# Lecture 6b: Clustering (cont'd)

Rebecka Jörnsten, Mathematical Sciences

**MSA220/MVE441** Statistical Learning for Big Data

1nd April 2021

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

# Bottom-up approach to clustering

# Two approaches to combinatorial clustering

## Top-down approach

- Start with all observations in one group and split them into clusters
- Examples: k-means and k-medoids

## Bottom-up approach

- Start with all observations individually and join them together to build clusters

# A bottom-up approach

Let $g_l^i$ be the set of samples in cluster $l$ at iteration $i$.

## Hierarchical clustering

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

# A bottom-up approach

Let $g_l^i$ be the set of samples in cluster $l$ at iteration $i$.

## Hierarchical clustering

1. **Initialization:** Let each observation $\mathbf{x}_l$ be in its own cluster $g_l^0$ for $l = 1, \ldots, n$
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

## Questions

- How do we measure distance between clusters?
- How do we get a final clustering with a certain number of clusters?

# Linkage

Cluster-cluster distance is called **linkage**

# Linkage

Cluster-cluster distance is called **linkage**

**Distance between clusters $g$ and $h$**

Let $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a distance matrix between samples.

# Linkage

Cluster-cluster distance is called **linkage**

**Distance between clusters $g$ and $h$**

Let $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a distance matrix between samples.

1. **Average Linkage:**

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

# Linkage

Cluster-cluster distance is called **linkage**

**Distance between clusters $g$ and $h$**

Let $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a distance matrix between samples.

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**

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

## Linkage

Cluster-cluster distance is called **linkage**

**Distance between clusters $g$ and $h$**

Let $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a distance matrix between samples.

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**

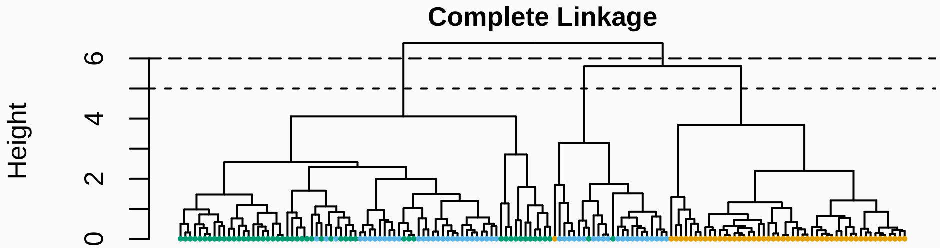
$$d(g,h) = \min_{\substack{\mathbf{x}_l \in g \\ \mathbf{x}_m \in h}} \mathbf{D}_{l,m}$$

3. **Complete Linkage**

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

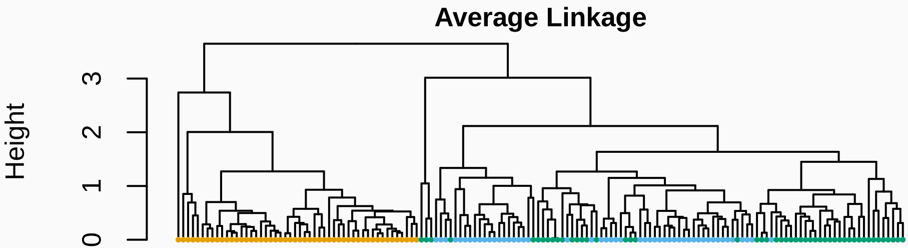
# Dendrograms

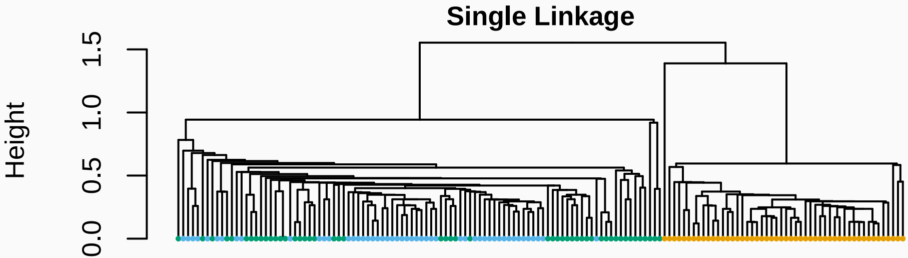
Hierarchical clustering applied to **iris dataset**




- 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



# Notes on hierarchical clustering and linkage

## Linkage criteria

- 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

# Notes on hierarchical clustering and linkage

## Linkage criteria

- 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

## New view on clustering

- Clusters are joined by closeness to each other, not by closeness to some centre
- e.g. single linkage hierarchical clustering can handle the circle around a disc example from last lecture

# 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

# 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 depended on **choices** such as the metric and how to select the cluster count

# 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 depended on **choices** such as the metric and how to select the cluster count
- 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

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

# 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 can be written 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 $\pi_i$, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$.

## Remember QDA

In Quadratic Discriminant Analysis (QDA) we assumed

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

This can be written as a **Gaussian Mixture Model (GMM)** for $\mathbf{x}$ where

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

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

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

## Maximum Likelihood for GMMs?

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

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

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 $\boldsymbol{\mu}_i$ gives

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

## Maximum Likelihood for GMMs?

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

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

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 $\boldsymbol{\mu}_i$ 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 **non-linear cyclic dependence** between $\eta_{li}$ and $\boldsymbol{\mu}_i$.

# Expectation-Maximization for GMMs

Finding the MLE for parameters $\theta$ in GMMs results in an iterative process called **Expectation-Maximization (EM)**

## Expectation-Maximization for GMMs

Finding the MLE for parameters $\theta$ in GMMs results in an iterative process called **Expectation-Maximization (EM)**

1. Initialize $\theta$

# Expectation-Maximization for GMMs

Finding the MLE for parameters $\theta$ in GMMs results in an iterative process called **Expectation-Maximization (EM)**

1. Initialize $\theta$
2. **E-Step:** Update

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

# Expectation-Maximization for GMMs

Finding the MLE for parameters $\theta$ in GMMs results in an iterative process called **Expectation-Maximization (EM)**

1. Initialize $\theta$
2. **E-Step:** Update

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

3. **M-Step:** Update

$$\boldsymbol{\mu}_i = \frac{\sum_{l=1}^{n} \eta_{li} \mathbf{x}_l}{\sum_{l=1}^{n} \eta_{li}} \qquad \pi_i = \frac{\sum_{l=1}^{n} \eta_{li}}{n}$$

$$\boldsymbol{\Sigma}_i = \frac{1}{\sum_{l=1}^{n} \eta_{li}} \sum_{l=1}^{n} \eta_{li} (\mathbf{x}_l - \boldsymbol{\mu}_i)(\mathbf{x}_l - \boldsymbol{\mu}_i)^\top$$

## Expectation-Maximization for GMMs

Finding the MLE for parameters $\theta$ in GMMs results in an iterative process called **Expectation-Maximization (EM)**

1. Initialize $\theta$
2. **E-Step:** Update

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

3. **M-Step:** Update

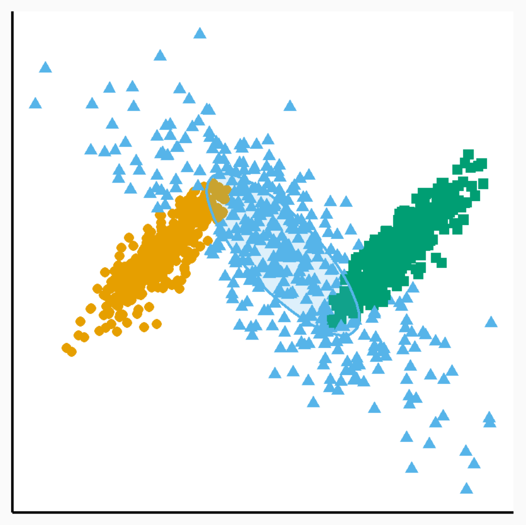
$$\boldsymbol{\mu}_i = \frac{\sum_{l=1}^{n} \eta_{li} \mathbf{x}_l}{\sum_{l=1}^{n} \eta_{li}} \qquad \pi_i = \frac{\sum_{l=1}^{n} \eta_{li}}{n}$$

$$\boldsymbol{\Sigma}_i = \frac{1}{\sum_{l=1}^{n} \eta_{li}} \sum_{l=1}^{n} \eta_{li} (\mathbf{x}_l - \boldsymbol{\mu}_i)(\mathbf{x}_l - \boldsymbol{\mu}_i)^\top$$

4. Repeat steps 2 and 3 until convergence

# GMM clustering example

- Yellow and green clusters share a covariance matrix
- The blue cluster has a different one
- GMM clustering on only the data points without knowledge of the class labels recovers the covariance structures and clusters



# Why does Expectation-Maximization work?

# Likelihood of the complete data

- **Assume** that the **classes $i_l$ are known** and code them as $z_{lj} = 1$ if $i_l = j$ and $z_{lj} = 0$ otherwise. Collect them in $\mathbf{Z} \in \mathbb{R}^{n \times K}$.

## Likelihood of the complete data

- **Assume** that the **classes $i_l$ are known** and code them as $z_{lj} = 1$ if $i_l = j$ and $z_{lj} = 0$ otherwise. Collect them in $\mathbf{Z} \in \mathbb{R}^{n \times K}$.
- $(\mathbf{X}, \mathbf{Z})$ are called the **complete data**, and **incomplete data** when only $\mathbf{X}$ is observed

# Likelihood of the complete data

- **Assume** that the **classes $i_l$ are known** and code them as $z_{lj} = 1$ if $i_l = j$ and $z_{lj} = 0$ otherwise. Collect them in $\mathbf{Z} \in \mathbb{R}^{n \times K}$.
- $(\mathbf{X}, \mathbf{Z})$ are called the **complete data**, and **incomplete data** when only $\mathbf{X}$ is observed
- The class assignments $\mathbf{Z}$ are called **latent variables**

## Likelihood of the complete data

- **Assume** that the **classes $i_l$ are known** and code them as $z_{lj} = 1$ if $i_l = j$ and $z_{lj} = 0$ otherwise. Collect them in $\mathbf{Z} \in \mathbb{R}^{n \times K}$.
- $(\mathbf{X}, \mathbf{Z})$ are called the **complete data**, and **incomplete data** when only $\mathbf{X}$ is observed
- The class assignments $\mathbf{Z}$ are called **latent variables**
- **Complete data likelihood**

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

  and the parameters in $\theta$ are easy to estimate (QDA).

## Likelihood of the complete data

- **Assume** that the **classes $i_l$ are known** and code them as $z_{lj} = 1$ if $i_l = j$ and $z_{lj} = 0$ otherwise. Collect them in $\mathbf{Z} \in \mathbb{R}^{n \times K}$.
- $(\mathbf{X}, \mathbf{Z})$ are called the **complete data**, and **incomplete data** when only $\mathbf{X}$ is observed
- The class assignments $\mathbf{Z}$ are called **latent variables**
- **Complete data likelihood**

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

  and the parameters in $\theta$ are easy to estimate (QDA).
- **Incomplete data likelihood**

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

# Decomposing the incomplete data likelihood

- **For known $\mathbf{Z}$**

$$p(\mathbf{X}|\theta) = \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X}, \theta)}, \quad \text{i.e.}$$

$$\log p(\mathbf{X}|\theta) = \log p(\mathbf{X}, \mathbf{Z}|\theta) - \log p(\mathbf{Z}|\mathbf{X}, \theta)$$

is a **decomposition** of the log-likelihood for $\mathbf{X}$ given $\theta$

# Decomposing the incomplete data likelihood

- **For known $\mathbf{Z}$**

$$p(\mathbf{X}|\theta) = \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X}, \theta)}, \quad \text{i.e.}$$

$$\log p(\mathbf{X}|\theta) = \log p(\mathbf{X}, \mathbf{Z}|\theta) - \log p(\mathbf{Z}|\mathbf{X}, \theta)$$

  is a **decomposition** of the log-likelihood for $\mathbf{X}$ given $\theta$

- For any density $q(\mathbf{Z})$ it holds that

$$\log p(\mathbf{X}|\theta) = \log \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})} - \log \frac{p(\mathbf{Z}|\mathbf{X}, \theta)}{q(\mathbf{Z})}$$

## Decomposing the incomplete data likelihood

- **For known $\mathbf{Z}$**

$$p(\mathbf{X}|\theta) = \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X}, \theta)}, \quad \text{i.e.}$$

$$\log p(\mathbf{X}|\theta) = \log p(\mathbf{X}, \mathbf{Z}|\theta) - \log p(\mathbf{Z}|\mathbf{X}, \theta)$$

  is a **decomposition** of the log-likelihood for $\mathbf{X}$ given $\theta$

- For any density $q(\mathbf{Z})$ it holds that

$$\log p(\mathbf{X}|\theta) = \log \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})} - \log \frac{p(\mathbf{Z}|\mathbf{X}, \theta)}{q(\mathbf{Z})}$$

- **Average** over $\mathbf{Z}$ according to the density $q(\mathbf{Z})$

$$\begin{aligned}\log(p(\mathbf{X}|\theta)) &= \mathbb{E}_{q(\mathbf{Z})}\left[\log \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{q(\mathbf{Z})}\right] - \mathbb{E}_{q(\mathbf{Z})}\left[\log \frac{p(\mathbf{Z}|\mathbf{X}, \theta)}{q(\mathbf{Z})}\right] \\ &=: F(q, \theta) + \mathrm{KL}(q||p(\cdot|\mathbf{X}, \theta))\end{aligned}$$

  where $\mathrm{KL}(q||p(\cdot|\mathbf{X}, \theta))$ is called the **Kullback-Leibler (KL) divergence** of $q(\mathbf{Z})$ and $p(\cdot|\mathbf{X}, \theta)$.

# Decomposing the incomplete data likelihood (II)

It can be shown (using **Jensen's inequality**) that

$$\mathrm{KL}(q||p(\cdot|\mathbf{X},\theta)) = -\mathbb{E}_{q(\mathbf{Z})}\left[\log \frac{p(\mathbf{Z}|\mathbf{X},\theta)}{q(\mathbf{Z})}\right] \geq 0$$

with equality if $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X},\theta)$.

This implies that

$$\log p(\mathbf{X}|\theta) \geq F(q,\theta)$$

is a **lower bound** which is tight (i.e. equality holds) if $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X},\theta)$.

This gives us a **recipe** on how to choose $q(\mathbf{Z})$.

# Expectation-Maximization

1. **Expectation step:** For given parameters $\theta^{(m)}$ the density $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})$ ensures that $F(q, \theta^{(m)}) = \log p(\mathbf{X}|\theta^{(m)})$. Note that then

$$\begin{aligned} F(q, \theta) &= \mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})}\left[\log p(\mathbf{X}, \mathbf{Z}|\theta)\right] - \mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})}\left[\log p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})\right] \\ &=: Q(\theta, \theta^{(m)}) + \text{constant} \end{aligned}$$

2. **Maximization step:** Maximize $F(q, \theta)$ through

$$\theta^{(m+1)} = \arg\max_{\theta} Q(\theta, \theta^{(m)})$$

## Expectation-Maximization

1. **Expectation step:** For given parameters $\theta^{(m)}$ the density $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})$ ensures that $F(q, \theta^{(m)}) = \log p(\mathbf{X}|\theta^{(m)})$. Note that then

$$F(q, \theta) = \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\theta^{(m)})} [\log p(\mathbf{X}, \mathbf{Z}|\theta)] - \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\theta^{(m)})} \left[\log p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})\right]$$
$$=: Q(\theta, \theta^{(m)}) + \text{constant}$$

2. **Maximization step:** Maximize $F(q, \theta)$ through

$$\theta^{(m+1)} = \arg\max_{\theta} Q(\theta, \theta^{(m)})$$

The incomplete data likelihood increases in each step until convergence to a **local maximum**.

# How to use the EM algorithm?

### Two step procedure

1. Compute for given $\theta^{(m)}$

$$q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \theta^{(m)}).$$

2. Maximize in $\theta$

$$Q(\theta, \theta^{(m)}) = \mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})}\left[\log p(\mathbf{X}, \mathbf{Z}|\theta)\right]$$

# Applying EM to the GMM clustering problem (I)

## Expectation step

Given $\mathbf{X}$ and $\theta^{(m)}$

$$p(\mathbf{Z}|\mathbf{X},\theta^{(m)}) = \frac{p(\mathbf{X},\mathbf{Z}|\theta^{(m)})}{p(\mathbf{X}|\theta^{(m)})} = \prod_{l=1}^{n} \frac{\prod_{i=1}^{K}(\pi_i N(\mathbf{x}_l|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i))^{z_{li}}}{\sum_{j=1}^{K}\pi_j N\left(\mathbf{x}_l|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\right)}$$

and recall that

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

To compute $Q(\theta,\theta^{(m)})$ we only need to compute

$$\mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\theta^{(m)})}[z_{li}] = \frac{\pi_i N(\mathbf{x}_l|\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)}{\sum_{j=1}^{K}\pi_j N(\mathbf{x}_l|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)} = \eta_{li}$$

the so-called **responsibility** of class $i$ for having generated the observation $\mathbf{x}_l$.

# Applying EM to the GMM clustering problem (II)

## Maximization step

This results in

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

which is maximized by the MLE estimates

$$\boldsymbol{\mu}_i = \frac{\sum_{l=1}^{n} \eta_{li} \mathbf{x}_l}{\sum_{l=1}^{n} \eta_{li}} \qquad \pi_i = \frac{\sum_{l=1}^{n} \eta_{li}}{n}$$

$$\boldsymbol{\Sigma}_i = \frac{1}{\sum_{l=1}^{n} \eta_{li}} \sum_{l=1}^{n} \eta_{li} (\mathbf{x}_l - \boldsymbol{\mu}_i)(\mathbf{x}_l - \boldsymbol{\mu}_i)^\top$$

# Cluster selection

A **final clustering** can be selected with

$$C(\mathbf{x}_l) = \arg\max_i \eta_{li}$$

or responsibilities can be used as a **soft clustering**

# Cluster selection

A **final clustering** can be selected with

$$C(\mathbf{x}_l) = \arg\max_i \eta_{li}$$

or responsibilities can be used as a **soft clustering**

## Cluster count selection

Model selection criteria for MLE can be used, e.g. minimal **Bayesian Information Criterion (BIC)**

$$\begin{aligned} \mathrm{BIC}(K) = & -2\log(p(\mathbf{X}|\theta, K)) \\ & + \log(n) \cdot \underbrace{[(K-1) + K \cdot p + K \cdot \frac{p(p+1)}{2}]}_{\text{number of model parameters}} \end{aligned}$$

which is valid for large $n$.

# Caveat with MLE for GMMs

- Centering one mixture component on an observation (i.e. $\boldsymbol{\mu}_i = \mathbf{x}_l$ for some $i$ and $l$) and letting its variance go to zero can drive the likelihood to infinity

## Caveat with MLE for GMMs

- Centering one mixture component on an observation (i.e. $\boldsymbol{\mu}_i = \mathbf{x}_l$ for some $i$ and $l$) and letting its variance go to zero can drive the likelihood to infinity
  - 'Outside of scope'-solution:
    Bayesian framework and Inverse-Wishart prior on $\boldsymbol{\Sigma}_i$

## Caveat with MLE for GMMs

- Centering one mixture component on an observation (i.e. $\boldsymbol{\mu}_i = \mathbf{x}_l$ for some $i$ and $l$) and letting its variance go to zero can drive the likelihood to infinity
  - 'Outside of scope'-solution:
    Bayesian framework and Inverse-Wishart prior on $\boldsymbol{\Sigma}_i$
  - Initialize $\boldsymbol{\Sigma}_i$ with large enough variances and potentially restart if bad convergence

# Caveat with MLE for GMMs

- Centering one mixture component on an observation (i.e. $\boldsymbol{\mu}_i = \mathbf{x}_l$ for some $i$ and $l$) and letting its variance go to zero can drive the likelihood to infinity
  - 'Outside of scope'-solution:
    Bayesian framework and Inverse-Wishart prior on $\boldsymbol{\Sigma}_i$
  - Initialize $\boldsymbol{\Sigma}_i$ with large enough variances and potentially restart if bad convergence
- Like k-means, this algorithm is sensitive to starting values

# GMMs and EM for classification

## GMM for classification

In QDA $p(\mathbf{x}|i) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ capture classes with **elliptic shape**.

## GMM for classification

In QDA $p(\mathbf{x}|i) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ capture classes with **elliptic shape**.

Assume features are described by a GMM, i.e.

$$p(\mathbf{x}|i) = \sum_{m=1}^{M_i} \pi_{im} N(\mathbf{x}|\boldsymbol{\mu}_{im}, \boldsymbol{\Sigma})$$

where

## GMM for classification

In QDA $p(\mathbf{x}|i) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ capture classes with **elliptic shape**.

Assume features are described by a GMM, i.e.

$$p(\mathbf{x}|i) = \sum_{m=1}^{M_i} \pi_{im} N(\mathbf{x}|\boldsymbol{\mu}_{im}, \boldsymbol{\Sigma})$$

where

- $M_i$ components for class $i$

## GMM for classification

In QDA $p(\mathbf{x}|i) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ capture classes with **elliptic shape**.

Assume features are described by a GMM, i.e.

$$p(\mathbf{x}|i) = \sum_{m=1}^{M_i} \pi_{im} N(\mathbf{x}|\boldsymbol{\mu}_{im}, \boldsymbol{\Sigma})$$

where

- $M_i$ components for class $i$
- $\pi_{im}$ is the probability of mixture component $m$ for class $i$

## GMM for classification

In QDA $p(\mathbf{x}|i) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ capture classes with **elliptic shape**.

Assume features are described by a GMM, i.e.

$$p(\mathbf{x}|i) = \sum_{m=1}^{M_i} \pi_{im} N(\mathbf{x}|\boldsymbol{\mu}_{im}, \boldsymbol{\Sigma})$$

where

- $M_i$ components for class $i$
- $\pi_{im}$ is the probability of mixture component $m$ for class $i$
- Covariance matrix $\boldsymbol{\Sigma}$ is assumed to be constant across mixture components and classes

## GMM for classification

In QDA $p(\mathbf{x}|i) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ capture classes with **elliptic shape**.

Assume features are described by a GMM, i.e.

$$p(\mathbf{x}|i) = \sum_{m=1}^{M_i} \pi_{im} N(\mathbf{x}|\boldsymbol{\mu}_{im}, \boldsymbol{\Sigma})$$

where

- $M_i$ components for class $i$
- $\pi_{im}$ is the probability of mixture component $m$ for class $i$
- Covariance matrix $\boldsymbol{\Sigma}$ is assumed to be constant across mixture components and classes

**Component membership** $z_{lm}$ is a latent variable for the observation $(\mathbf{x}_l, i_l)$ with $z_{lm} = 1$ if $\mathbf{x}_l$ is in component $m \in \{1, \dots, M_{i_l}\}$ and $z_{lm} = 0$ otherwise

# Mixture DA

Finding the MLE for the mixture DA parameters can be achieved through **Expectation-Maximization (EM)**

## Mixture DA

Finding the MLE for the mixture DA parameters can be achieved through **Expectation-Maximization (EM)**

1. Initialize $\theta$

## Mixture DA

Finding the MLE for the mixture DA parameters can be achieved through **Expectation-Maximization (EM)**

1. Initialize $\theta$
2. **E-Step:** Update

$$\eta_{lm} = \frac{\pi_{i_l m} N(\mathbf{x}_l | \boldsymbol{\mu}_{i_l m}, \boldsymbol{\Sigma})}{\sum_{j=1}^{M_{i_l}} \pi_{i_l j} N(\mathbf{x}_l | \boldsymbol{\mu}_{i_l j}, \boldsymbol{\Sigma})}$$

## Mixture DA

Finding the MLE for the mixture DA parameters can be achieved through **Expectation-Maximization (EM)**

1. Initialize $\theta$
2. **E-Step:** Update

$$\eta_{lm} = \frac{\pi_{i_l m} N(\mathbf{x}_l | \boldsymbol{\mu}_{i_l m}, \boldsymbol{\Sigma})}{\sum_{j=1}^{M_{i_l}} \pi_{i_l j} N(\mathbf{x}_l | \boldsymbol{\mu}_{i_l j}, \boldsymbol{\Sigma})}$$

3. **M-Step:** Update

$$\boldsymbol{\mu}_{im} = \frac{\sum_{i_l = i} \eta_{lm} \mathbf{x}_l}{\sum_{i_l = i} \eta_{lm}} \qquad \pi_{im} = \frac{\sum_{i_l = i} \eta_{lm}}{n_i}$$

$$\boldsymbol{\Sigma} = \frac{1}{n} \sum_{i=1}^{K} \sum_{i_l = i} \sum_{m=1}^{M_i} \eta_{lm} (\mathbf{x}_l - \boldsymbol{\mu}_{im})(\mathbf{x}_l - \boldsymbol{\mu}_{im})^\top$$

## Mixture DA

Finding the MLE for the mixture DA parameters can be achieved through **Expectation-Maximization (EM)**

1. Initialize $\theta$
2. **E-Step:** Update

$$\eta_{lm} = \frac{\pi_{i_l m} N(\mathbf{x}_l | \boldsymbol{\mu}_{i_l m}, \boldsymbol{\Sigma})}{\sum_{j=1}^{M_{i_l}} \pi_{i_l j} N(\mathbf{x}_l | \boldsymbol{\mu}_{i_l j}, \boldsymbol{\Sigma})}$$

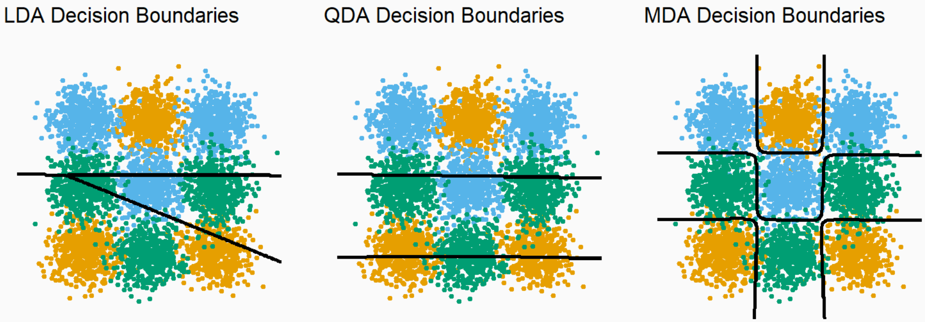
3. **M-Step:** Update

$$\boldsymbol{\mu}_{im} = \frac{\sum_{i_l = i} \eta_{lm} \mathbf{x}_l}{\sum_{i_l = i} \eta_{lm}} \qquad \pi_{im} = \frac{\sum_{i_l = i} \eta_{lm}}{n_i}$$

$$\boldsymbol{\Sigma} = \frac{1}{n} \sum_{i=1}^{K} \sum_{i_l = i} \sum_{m=1}^{M_i} \eta_{lm} (\mathbf{x}_l - \boldsymbol{\mu}_{im})(\mathbf{x}_l - \boldsymbol{\mu}_{im})^\top$$

4. Repeat steps 2 and 3 until convergence

# MDA example


LDA Decision Boundaries
QDA Decision Boundaries
MDA Decision Boundaries

## Take-home message

- Hierarchical clustering and its linkage-methods allow for a different non-parametric approach with visual output (dendrogram)
- Expectation-Maximization allows us to perform model-based clustering
- Both clustering and classification methods profit from using Gaussian Mixture Models