Lecture 7: Other approaches to clustering

Felix Held, Mathematical Sciences

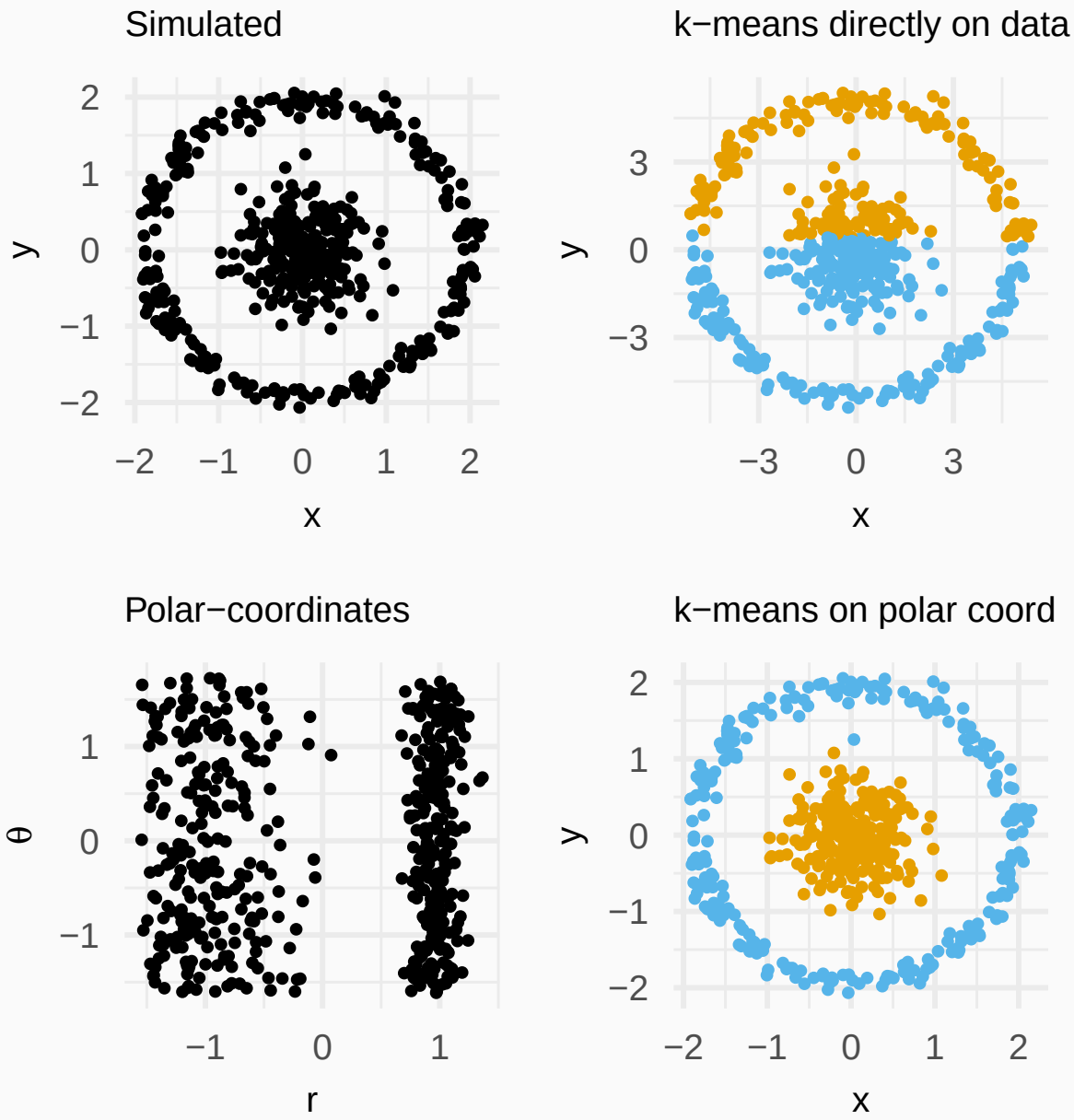
MSA220/MVE440 Statistical Learning for Big Data

8th April 2019

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

k-means and the assumption of spherical geometry



Challenges in clustering

Two main challenges

1. How many clusters are there?
2. Given a number of clusters, how do we find them?

Challenge 2 is typically approached by minimizing **within-cluster point scatter** over clusterings C

$$W(C) = \sum_{i=1}^{K} \sum_{\substack{l=1 \\ C(\mathbf{x}_l)=i}}^{n} \sum_{\substack{m<l \\ C(\mathbf{x}_m)=i}} D(\mathbf{x}_l, \mathbf{x}_m)$$

Full exploration of all clusterings is computationally too expensive. One popular approximation is **k-means**.

Partition around medoids (PAM) or k-medoids

Restrictions of k-means: Features have to be continuous and the ℓ_2 norm has to be used as a distance measure.

Idea: Similar approximation but use general distance measure. Also, use one of the observations as cluster centre (a **medoid**), not the centroid.

Solve

$$\underset{\substack{C \\ l_i \text{ for } 1 \leq i \leq K}}{\arg\min} \sum_{i=1}^{K} N_i \sum_{\substack{l=1 \\ C(\mathbf{x}_l)=i}}^{n} D(\mathbf{x}_l, \mathbf{x}_{l_i})$$

Notation: For observed feature vectors $\mathbf{x}_l$ and $\mathbf{x}_m$ set $\mathbf{D}_{l,m} = D(\mathbf{x}_l, \mathbf{x}_m)$. This results in $\mathbf{D} \in \mathbb{R}^{n \times n}$.

PAM/k-medoids algorithm

Computational procedure:

1. **Initialize:** Randomly choose K observation indices as cluster centres l_i and set $J_{\max}$
2. For steps $j = 1, \dots, J_{\max}$
   - 2.1 **Cluster allocation:** $C(\mathbf{x}_l) = \underset{1 \le i \le K}{\arg\min}\, \mathbf{D}_{l,l_i}$
   - 2.2 **Cluster centre update:** $l_i = \underset{\substack{1 \le l \le n \\ C(\mathbf{x}_l)=i}}{\arg\min} \sum_{C(\mathbf{x}_m)=i} \mathbf{D}_{l,m}$
   - 2.3 Stop if clustering C did not change

Computational Complexity: Step 2.2 is now quadratic in n_i instead of linear as in k-means

Note: All PAM requires is a matrix of distances $\mathbf{D}$ and no additional distance computations are necessary. Very diverse types of features can be used.

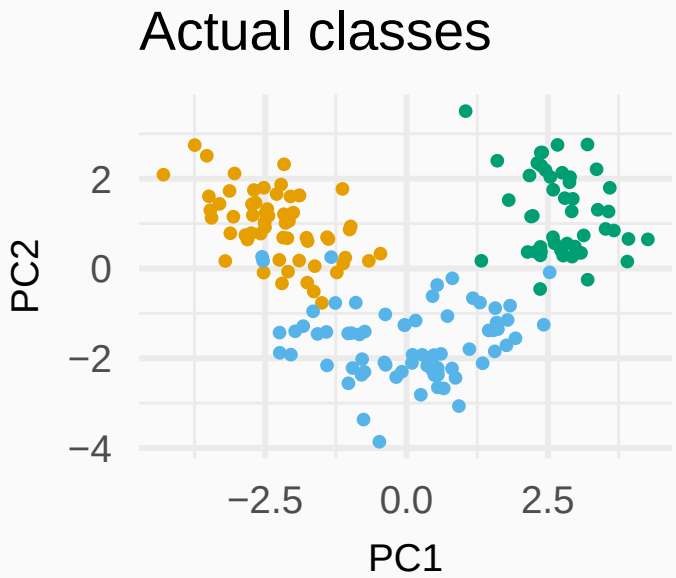
Selection of cluster count

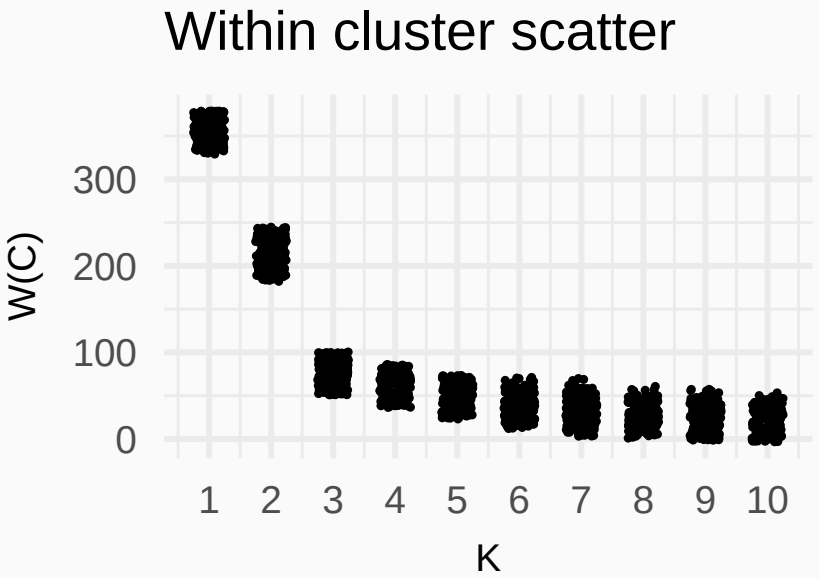
A simple heuristic to pick cluster count

Challenge: How many clusters?

Elbow heuristic:

- $W(C)$ decreases with cluster count K, but decreases are less substantial if data does not support more clusters.
- K is chosen such that the **decrease it provided is substantially larger than the next value** of K.





Silhouette Width

Clustering goal: **Maximize** between cluster scatter and **minimize** within cluster scatter

For every observation $\mathbf{x}_l$ do

1. **Average distance within cluster:**

$$a_l = \frac{1}{n_{C(\mathbf{x}_l)}} \sum_{C(\mathbf{x}_m)=C(\mathbf{x}_l)} \mathbf{D}_{l,m}$$

2. **Average distance to nearest cluster:**

$$b_l = \arg\min_{\substack{1 \le i \le K \\ i \neq C(\mathbf{x}_l)}} \frac{1}{n_i} \sum_{C(\mathbf{x}_m)=i} \mathbf{D}_{l,m}$$

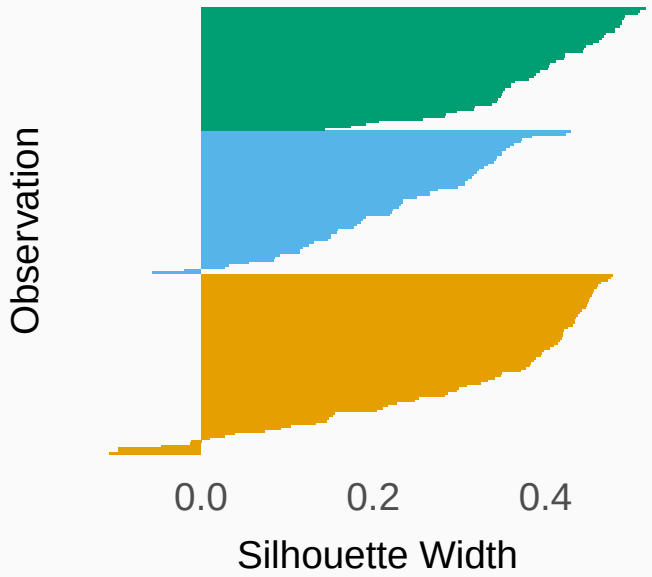
3. **Silhouette width:** $s_l = \dfrac{b_l - a_l}{\max(a_l, b_l)} \in [-1, 1]$

Notes on silhouette width

- Interpretation
  - Close to 1 when observation is well located inside the cluster and separated from the nearest cluster
  - Close to 0 when observation is between two clusters
  - Negative if observation on average closer to another cluster. **Warning sign:** Hints at which observations should be investigated.
- **Average silhouette width:** $S = \frac{1}{n} \sum_{l=1}^{n} s_l$ should be maximal for a good clustering
- Limitations
  - Needs at least two clusters
  - Based on the same ideas as PAM/k-medoids and therefore considers clusters to be spherical
  - Silhouette width tends to favour fewer clusters

Silhouette Width: Example

Silhouette width applied to the UCI wine data. Sorted by cluster and arranged in decreasing order.



- Silhouette width gives a clear signal that more than three clusters lead to decreasing performance
- However, two and three clusters are indicated as almost equally good.

Combining clustering and classification

Observation: A clustering with the appropriate number of clusters should be based on **non-random structures** in the data.

Idea: The finding of the groups should be **reproducible**. Therefore, combine clustering with classification to determine the **prediction strength** of a given clustering on new data.

Cluster Prediction Strength

Procedural overview:

1. **Divide data** into two parts A and B
2. **Cluster** the data into K groups on each part separately
3. Treat the clusterings C_A and C_B as the true classes and **learn classification rules** c_A and c_B on A and B, respectively
4. Use B as a test set for c_A and A as a test set for c_B, i.e. compare $c_A(\mathbf{x})$ to $C_B(\mathbf{x})$ for $\mathbf{x} \in B$ and vice versa for A. (**Note:** Clustering labels have arbitrary order, i.e. **label matching** might have to be performed first)
5. Compute the **overall test error rate** as the average test error rate in both data sets

Selection rule: Choose K which minimizes prediction error

Notes on Cluster Prediction Strength

1. Many observations are necessary so structures are preserved in the 50:50 split datasets
2. Matching of clustering algorithm and classification method is important. They need to make similar assumptions, e.g.
   - k-means and nearest centroids make similar assumptions
   - k-means and LDA can work, even though LDA makes more flexible assumptions (ellipsoids instead of spheres)
   - PAM with categorical loss and kNN

Bottom-up approach to clustering

Two approaches to combinatorial clustering

1. **Top-down approach:** Start with all observations in one group and split them into clusters
   - e.g. k-means, PAM, ...
2. **Bottom-up approach:** Start with all observations individually and join them together to build clusters

Hierarchical Clustering

Procedural idea:

1. **Initialization:** Let each observation $\mathbf{x}_l$ be in its own cluster g_l^0 for $l = 1, \dots, n$
2. **Joining:** In step i, join the two clusters g_l^{i-1} and g_m^{i-1} that are closest to each other resulting in $n - i$ clusters
3. After $n - 1$ steps all observations are in one big cluster

Subjective choices:

- How do we measure distance between observations?
- What is **closeness** for clusters?

Linkage

Cluster-cluster distance is called **linkage**

Distance between clusters g and h

1. **Average Linkage:**

$$d(g,h) = \frac{1}{|g| \cdot |h|} \sum_{\substack{\mathbf{x}_l \in g \\ \mathbf{x}_m \in h}} \mathbf{D}_{l,m}$$

2. **Single Linkage**

$$d(g,h) = \min_{\substack{\mathbf{x}_l \in g \\ \mathbf{x}_m \in h}} \mathbf{D}_{l,m}$$

3. **Complete Linkage**

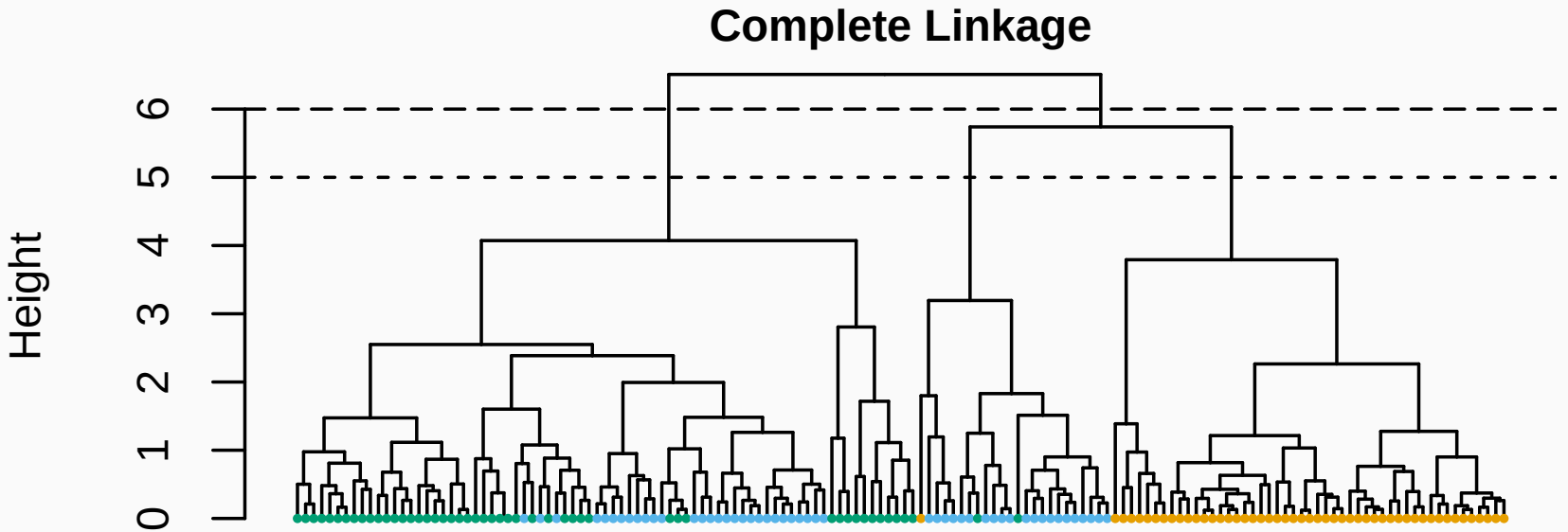
$$d(g,h) = \max_{\substack{\mathbf{x}_l \in g \\ \mathbf{x}_m \in h}} \mathbf{D}_{l,m}$$

Notes on hierarchical clustering and linkage

- Effect of linkage criterion
  - Average linkage is most commonly used and encourages average similarity between all pairs in the two clusters.
  - Single linkage tends to create clusters that are quite spread out since it only considers the closest observations between clusters
  - Complete linkage tends to produce “tight” clusters
- Linkage criteria lead to different performance on different datasets. **Try different ones and think about their assumptions.**
- Different assumptions (from e.g. k-means)
  - Clusters are joined by closeness to each other, not by closeness to some centre
  - e.g. single linkage hierarchical clustering can handle the circular data example from the beginning

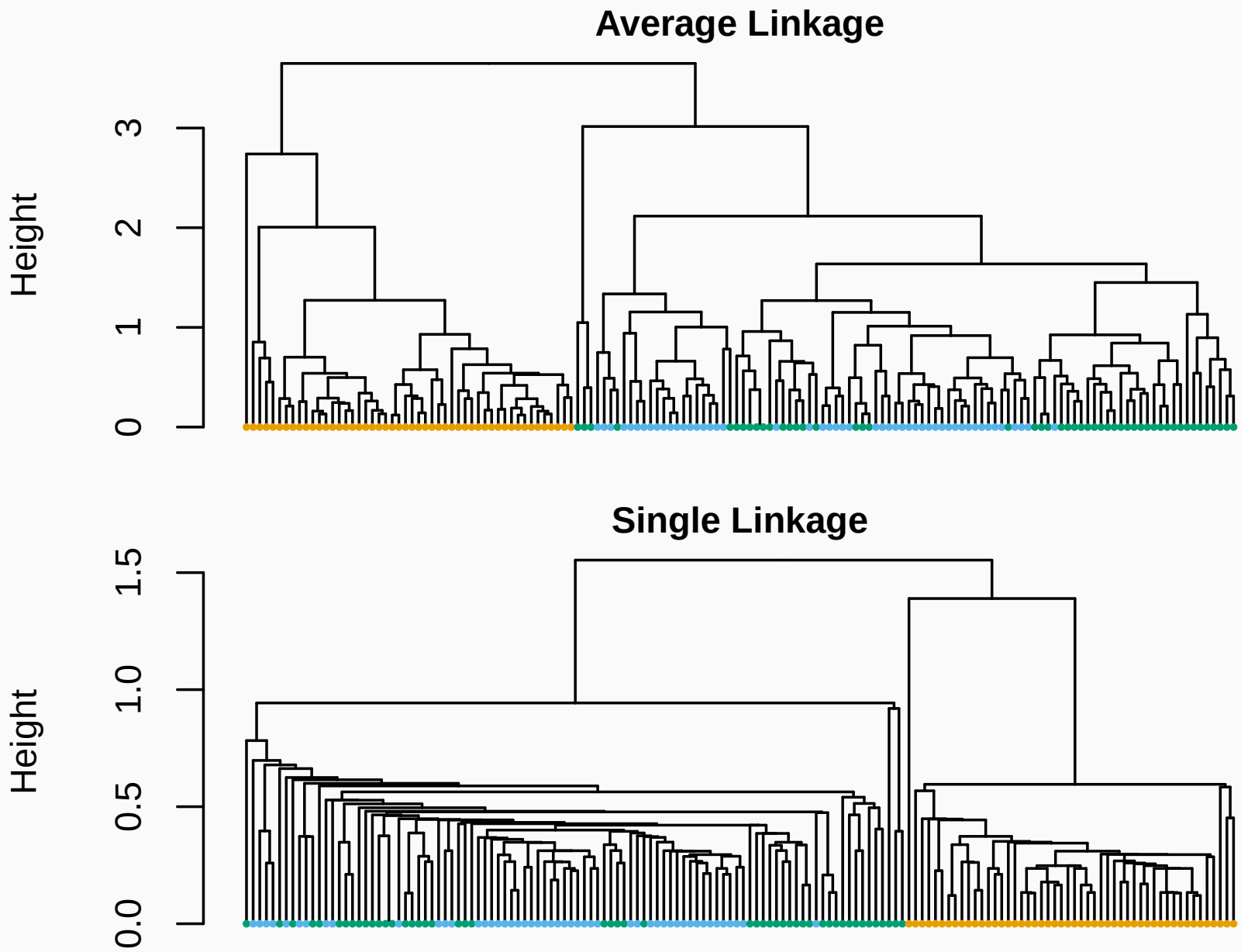
Dendrograms

Hierarchical clustering applied to **iris dataset**



- Leaf colours represent iris type: setosa, versicolor and virginica
- **Height** is the distance between clusters
- The tree can be **cut** at a certain height to achieve a final clustering. Long branches mean large increase in within cluster scatter at join

Dendrograms for other linkages



Model-based clustering

Model-based clustering

- All methods discussed so far were **non-parametric clustering methods** based on
  1. a distance/dissimilarity measure
  2. a construction algorithm
- Performance depends on **subjective choices** such as the metric, but we also have **flexibility**
- Assuming an underlying theoretical model for the feature space worked well in classification (LDA, QDA, logistic regression). **Is this transferable to clustering?**

Remember QDA

In Quadratic Discriminant Analysis (QDA) we assumed

$$p(\mathbf{x}|i) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad \text{and} \quad p(i) = \pi_i$$

This is known as a **Gaussian Mixture Model (GMM)** for $\mathbf{x}$ where

$$p(\mathbf{x}) = \sum_{i=1}^{K} p(i)p(\mathbf{x}|i) = \sum_{i=1}^{K} \pi_i N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

QDA used that the classes i_l and feature vectors $\mathbf{x}_l$ of the observations were known to calculate π_i, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$.

What if we only know the features $\mathbf{x}_l$?

Maximum Likelihood for GMMs?

The log-likelihood for the data $\mathbf{X} \in \mathbb{R}^{n \times p}$ and all unknowns

$$\theta = (\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)$$

is

$$\log(p(\mathbf{X}|\theta)) = \sum_{l=1}^{n} \log \left(\sum_{i=1}^{K} \pi_i N\left(\mathbf{x}_l | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \right)$$

Taking the gradient (with chain-rule) and solving for some $\boldsymbol{\mu}_i$ gives

$$\boldsymbol{\mu}_i = \frac{\sum_{l=1}^{n} \eta_{li} \mathbf{x}_l}{\sum_{l=1}^{n} \eta_{li}} \quad \text{where} \quad \eta_{li} = \frac{\pi_i N(\mathbf{x}_l | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{j=1}^{K} \pi_j N(\mathbf{x}_l | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Note: There is a **cyclical dependence** between η_{li} and $\boldsymbol{\mu}_i$.

What now? Thursday's lecture

Take-home message

- Selection of appropriate cluster count through
  - Elbow-method: Reduction in $W(C)$
  - Maximal average silhouette width
  - Minimal cluster prediction error
- Hierarchical clustering and its linkage-methods allow for a different non-parametric approach with visual output (dendrogram)
- Model-based clustering is more involved than model-based classification