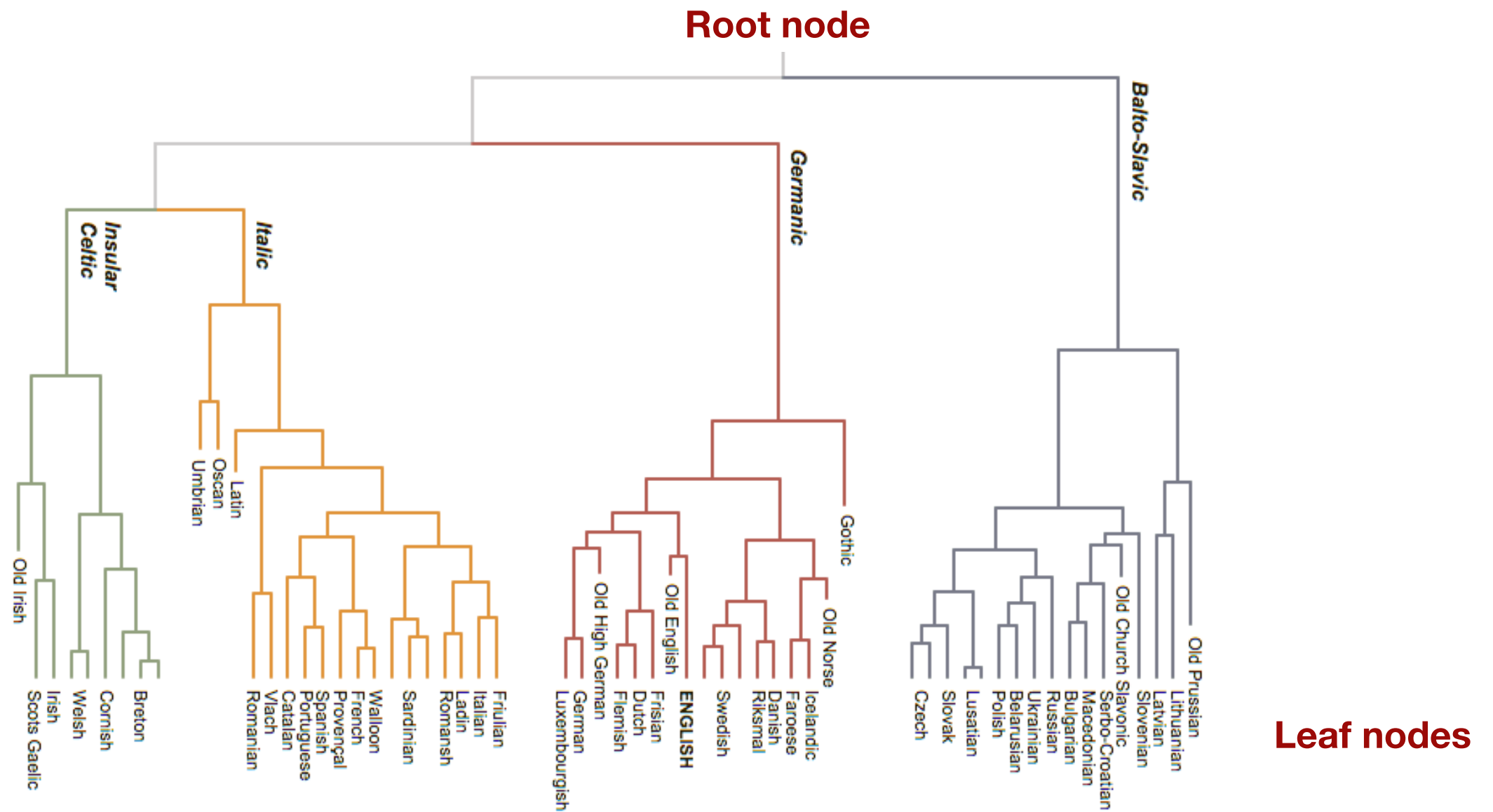
Dendrograms

- A dendrogram contains nodes for each cluster, with relations illustrating the merge or split operations that were performed during the clustering process.

$$\mathcal{C} = \{\{x_1, x_2, x_3, x_4, x_5\}\}$$

$$\mathcal{C} = \{\{x_1, x_2, x_3\}, \{x_4, x_5\}\}$$

$$\mathcal{C} = \{\{x_1\}, \{x_2, x_3\}, \{x_4, x_5\}\}$$

$$\mathcal{C} = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5\}\}$$

$$\mathcal{C} = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$$

- **Advantage:** We generally do not need to specify the number of clusters k in advance.

- Construct a tree, then allow the user to manually select k by examining the dendrogram to find an appropriate cut-off point.

Hierarchical Clustering: Applications

Hierarchical clustering is frequently applied in biology when studying gene expression data to infer biological function of unknown genes. Often want to cluster both genes and experiments (conditions).
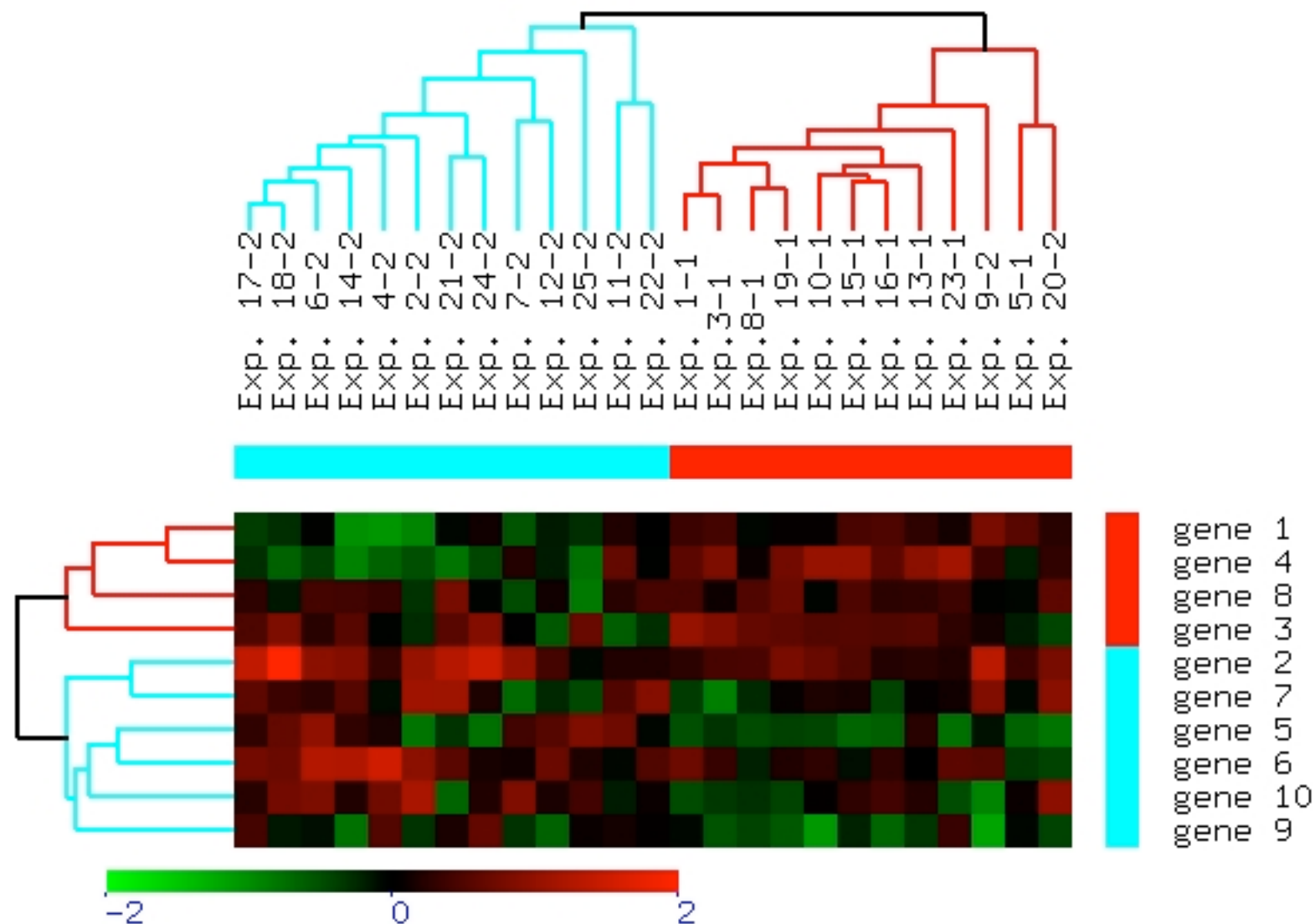
Hierarchical Clustering: Applications

Hierarchical clustering also often applied to document collections.

Algorithms can identify both high-level (broad) topics and low-level (more granular) topics.

http://nlp.stanford.edu/IR-book
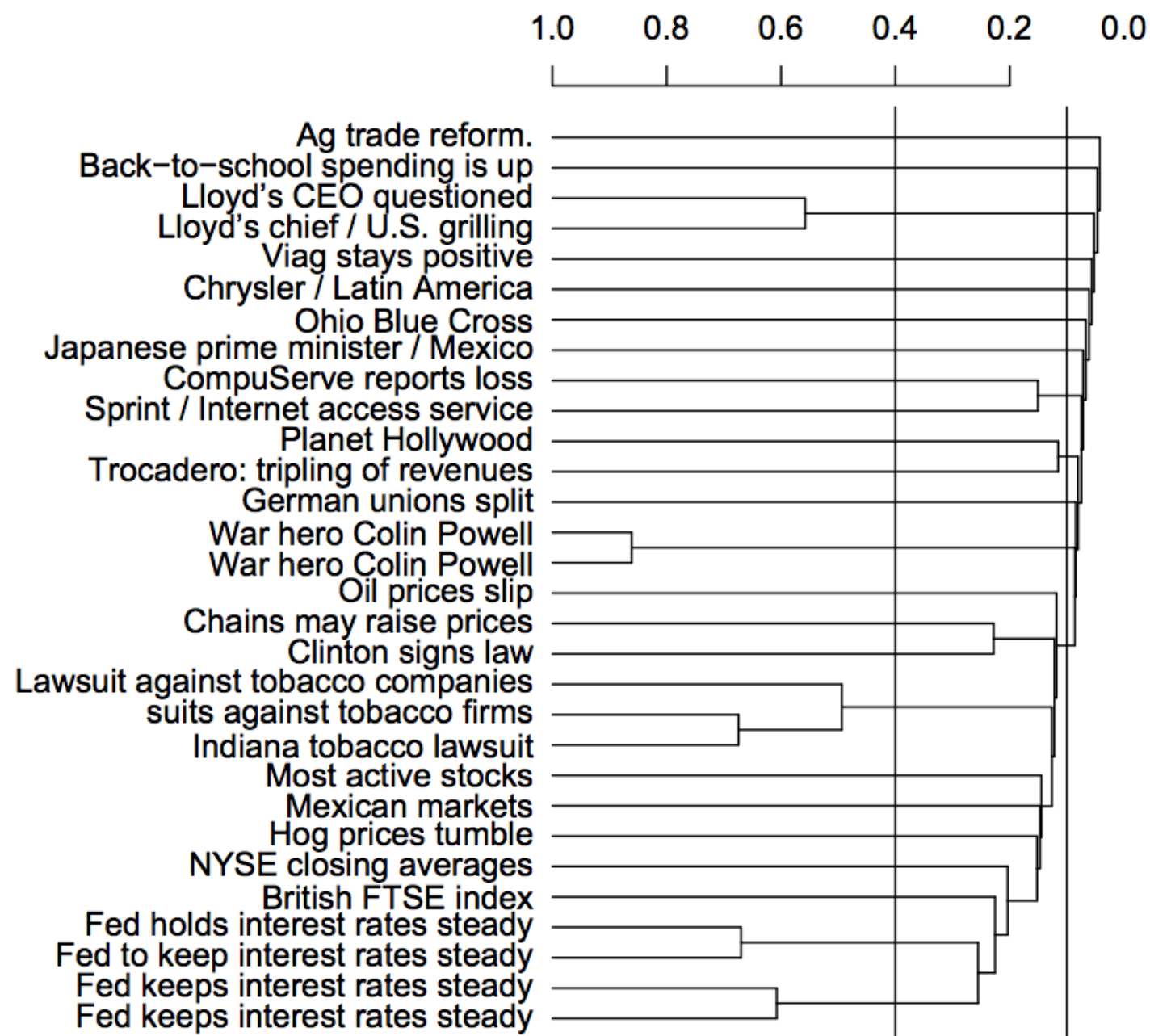
Agglomerative Clustering

Algorithm Inputs:

- *Distance matrix* $\mathbf{D}$, specifying the distance between each pair of items in the data, computed using some appropriate measure (e.g. Euclidean).

- *Cluster metric* which helps decide which pair of clusters to merge at each step, using values from $\mathbf{D}$.

Algorithm summary:

1. Assign every item to its own cluster, each just containing that item. These are the "leaf nodes" of the tree.

2. Find the closest (i.e. most similar) pair of clusters, according to the cluster metric, and merge them into a single cluster, so that now you have one less cluster.

3. Compute distances (similarities) between the new cluster and each of the remaining old clusters.

4. Repeat from Step 2 until all items are clustered into a single cluster. This is the "root node" of the tree.
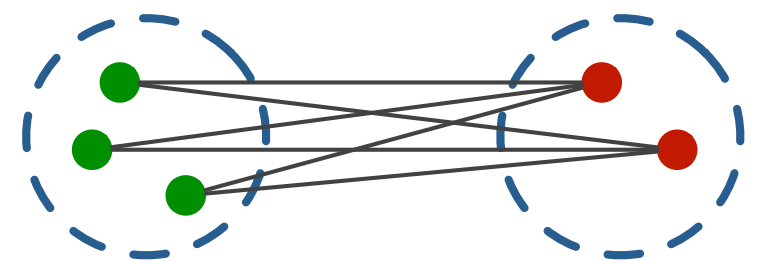
Cluster Metrics

- A variety of metrics exist for determining which pair of clusters should be merged next from among all possible pairs. These specify how we use values from $\mathbf{D}$ to measure the distance between two clusters.

  - Single linkage: Define cluster distance as the <u>smallest</u> pairwise distance between items from each cluster.

  - Complete linkage: Define cluster distance as the <u>largest</u> pairwise distance between items from each cluster.

  - Average linkage: Define cluster distance as the <u>average</u> of all pairwise distances between items from each cluster.
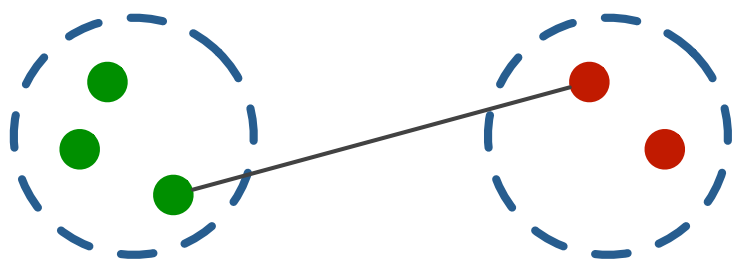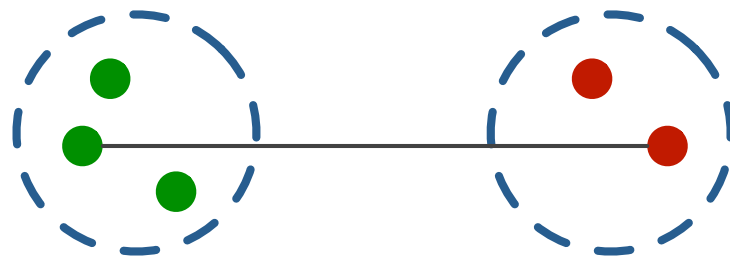
Single Linkage   Complete Linkage   Average Linkage

Cluster Metrics

- Formulae for cluster distance metrics, where D_{ij} is the distance between the *i*-th and *j*-th items in distance matrix **D**:

Single linkage

$$d(C_a, C_b) = \min_{x_i \in C_a, x_j \in C_b} D_{ij}$$

Complete linkage

$$d(C_a, C_b) = \max_{x_i \in C_a, x_j \in C_b} D_{ij}$$

Average linkage

$$d(C_a, C_b) = \frac{\sum_{x_i \in C_a} \sum_{x_j \in C_b} D_{ij}}{|C_a| \, |C_b|}$$

- The choice of cluster distance metric can substantially affect the resulting clustering.

➡ Complete linkage is sensitive to outliers. Single linkage tends to produce long chains, not cohesive clusters.

Example: Agglomerative Clustering

- Given a data set of four items represented by two features, construct a Euclidean distance matrix **D**.

data matrix

x_1	2	1
x_2	0	0
x_3	1	1
x_4	0	3

$$\mathrm{ED}(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{f \in F} (q_f - p_f)^2}$$
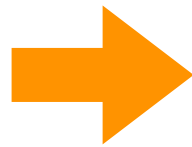
4 x 4 distance matrix

	x_1	x_2	x_3	x_4
x_1	0	2.24	1.00	2.83
x_2	2.24	0	1.41	3.00
x_3	1.00	1.41	0	2.24
x_4	2.83	3.00	2.24	0

- Two clusters $C_a = \{x_1, x_3\}$ and $C_b = \{x_2, x_4\}$. Then $d(C_a, C_b)$ is…

Single linkage   min(2.24,2.83,1.41,2.24) = 1.41

Complete linkage   max(2.24,2.83,1.41,2.24) = 2.83

Average linkage   (2.24+2.83+1.41+2.24)/4 = 2.18

Agglomerative Clustering in WEKA

In the Weka *Cluster* tab, choose *HierarchicalClusterer* as the clusterer. Right-click on *HierarchicalClusterer* name to change parameters (e.g. distance function, linkage, metric).
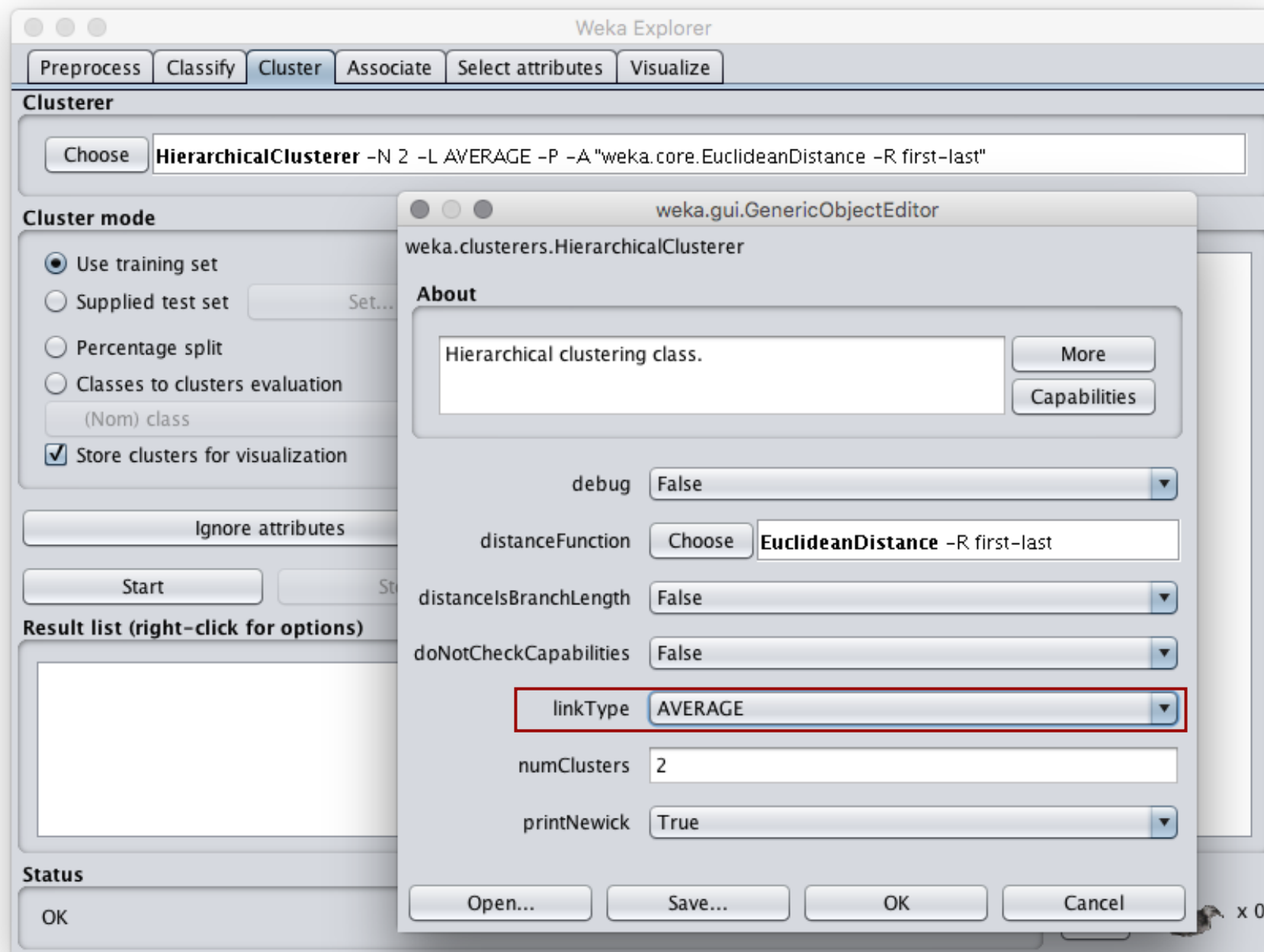
Example: diabetes.arff

Agglomerative Clustering in WEKA

Hit *Start* to run agglomerative clustering on the data until the specified minimum number of clusters remains (e.g. 2 clusters).
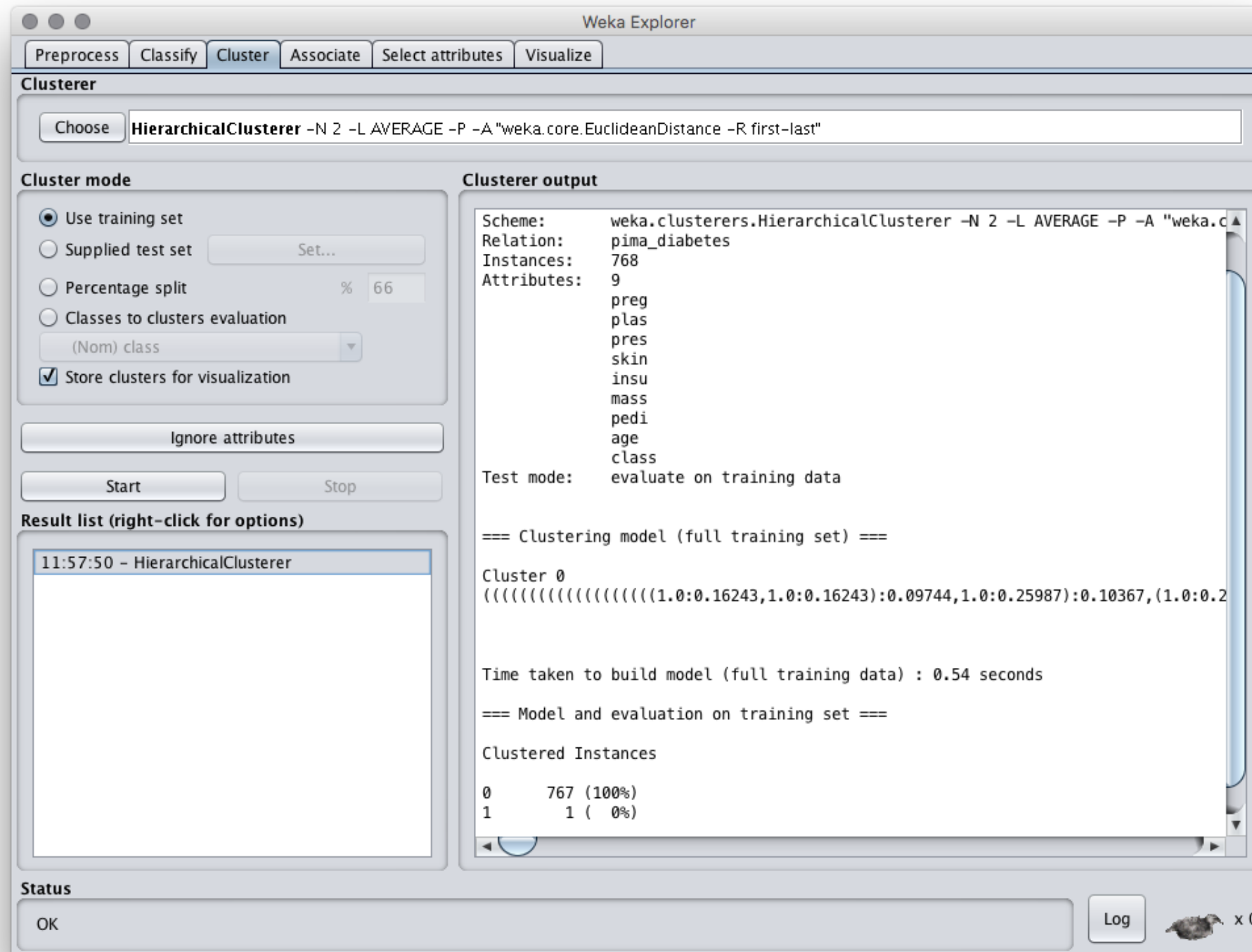
Agglomerative Clustering in WEKA

Right-click on the result in *Result List* and choose *Visualize Tree* to display the resulting dendrogram.
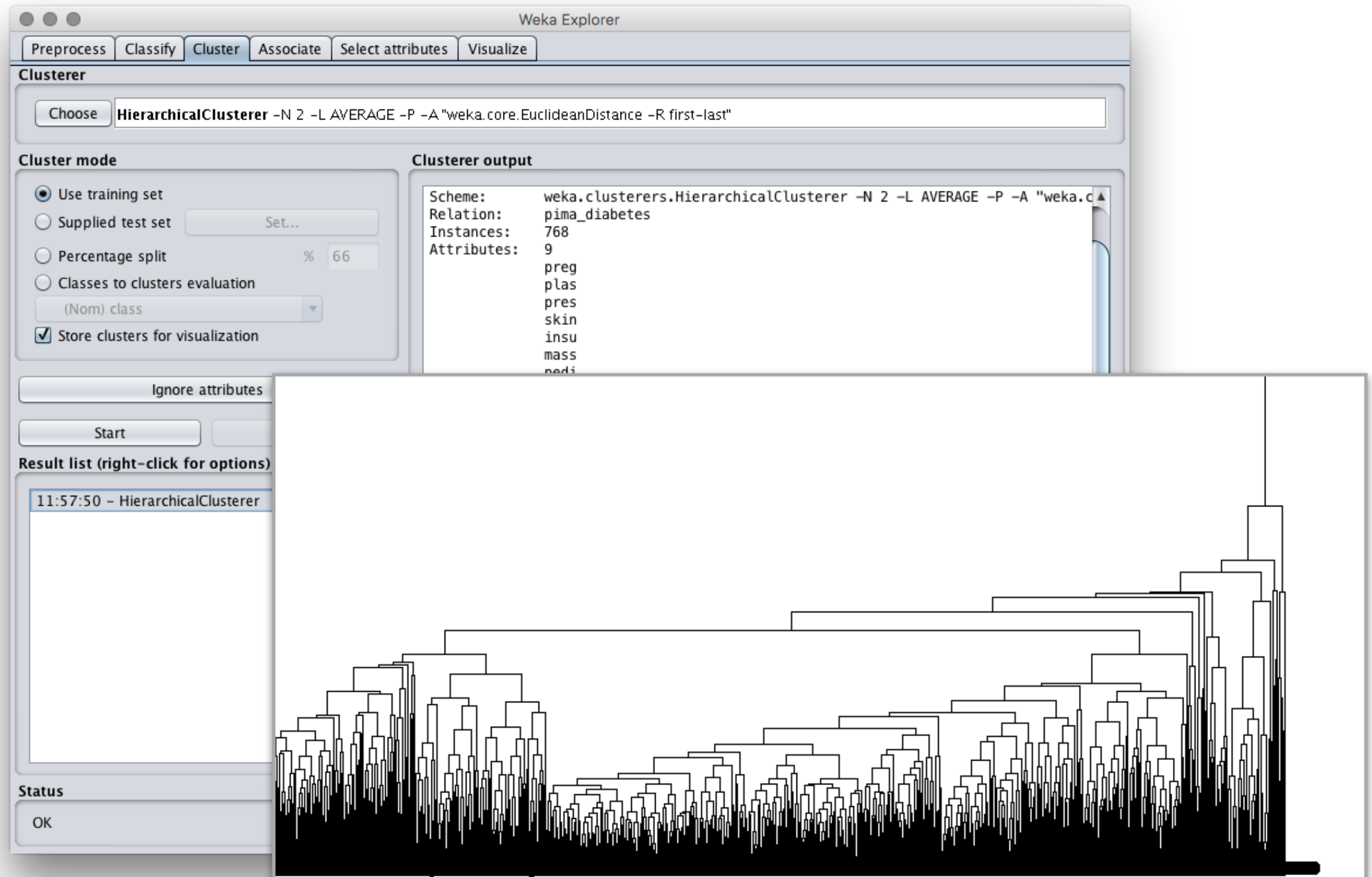
Divisive Algorithms

Divisive Hierarchical Clustering Template:

- Start with all items in a single cluster.

- REPEAT until all items are in their own cluster

  - Choose an existing cluster to split using some splitting criterion.

  - Replace the chosen cluster into two sub-clusters.
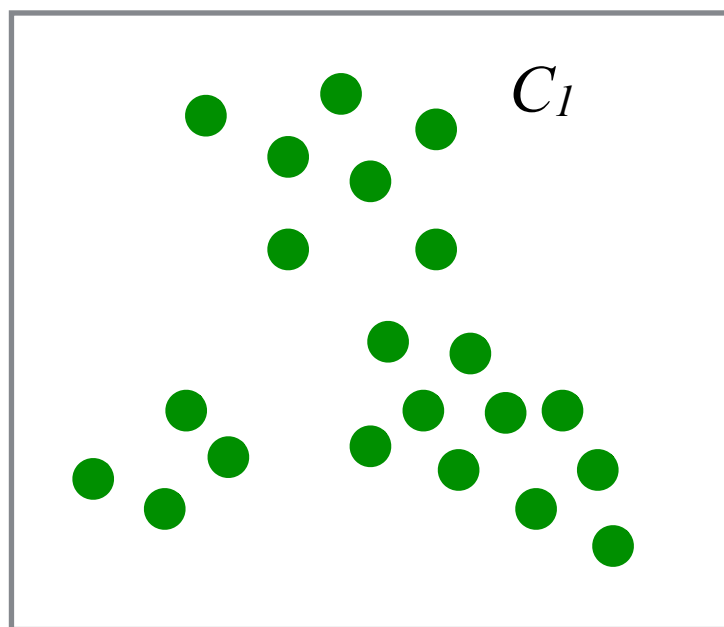
Assign all items to C_1
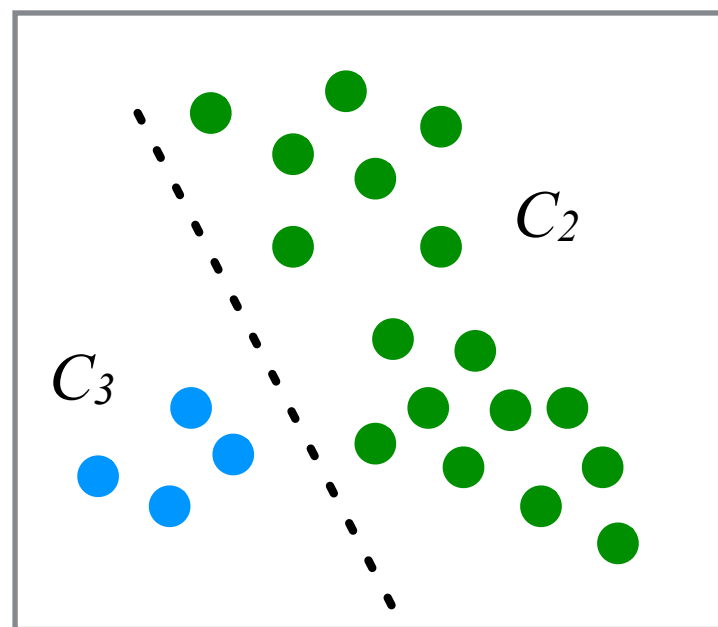
Split C_1 and replace with sub-clusters C_2 and C_3
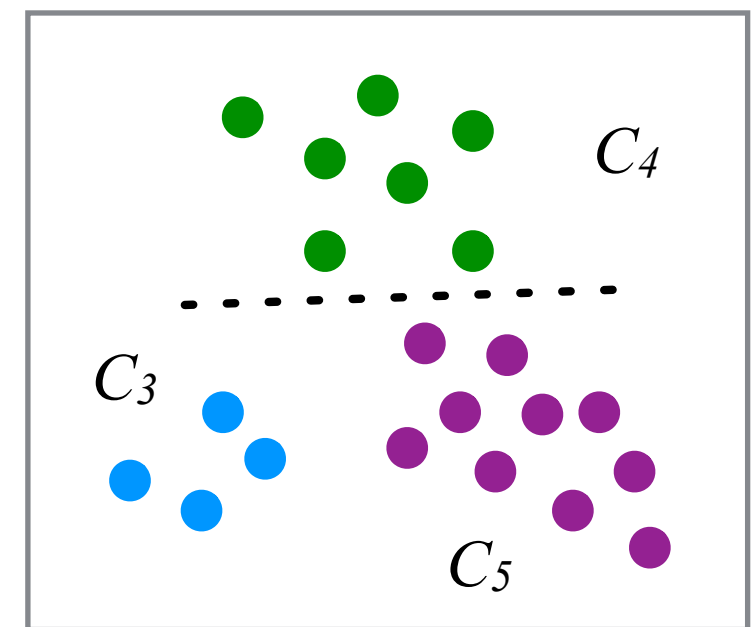
Split C_2 and replace with sub-clusters C_4 and C_5

Divisive Algorithms

By recursively using a divisive bisecting clustering procedure, the obtained clusters are structured as a hierarchical binary tree.
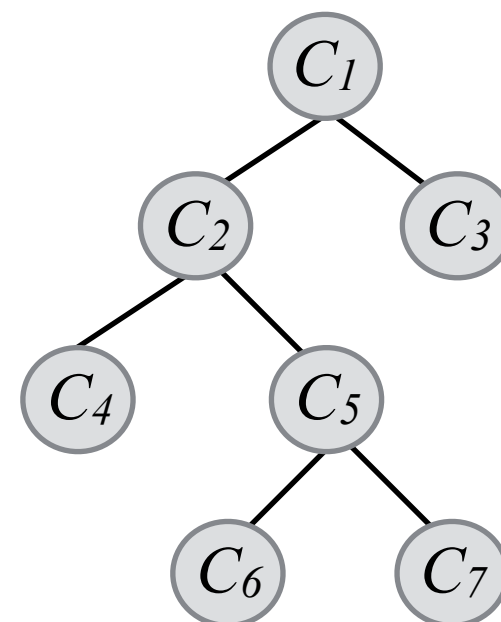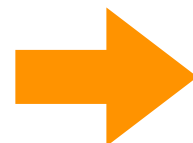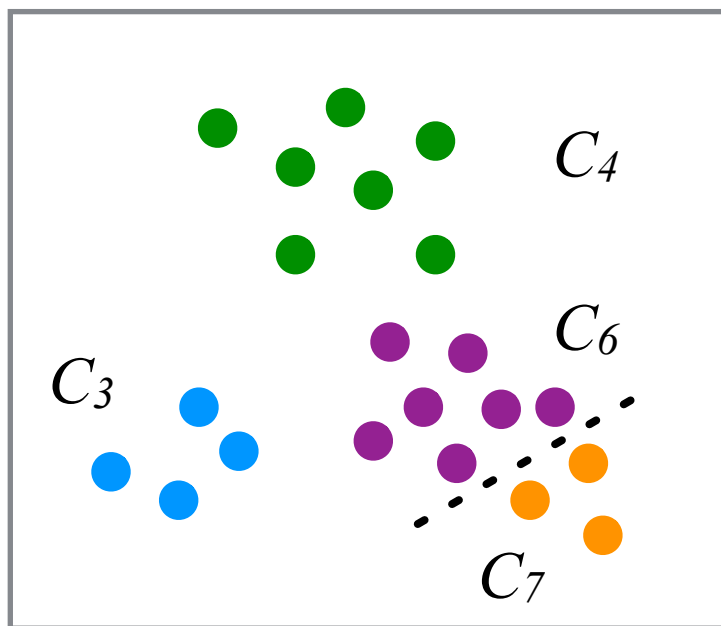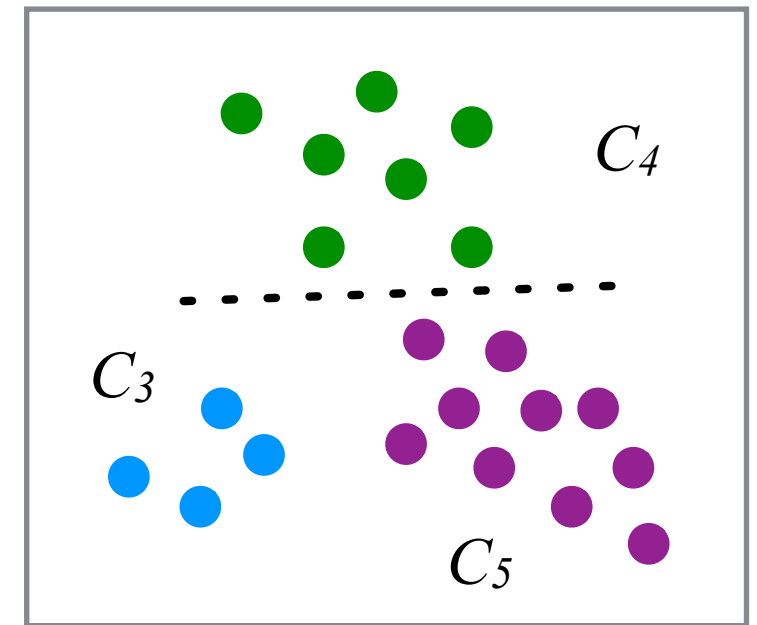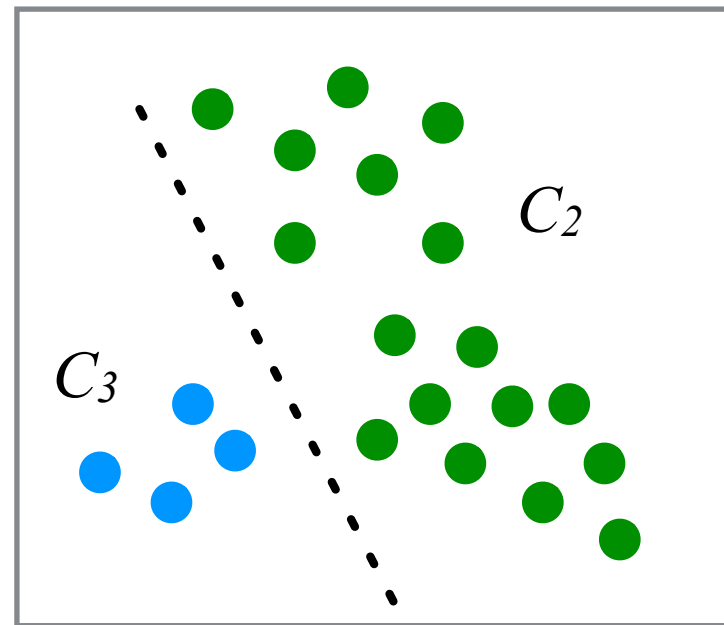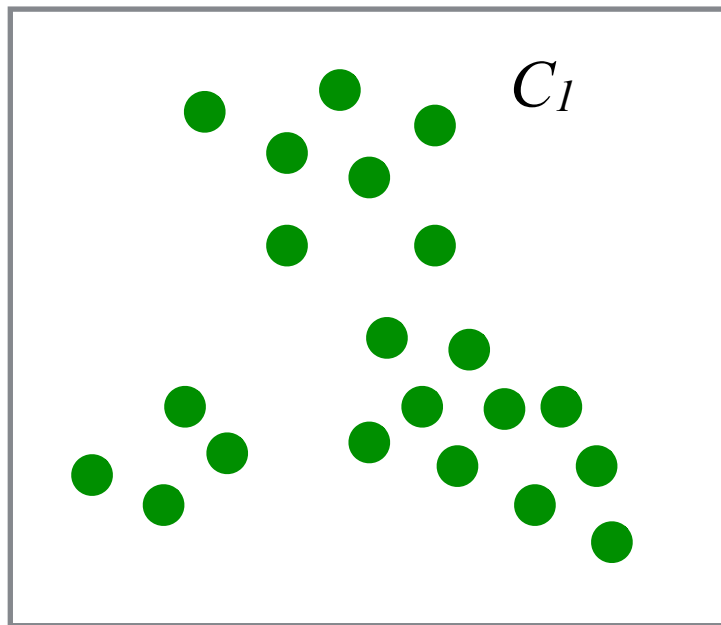
Binary tree with root node C_1 and four leaf nodes $\{C_3, C_4, C_6, C_7\}$

Bisecting *k*-Means Algorithm

- Key idea: Apply a standard partitional algorithm (i.e. *k*-Means) to split a single cluster into two sub-clusters.

Algorithm Summary:

1. Assign all items to a single cluster.

2. Choose a cluster C_i to split that optimises a given splitting criterion. Common approach: select cluster that has the lowest mean intra-cluster similarity score:
$$MeanIntra(C_c) = \frac{\sum_{x_i, x_j \in C_c} S_{ij}}{|C_c|^2}$$

3. Generate two sub-clusters of C_i by applying *k*-Means with *k=2* to only the items assigned to C_i.

4. Replace the original cluster C_i with the resulting pair of sub-clusters.

5. Repeat from Step 2 until *k* clusters have been generated.

Hierarchical Clustering

- **Advantages:**

  - Allows for multiple levels of granularity, both broad clusters and niche clusters.

  - No requirement to select the "correct" value for number of clusters k in advance.

- **Disadvantages:**

  - Poor decisions made early in the clustering process can greatly influence the quality of the final clustering.

  - Once a merging or splitting decision has been made, there exists no facility to rectify a mistake at a later stage.

  - More computationally expensive than partitional methods, particularly for agglomerative clustering.

Cluster Validation

Q. How many clusters in a given data set? Usually will not know in advance…

Well-separated clusters *(k=3)*
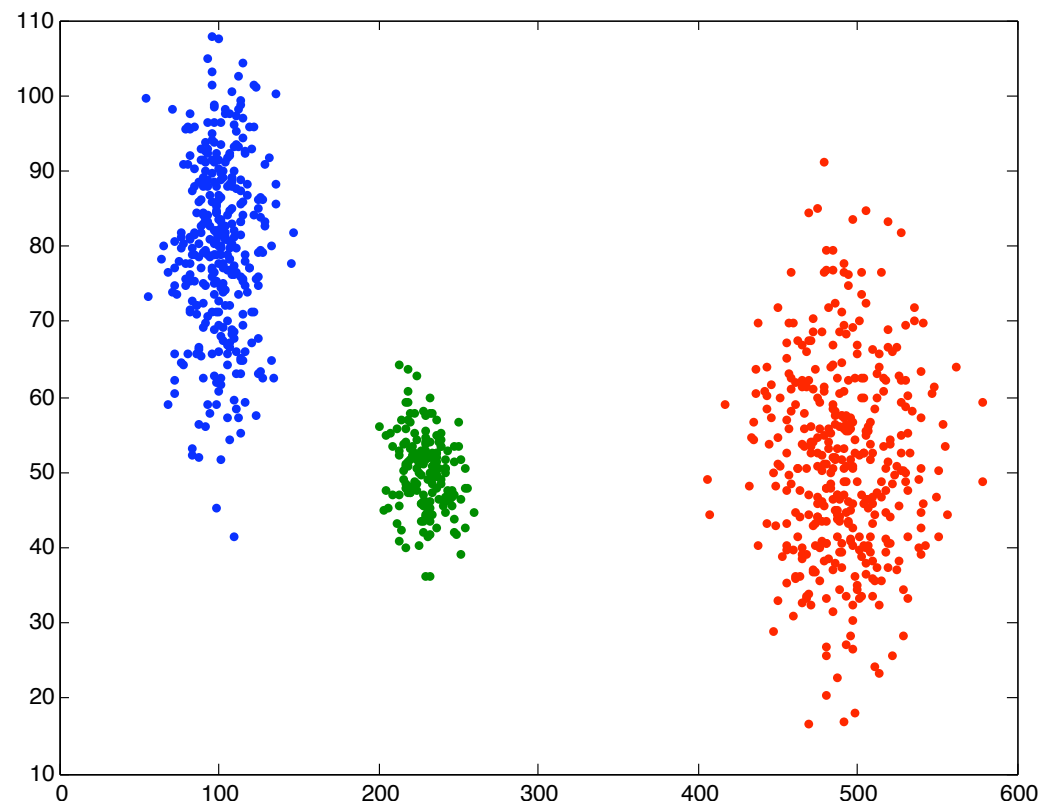
Poorly-separated clusters *(k=4)*
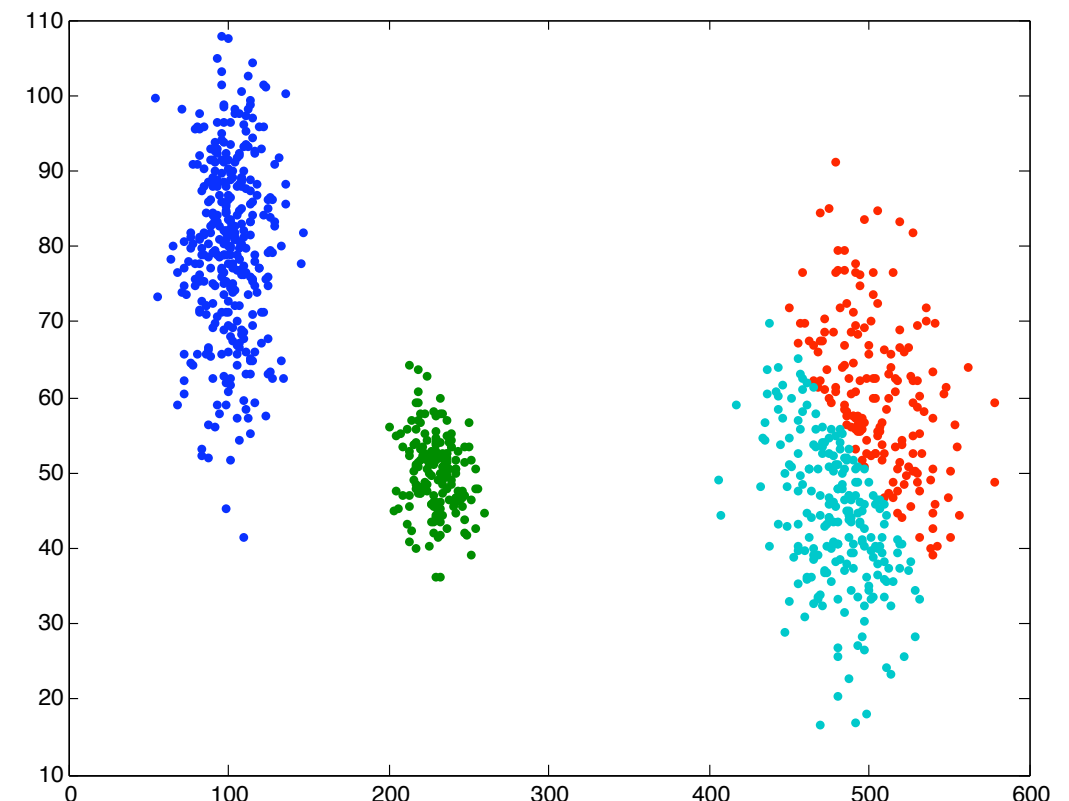
Q. How can we distinguish between a "good" and a "bad" clustering? How can we choose between different clusterings or different clustering algorithms?

Cluster Validation

- Cluster validation: Measures for automatically producing a quantitative evaluation of the quality of a clustering.

- Common motivation - "good" clusters have the property that cluster members are close to each other and far from members of other clusters.

- Cluster validation is often applied for parameter selection - e.g. select an appropriate value k for the k-Means algorithm.

- **Typical Strategy:**

  1. Apply k-Means for each value from k_{min} to k_{max}.

  2. Calculate score for each clustering using a cluster validation measure.

  3. Examine plot of scores to identify a peak for the best value for k.
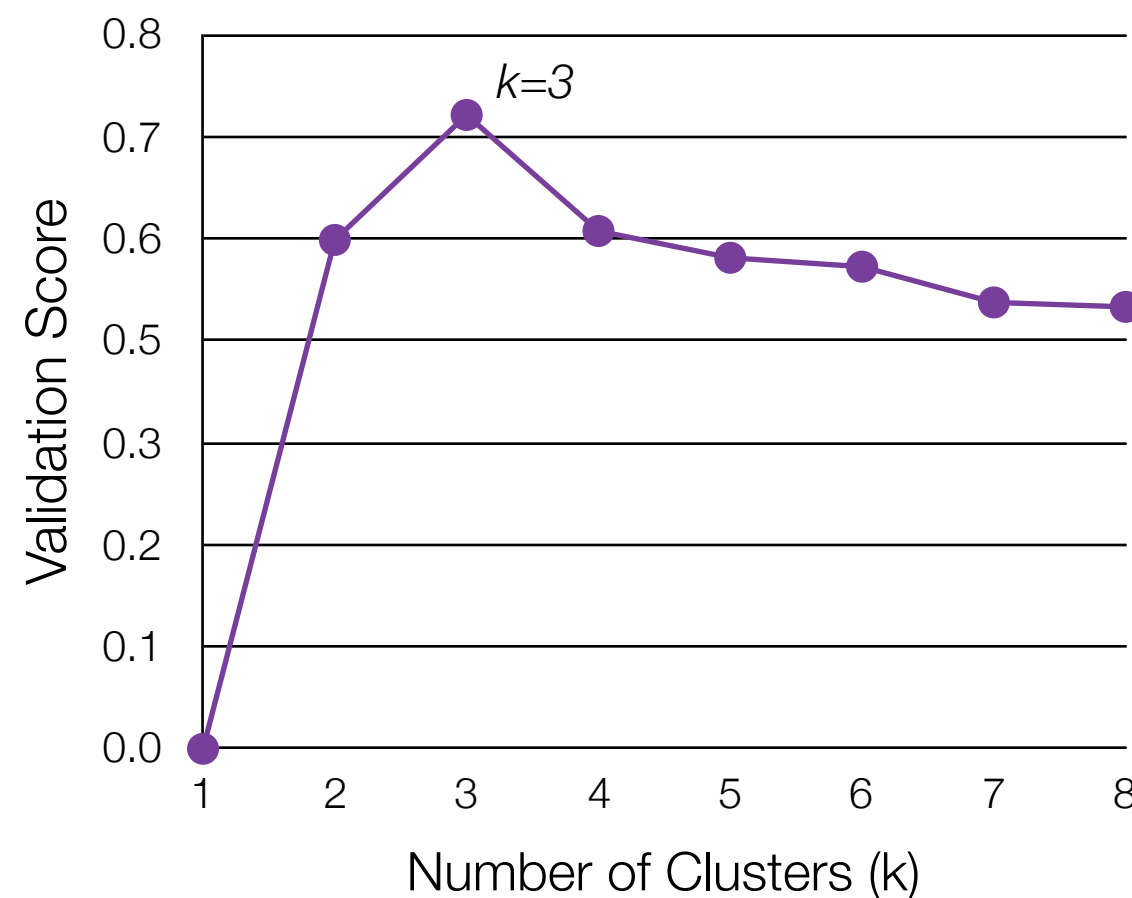
Silhouette Measure

- Validation measure which quantifies degree to which each item belongs in its assigned cluster, relative to the other clusters.
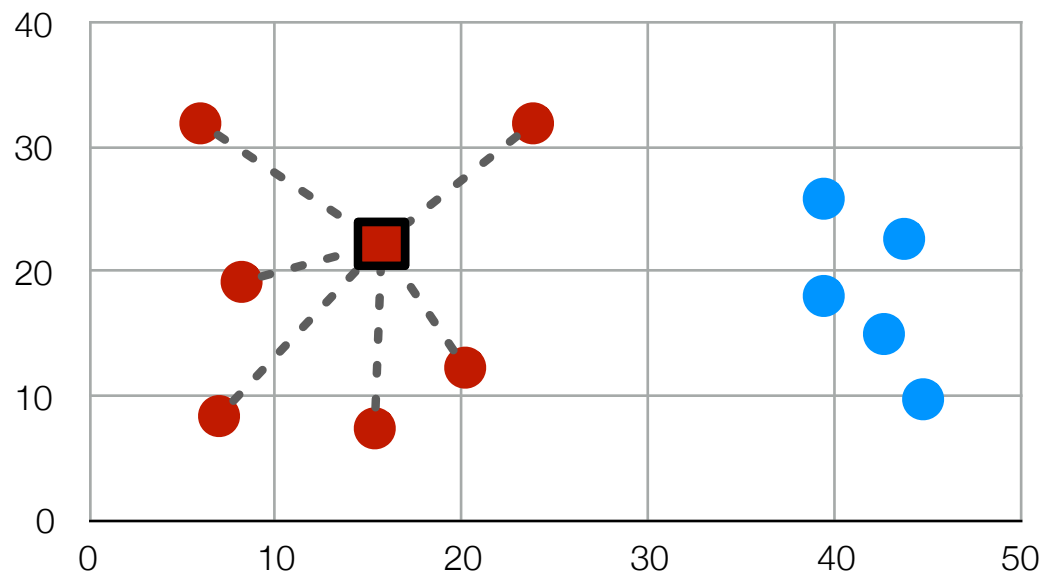
Measure average distance to all other items in same cluster.

$$a_i = \frac{1}{|C_h| - 1} \sum_{j \in C_h, j \neq i} d(i,j)$$
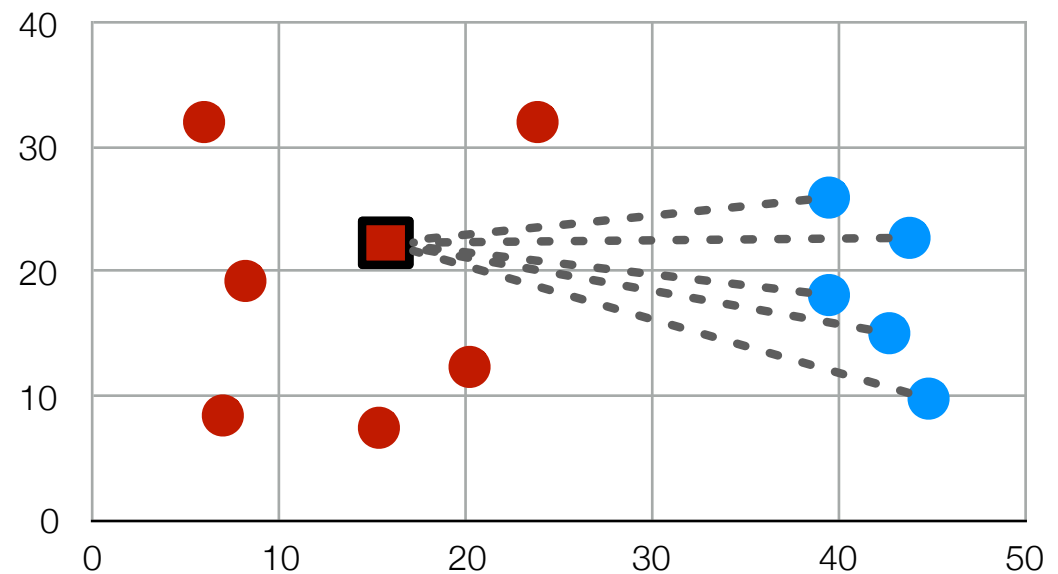
Measure average distance to all other items in nearest competing cluster.

$$b_i = \frac{1}{|C_l|} \sum_{j \in C_l} d(i,j)$$

- Silhouette width for an item x_i is given by s_i. Values are in the range [-1,1], a larger value is better.

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

Silhouette Measure

- **Silhouette width** for a single item x_i is calculated as s_i.

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

- **Average Silhouette Width (ASW)**: Calculate overall score for a clustering by averaging the silhouette widths for all n items.

$$ASW(\mathcal{C}) = \frac{1}{n} \sum_{i=1}^{n} s_i$$

Strategy:

1. Apply *k*-Means for each value from k_{min} to k_{max}.

2. Calculate ASW for each clustering.

3. Examine plot of scores to identify a peak for the best value for *k*.
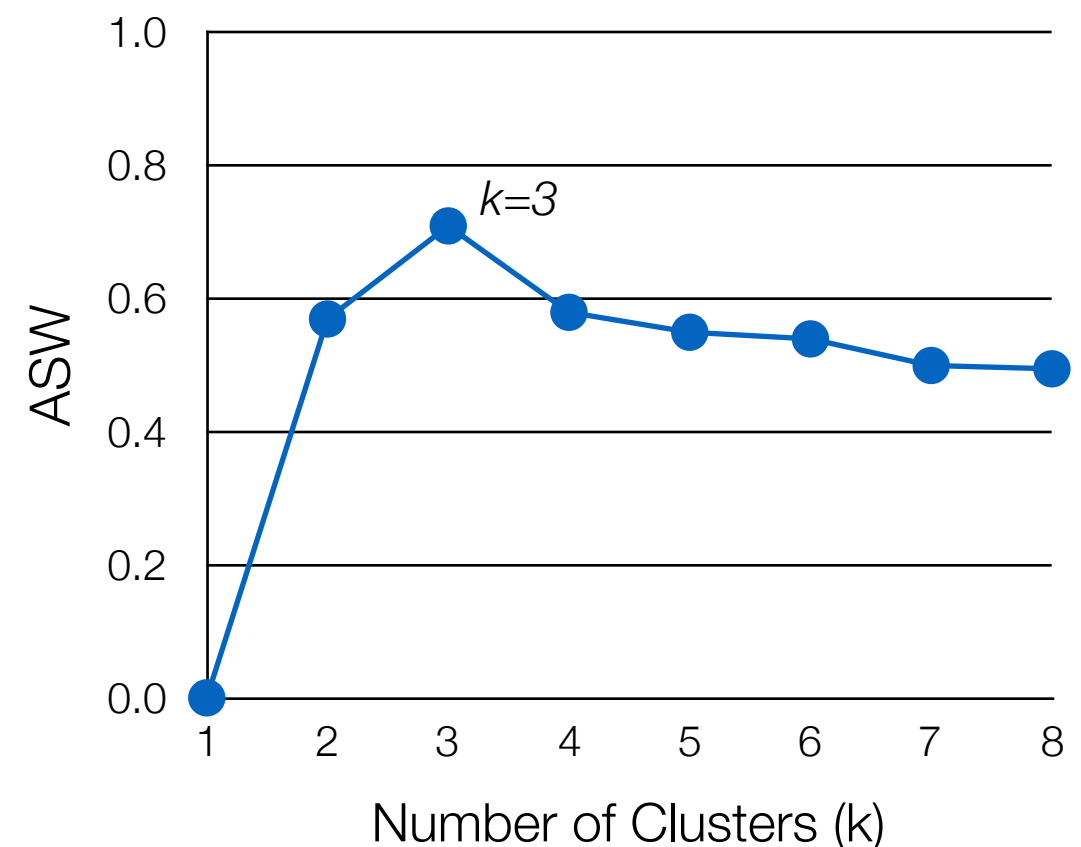
Hierarchical Cluster Validation

- Hierarchical clusterings are represented as a dendrogram.

- Cutting a dendrogram at a certain level gives a set of flat clusters. Cutting at another level gives another set of flat clusters.

- How can we select where to cut the dendrogram?

$k = 2$

$k = 4$
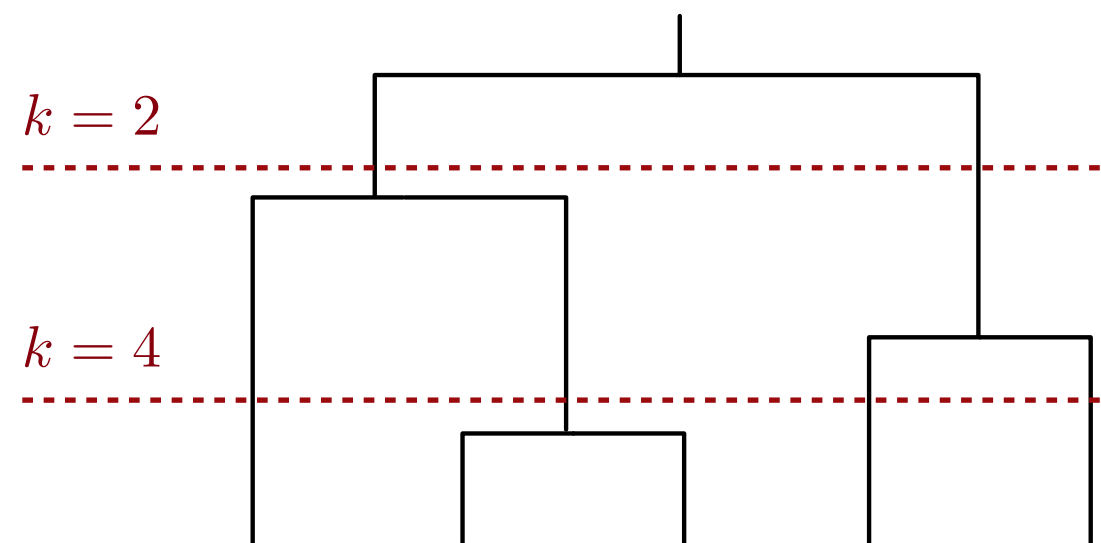
Example cuts of a hierarchy for 2 clusters and 4 clusters

➡ Generate hierarchical clustering, then apply cluster validation measures at all possible cut-off levels to identify best level.

- Many validation measures that can be applied, typically rewarding "compact and well separated clusters" - e.g. Dunn index, DB index

On Clustering Validation Techniques - Halkidi et al (2001)

http://dl.acm.org/citation.cfm?id=607609

External Cluster Validation

In practice, we may have some type of external information that can be used to evaluate a clustering, and compare different clusterings.

Gene Ontology Database

http://geneontology.org

DBpedia Database

http://dbpedia.org

Summary

- Part 1

  - Supervised v Unsupervised Learning

  - Partitional Clustering

    - *k*-Means clustering

    - Cluster initialisation

- Part 2

  - Hierarchical Clustering

    - Agglomerative algorithms

    - Cluster metrics

    - Divisive algorithms - Bisecting *k*-Means

  - Cluster Validation