Lecture 8: Model- and density-based clustering

Felix Held, Mathematical Sciences

MSA220/MVE440 Statistical Learning for Big Data

11th April 2019

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Model-based 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

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

is

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

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

Expectation-Maximization for GMMs

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

1. Initialize θ
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

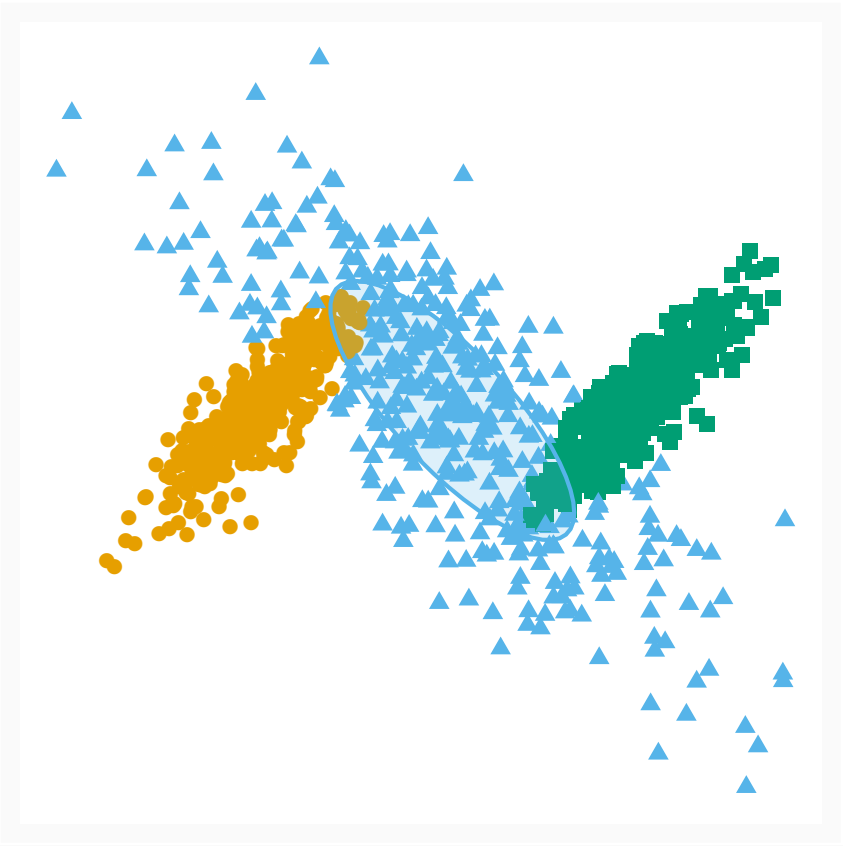
$$\boldsymbol{\mu}_i = \frac{\sum_{l=1}^{n} \eta_{li} \mathbf{x}_l}{\sum_{l=1}^{n} \eta_{li}} \qquad \pi_i = \frac{\sum_{l=1}^{n} \eta_{li}}{n}$$

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

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
- GGM 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 $\mathbf{Z} \in \mathbb{R}^{n \times K}$.
- $(\mathbf{X}, \mathbf{Z})$ 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 θ 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

- **If we knew $\mathbf{Z}$** then

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

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

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

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

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

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

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

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

$$\mathbb{E}_{q(\mathbf{Z})}\left[\log\frac{p(\mathbf{Z}|\mathbf{X},\theta)}{q(\mathbf{Z})}\right] \leq 0$$

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

Expectation-Maximization

New target function: Maximize

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

with respect to $q(\mathbf{Z})$ and θ

1. **Expectation step:** For given parameters $\theta^{(m)}$ the density $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X},\theta^{(m)})$ minimizes the second term and thereby maximizes the first one. Set

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

2. **Maximization step:** Maximize the first term with

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

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

Applying EM to the GMM clustering problem

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

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

and it holds that

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

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

This results in

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

which is maximized by estimates seen earlier weighted by η_{li}.

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)**

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

where n is much larger than the number of model parameters

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 $p(\mathbf{x}|i) = N(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ can only capture **elliptic class shapes**.

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
- π_{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)**

1. Initialize θ
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

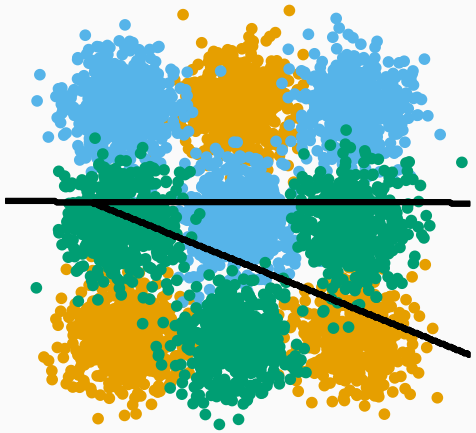
$$\boldsymbol{\mu}_{im} = \frac{\sum_{i_l = i} \eta_{lm} \mathbf{x}_l}{\sum_{i_l = i} \eta_{lm}} \qquad \pi_{im} = \frac{\sum_{i_l = i} \eta_{lm}}{n_i}$$

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

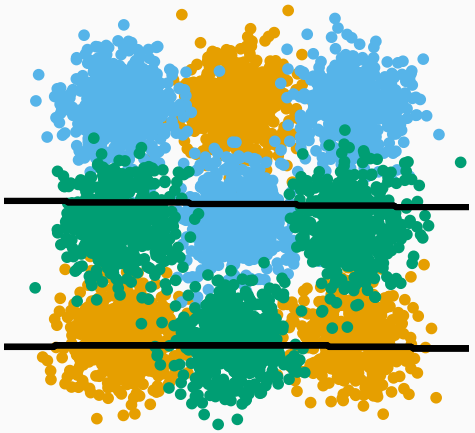
4. Repeat steps 2 and 3 until convergence

MDA example

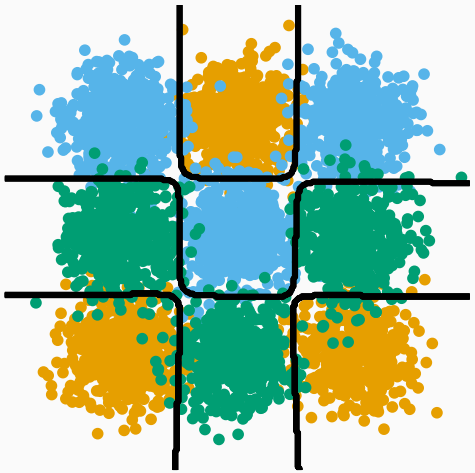
LDA Decision Boundaries



QDA Decision Boundaries



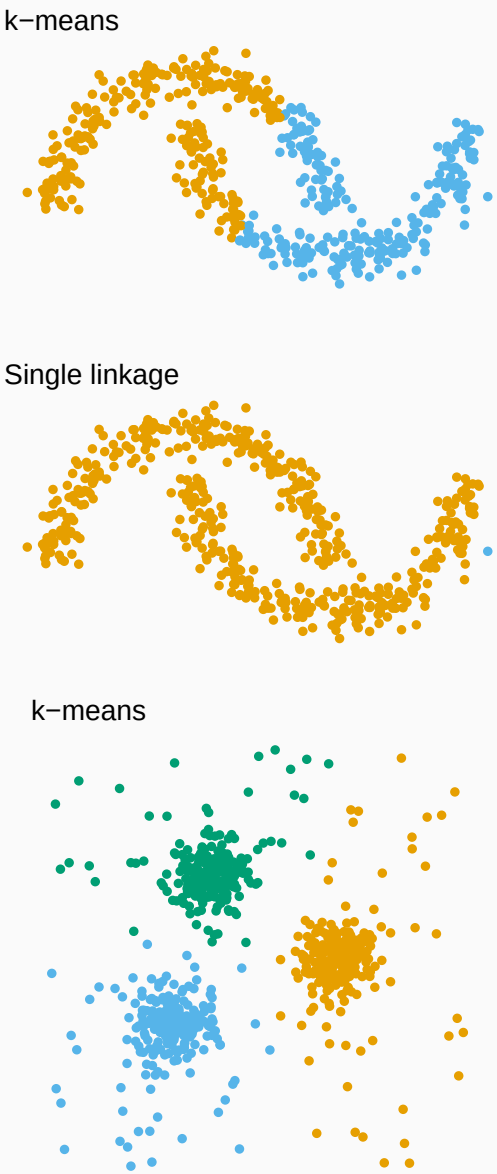
MDA Decision Boundaries



Density-based clustering

Yet another approach to clustering

- Most methods discussed so far have problems with odd, non-convex shapes
- What about **noise**? Some observations might not fit into any cluster
- **New cluster definition:** Clusters are dense regions in feature space
  - What is dense?
  - How to find groups and separate the noise?
- **Naive approach:** Find points surrounded by many other points and connect them to a cluster. Points that do not end up in a cluster are noise.



Notation in density-based clustering

The presented methodology has **two tuning parameters** $\varepsilon > 0$ and $n_{\min} \in \mathbb{N}$. Assume each observation is a **point p in a database/dataset D** and there is a **distance measure** $d(p, q)$.

- **ε-neighbourhood of p:** $N_\varepsilon(p) = \{q \in D | d(p, q) \leq \varepsilon\}$
- **Core point:** A $p \in D$ s.th. $|N_\varepsilon(p)| \geq n_{\min}$
- p is **directly density-reachable** from a core-point q if $p \in N_\varepsilon(q)$
- p is **density-reachable** from a core-point q if there is a chain $q = p_1, p_2, \ldots, p_m = p$ s.th. p_{i+1} is directly density-reachable from p_i
- p and q are **density-connected** if there is a core-point o s.th. p and q are density-reachable from o

Density-based clusters

A **cluster** C is a set of points in D s.th.

1. If $p \in C$ and q is density-reachable from p then $q \in C$ (**maximality**)
2. For all $p, q \in C$: p and q are density-connected (**connectivity**)

This leads to **three types of points**

1. **Core points:** Part of a cluster and at least $n_{\min}$ points in neighbourhood
2. **Border points:** Part of a cluster but not core points
3. **Noise:** Not part of any cluster

Note: Border points can have **non-unique cluster assignments**

DBSCAN algorithm

Computational procedure:

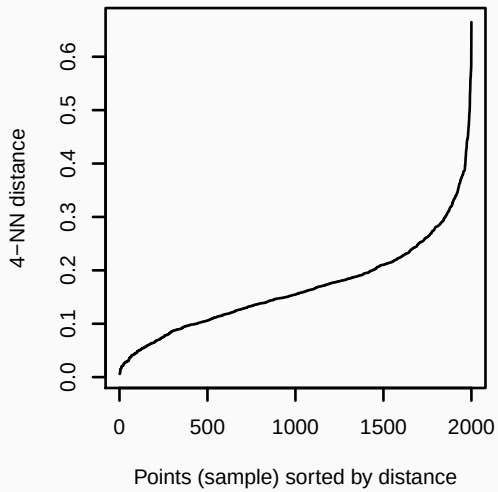
1. Go through each point p in the dataset D
2. If it has already been processed take the next one
3. Else determine its ε-neighbourhood. If less than $n_{\min}$ points in neighbourhood, label as noise. Otherwise, start a new cluster.
4. Find all points that are density-reachable from p and add them to the cluster.

Dependence on $n_{\min}$

- Controls how easy it is to connect components in a cluster
  - Too small and most points are core points, creating many small clusters
  - Too large and few points are core points, leading to many noise labelled observations
- A cluster has by definition at least $n_{\min}$ points
- Choice of $n_{\min}$ is very dataset dependent
- Tricky in high-dimensional data (**curse of dimensionality**, everything is far apart)

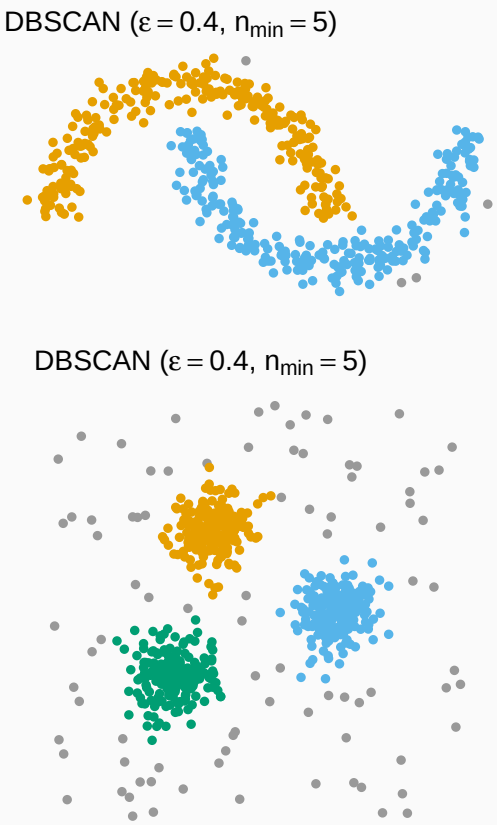
Dependence on ε

- Controls how much of the data will be clustered
  - Too small and small gaps in clusters cannot be bridged, leading to isolated islands in the data
  - Too large and everything is connected
- Choice of ε is also dataset dependent but there is a **decision tool**
  - Determine distance to the k nearest neighbours for each point in the dataset
  - Inside clusters, increasing k should not lead to a large increase of d
  - The optimal ε is supposed to be roughly at the knee



DBSCAN example

- DBSCAN is able to cluster points in the situations advertised and correctly identifies noise points
- Very sensitive to the choice of tuning parameters



Take-home message

- Expectation-Maximization allows maximum likelihood estimation even in situation where additional data would be necessary
- Both clustering and classification methods profit from using Gaussian Mixture Models
- Density-based clustering allows to capture complex shapes and the identification of noise during clustering