Lecture 8: Clustering Part II

Statistical Methods for Data Science

Yinan Yu

Department of Computer Science and Engineering

November 28, 2022

Today

Learning outcome

- Be able to explain the difference between Gaussian naive Bayes classifier and GMM in terms of parameter estimation
- Be able to explain the objective function $Q(\theta)$ for GMM
- Understand what EM algorithm is used for and why we need it
- Be able to calculate AIC/BIC and use them to determine K for GMM
- Be able to explain the EM algorithm for one dimensional GMM
- Be able to explain the difference between K-means and the EM algorithm in terms of their assumptions and parameter estimation

Today

Gaussian Mixture Models (GMM)

Four categories

Clustering models

- **Centroid clustering** (lecture 7)
- **Distribution clustering**
- Density clustering
- Hierarchical clustering

Gaussian Mixture Models (GMM) - overview

Distribution clustering:

- Each cluster is **modeled** using a probability distribution
- Each data point is **modeled** using a "combination of all clusters"

Gaussian Mixture Models:

- **Data $\boldsymbol{x}$**: a d dimensional feature vector $\boldsymbol{x} = [x_1, \cdots, x_d]$

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f(\boldsymbol{x} \mid k)$$

where

- $f(\boldsymbol{x})$ is the PDF of $\boldsymbol{x}$ ("multivariate");
- $\pi_k = P(k) > 0$ and $\sum_{k=1}^{K} \pi_k = 1$;
- $f(\boldsymbol{x} \mid k)$ is a d **dimensional multivariate Gaussian PDF** describing cluster k.

Gaussian Mixture Models (GMM) - overview (cont.)

- Gaussian Mixture Model:

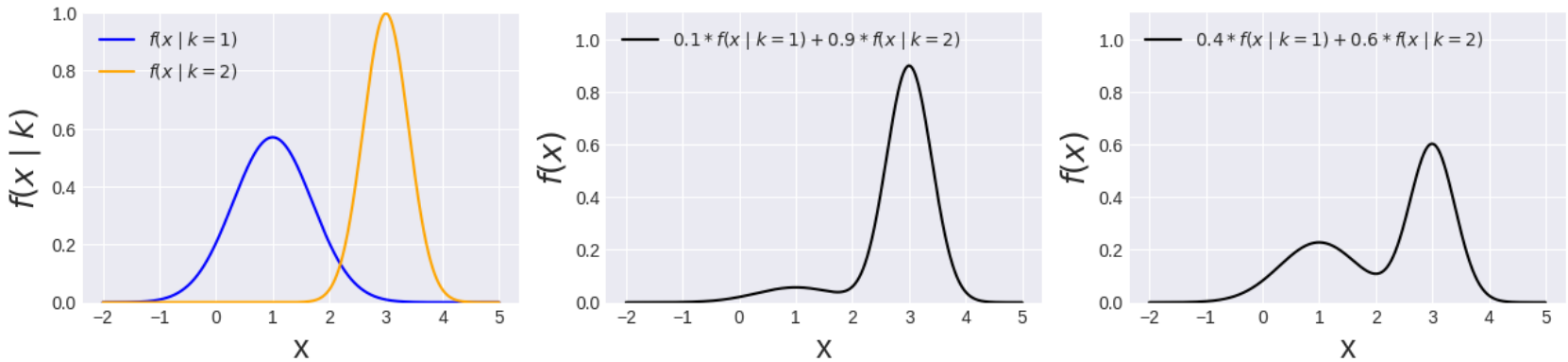
$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f(\boldsymbol{x} \mid k)$$

- Examples of the mixture distribution with $d = 1$
  - Example 1: $\pi_1 = 0.1$, $\pi_2 = 0.9$

$$f(\boldsymbol{x}) = 0.1 \times f(x \mid k = 1) + 0.9 \times f(x \mid k = 2) = 0.1 \times f(x \mid \mu_1, \sigma_1) + 0.9 \times f(x \mid \mu_2, \sigma_2)$$

  - Example 2: $\pi_1 = 0.4$, $\pi_2 = 0.6$

$$f(\boldsymbol{x}) = 0.4 \times f(x \mid k = 1) + 0.6 \times f(x \mid k = 2) = 0.4 \times f(x \mid \mu_1, \sigma_1) + 0.6 \times f(x \mid \mu_2, \sigma_2)$$

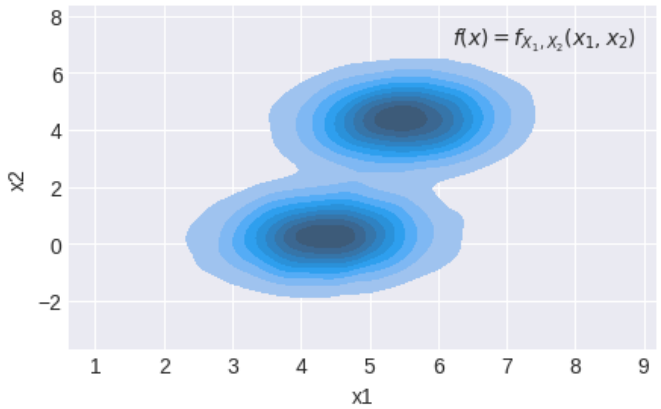


Gaussian Mixture Models (GMM) - overview (cont.)

- Examples of the mixture distribution with $d = 2$: $\pi_1 = 0.5$, $\pi_2 = 0.5$

  $$f(\boldsymbol{x}) = 0.5 \times f(\boldsymbol{x} \mid k = 1) + 0.5 \times f(\boldsymbol{x} \mid k = 2) = 0.5 \times f(\boldsymbol{x} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + 0.5 \times f(\boldsymbol{x} \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

  where $\boldsymbol{\mu}_k \in \mathbb{R}^2$ is the mean and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{2\times 2}$ is the covariance matrix



Gaussian Mixture Models (GMM) - overview (cont.)

- **Data $\boldsymbol{x}$**: a d dimensional feature vector $\boldsymbol{x} = [x_1, \cdots, x_d]$ with PDF $f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f(\boldsymbol{x} \mid k)$
- **Target y**: y is a set of K posterior probabilities; for $k = 1, \cdots, K$

$$\overbrace{P(k \mid \boldsymbol{x})}^{\textit{posterior}} = \frac{\overbrace{P(k)}^{\textit{prior}} \quad \overbrace{f(\boldsymbol{x} \mid k)}^{\textit{likelihood of } k \textit{ given data}}}{\sum_{c=1}^{K} P(c) f(\boldsymbol{x} \mid c)}$$

It is **soft clustering** - $\boldsymbol{x}$ is assigned to **all clusters** with a **probability** - the **posterior** $P(k \mid \boldsymbol{x})$
Alternatively, y can be defined as the **cluster index** with the **highest posterior probability**, i.e.

$$y = \arg \max_{k \in \{1, \cdots, K\}} P(k \mid \boldsymbol{x}) = \arg \max_{k \in \{1, \cdots, K\}} P(k) f(\boldsymbol{x} \mid k)$$

- **Parameter**:
  - The parameters for each Gaussian likelihood $f(\boldsymbol{x} \mid k)$
  - The prior $P(k)$, typically denoted as π_k

Parameter estimation: the EM algorithm

Parameter estimation for GMM

- **Parameter estimation**
  - What's special about this? We know how to do it! It's almost the same as the **Gaussian naive Bayes classifier!** ...which you just struggled a lot with... :D
  - Let's discuss the key differences between these two algorithms
  - **Set up**: given a data set $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$, we need to estimate the parameter of interest from $\mathcal{X}$

	Gaussian naive Bayes classifier	Gaussian Mixture Models
Parameter of interest	$P(k)$, Gaussian PDF $f(\boldsymbol{x} \mid k)$, for $k = 1, \cdots, K$	
Training data (labels) available?	Yes	No
Probabilistic model	One Gaussian for each $\boldsymbol{x}_i$	A linear combination of K Gaussians for each $\boldsymbol{x}_i$
Interpretation	One label for each $\boldsymbol{x}_i$ (hard assignment)	K probabilities for each $\boldsymbol{x}_i$ (soft assignment)
Assumption	$\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ independent for $i \neq j$	
	x_m^i and x_n^i independent for dimensions $m \neq n$ (NAIVE!)	x_m^i and x_n^i NOT necessarily independent for dimensions $m \neq n$

Note: the subscripts here are the indices for the dimensions of the feature space; they are not the indices for the data points - data points are still independent!

Parameter estimation for GMM (cont.)

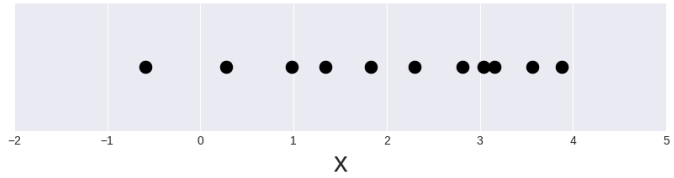
In summary, we have the following additional challenges compared to the Gaussian naive Bayes classifier:

1. We **do not have the labels** - we cannot easily estimate $P(k)$ and $f(\boldsymbol{x} \mid k)$
2. The distribution $f(\boldsymbol{x} \mid k)$ is a **multivariate Gaussian PDF** and the features are **not necessarily independent** - now we need to explicitly work with **joint probability distributions** $f_{X_1,\cdots,X_d}(x_1, \cdots, x_d \mid k)$ and **covariance matrices** (ugh!!)

Let's focus on the first issue by working with **one dimensional** feature vectors so we don't get overwhelmed by dealing with all the problems at once
Note: in this lecture, sometimes we simply use θ to denote the estimate (instead of $\hat{\theta}$) in order to reduce clutter since the notations are already quite complex

Parameter estimation for one dimensional GMM

- **Data**: $x_1, \cdots, x_N$



- **Random variable**: $X_1, \cdots, X_N$ i.i.d. with **PDF**

$$f(x) = \sum_{k=1}^{K} \pi_k f(x \mid k)$$

where $f(x \mid k)$ is a Gaussian PDF.
The joint probability distribution of all data points is defined as

$$f_{X_1,\cdots,X_N}(x_1, \cdots, x_N) \overset{\text{i.i.d.}}{=} \prod_{i=1}^{N} f(x_i) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k f(x_i \mid k) \tag{1}$$

This is the **likelihood** $L(\theta \mid x_1, \cdots, x_N)$ of the **mixture distribution** given data $x_1, \cdots, x_N$

- **Parameter of interest**: π_k (prior), μ_k, σ_k, for all $k = 1, \cdots, K$
- **Parameter estimation method**: maximum likelihood estimation

Parameter estimation for one dimensional GMM (cont.)

- The **log likelihood** (cf. Eq. (1) on page 14) is

$$\begin{aligned} Q(\theta) &= \log L\big(\theta \mid x_1, \cdots, x_N\big) = \log f_{X_1,\cdots,X_N}(x_1, \cdots, x_N) \\ &= \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k f(x_i \mid k) \right) \end{aligned} \tag{2}$$

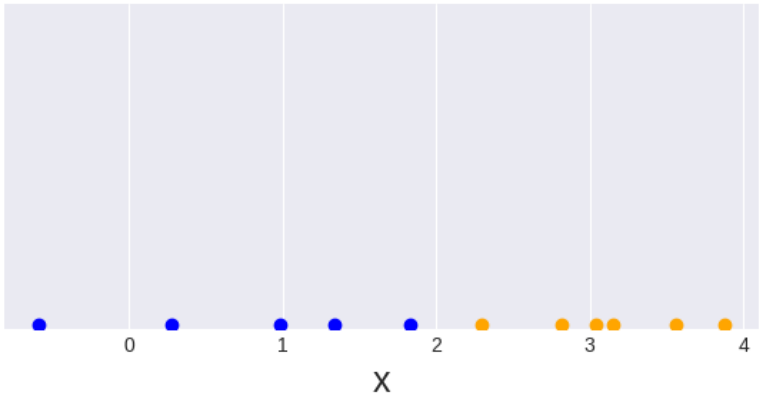
  where $\theta = (\mu_1, \cdots, \mu_K, \sigma_1, \cdots, \sigma_K, \pi_1, \cdots, \pi_K)$
- The parameters are estimated by **maximizing the log likelihood**

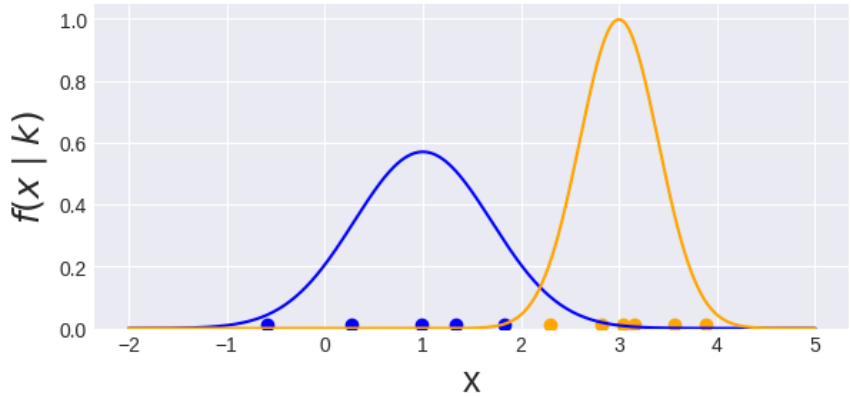
$$\hat{\theta} = \arg\max_{\theta} Q(\theta)$$

- There is no closed-form solution due to the **summation** inside the log!
- We need to apply an iterative method to find the solution - the **EM algorithm**

Intuition behind (simplified) EM

Scenario 1: if we knew the label of each data point, the task would be to estimate the parameters (similar to Gaussian Naive Bayes)



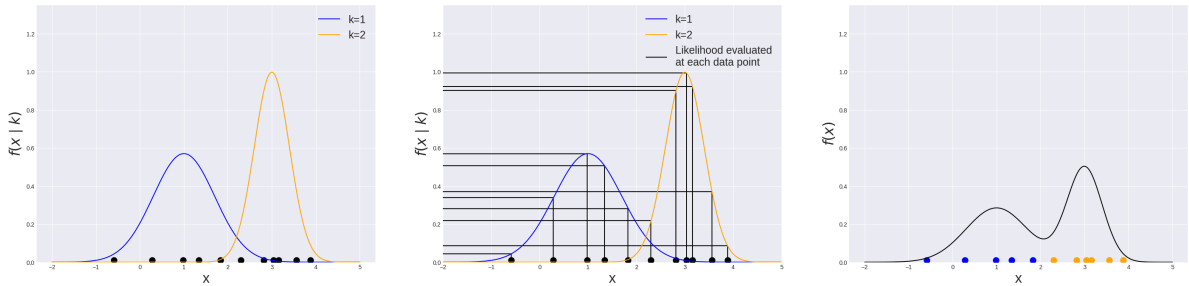


- $\pi_k = P(k) = \frac{N_k}{N}$, where N_k is the count of data points that belong to cluster k, i.e. $\pi_1 = P(\text{blue}) = \frac{5}{11}$ and $\pi_2 = P(\text{orange}) = \frac{6}{11}$
- $\mu_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i$
- $\sigma_k^2 = \frac{1}{N_k} \sum_{i=1}^{N_k} (x_i - \mu_k)^2$

Scenario 1 $\approx$ **maximization** step in the EM algorithm

Intuition behind (simplified) EM (cont.)

Scenario 2: if we knew the two priors $P(1)$, $P(2)$ and the two Gaussian distributions $f(x \mid 1)$, $f(x \mid 2)$, the task would be to compute the posterior probability $P(k \mid x)$ for $k \in \{\text{orange}, \text{blue}\}$; then we can assign each data point x to a cluster y by the maximum a posteriori estimation $y = \begin{cases} \text{blue}, & P(\text{orange} \mid x) < P(\text{blue} \mid x) \\ \text{orange}, & P(\text{orange} \mid x) \geq P(\text{blue} \mid x) \end{cases}$



- **Prior**: $\pi_1 = P(z_1) = P(\textit{blue})$, $\pi_2 = P(z_2) = P(\textit{orange})$
- **Likelihood**: $f(x \mid \text{blue}) = \text{blue}$, $f(x \mid \text{orange}) = \text{orange}$
- **Posterior**:

$$P(\text{blue} \mid x) = \frac{P(\text{blue}) f(x \mid \text{blue})}{P(\text{blue}) f(x \mid \text{blue}) + P(\text{orange}) f(x \mid \text{orange})}$$

$$P(\text{orange} \mid x) = \frac{P(\text{orange}) f(x \mid \text{orange})}{P(\text{blue}) f(x \mid \text{blue}) + P(\text{orange}) f(x \mid \text{orange})}$$

Scenario 2 $\approx$ **expectation** step in the EM algorithm

Intuition behind (simplified) EM (cont.)

- In reality, we don't know any of these!
- The idea here is that we start with some initial guesses and alternate scenario 1 and 2 **iteratively** until convergence
- This is essentially how the **Expectation-Maximization (EM) algorithm** works
  - **E-step (expectation)**: estimate the posterior for all data points given each cluster
  - **M-step (maximization)**: estimate the parameters for each cluster

The EM algorithm: two main steps

Two main steps in the EM algorithm

- **E-step (expectation)**: compute the **posterior probability** of the cluster for each data point x_i

$$\gamma_{ik} = P(k \mid x_i) = \frac{\pi_k f(x_i \mid k)}{\sum_{c=1}^{K} \pi_c f(x_i \mid c)}, \quad \text{for all } k = 1, \cdots, K$$

  This posterior is also called the **responsibility**, denoted as γ_{ik}
- **M-step (maximization)**: estimate the parameters
  - $\pi_k = P(k) = \frac{N_k}{N}$, where $N_k = \sum_{i=1}^{N} \gamma_{ik}$ - **soft clustering**
  - $\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} x_i$
  - $\sigma_k^2 = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)^2$

  for each cluster $k = 1, \cdots, K$

K-means as a special case of the EM algorithm

Expectation Step

- EM: **soft clustering** - posterior

$$\gamma_{ik} = P(k \mid x_i)$$

  Why soft? - It is a probability, i.e. $\gamma_{ik} \in [0, 1]$
- K-means: **hard clustering** - equivalent to

$$\gamma_{ik} = \begin{cases} 1, & \text{if the centroid of cluster } k \text{ is the closest to } x_i \\ 0, & \text{otherwise} \end{cases}$$

  Why hard? - It is a binary decision, i.e. $\gamma_{ik} \in \{0, 1\}$

Maximization Step

- EM: need to estimate μ_k, σ_k, where

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{K} \gamma_{ik} x_i, \quad \sigma_k^2 = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)^2, \quad N_k = \sum_{i=1}^{N} \gamma_{ik}, \quad \text{for } \gamma_{ik} \in [0, 1]$$

- K-means: only need to estimate μ_k, where

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} x_i = \frac{1}{N_k} \sum_{x \in \text{cluster } k} x, \quad N_k = \sum_{i=1}^{N} \gamma_{ik}, \quad \text{for } \gamma_{ik} \in \{0, 1\}$$

The complete EM algorithm

- These two steps are the core of the EM algorithm
- There are some extra steps involved
- The EM algorithm is an **iterative method**
- There are three important components in an iterative method:
  1) Initialization step
  2) Update the parameters in a while loop (the core, i.e. the maximization step and the expectation step)
  3) Stopping criteria

The complete EM algorithm (cont.)

- **1) Initialization step**: initialize π_k, μ_k, σ_k manually (randomly) or using, e.g. the K-means algorithm μ_k, for all $i = 1, \cdots, N$, $k = 1, \cdots, K$
- **2) Update the parameters in a while loop**: repeat the expectation step and the maximization step until the stopping criteria are met; each repetition of this process is called **one iteration**
- **3) Stopping criteria**: something you check (using e.g. a conditional statement) inside the while loop; if the stopping criteria are true, the loop shall be escaped - then you are done! There are two alternative stopping criteria for the EM algorithm:
  - Has the objective function, i.e. the log likelihood (cf. Eq. (2)), stopped changing since the last iteration?
  - Have any of the parameters, i.e. π_k, μ_k, σ_k, stopped changing since the last iteration?

Take away

- GMM: weighted summation of PDFs
- GMM vs K-means?
- GMM vs Gaussian naive Bayes?
- EM implementation (what does each step do)?

Hyperparameter K: AIC and BIC

How to choose hyperparameter K

- For a given data set, we need to choose the number of clusters K
- Similar to K-means, we first estimate $\hat{\theta}$ and then we choose the K value that gives the best clustering quality
- We introduce two **alternative** criteria for this task: **Akaike Information Criterion (AIC)** and **Bayesian Information Criterion (BIC)**
- Principle: low error + low model complexity

How to choose hyperparameter K (cont.)

Let c_K be the number of parameters to be estimated:

$$c_K = \overbrace{K \times d \times (d+1)/2}^{\text{covariance matrices}} + \overbrace{(K-1)}^{\text{priors}} + \overbrace{d \times K}^{\text{means}}$$

Note: the covariance matrix is symmetric $\Rightarrow$ (d diagonal elements + d^2 all elements)/2

- **Akaike Information Criterion (AIC)**

$$\begin{aligned} AIC(K) &= \overbrace{-\log(\text{likelihood})}^{\text{How well the model explains data ("error")}} + \overbrace{c_K}^{\text{Complexity of the model}} \\ &= -Q(\hat{\theta}) + c_K \end{aligned}$$

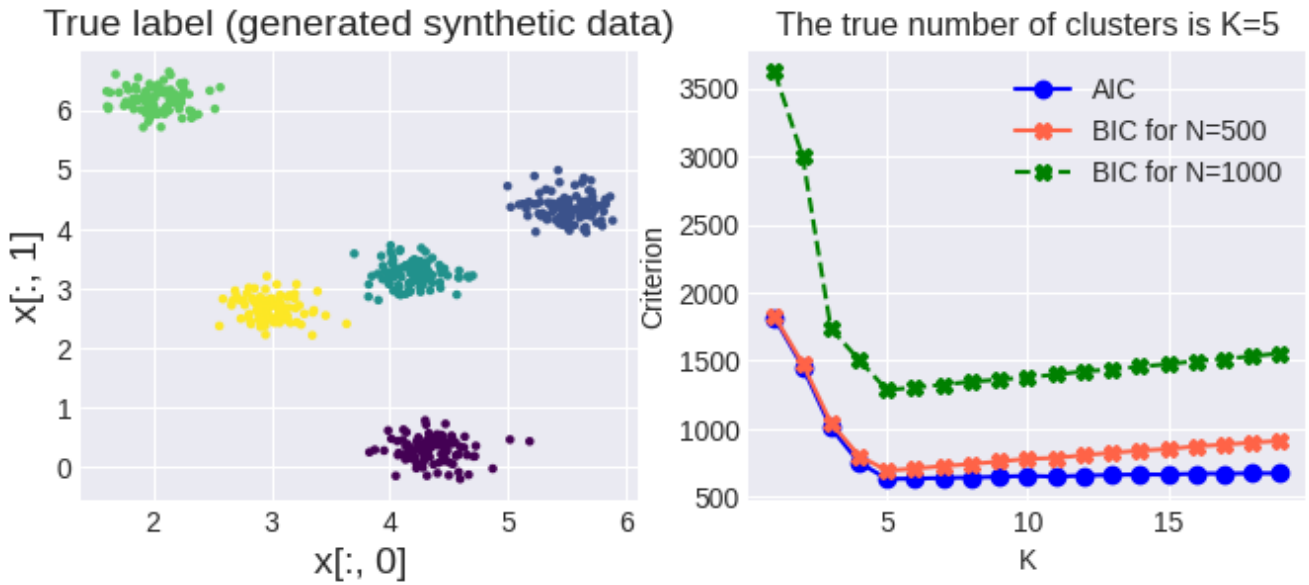
- **Bayesian Information Criterion (BIC)**

$$\begin{aligned} BIC(K) &= \overbrace{-\log(\text{likelihood})}^{\text{How well the model explains data ("error")}} + \overbrace{\frac{1}{2} c_K \log N}^{\text{Complexity of the model}} \\ &= -Q(\hat{\theta}) + \frac{1}{2} c_K \log N \end{aligned}$$

Note: an alternative definition is to multiply this definition of AIC and BIC by 2

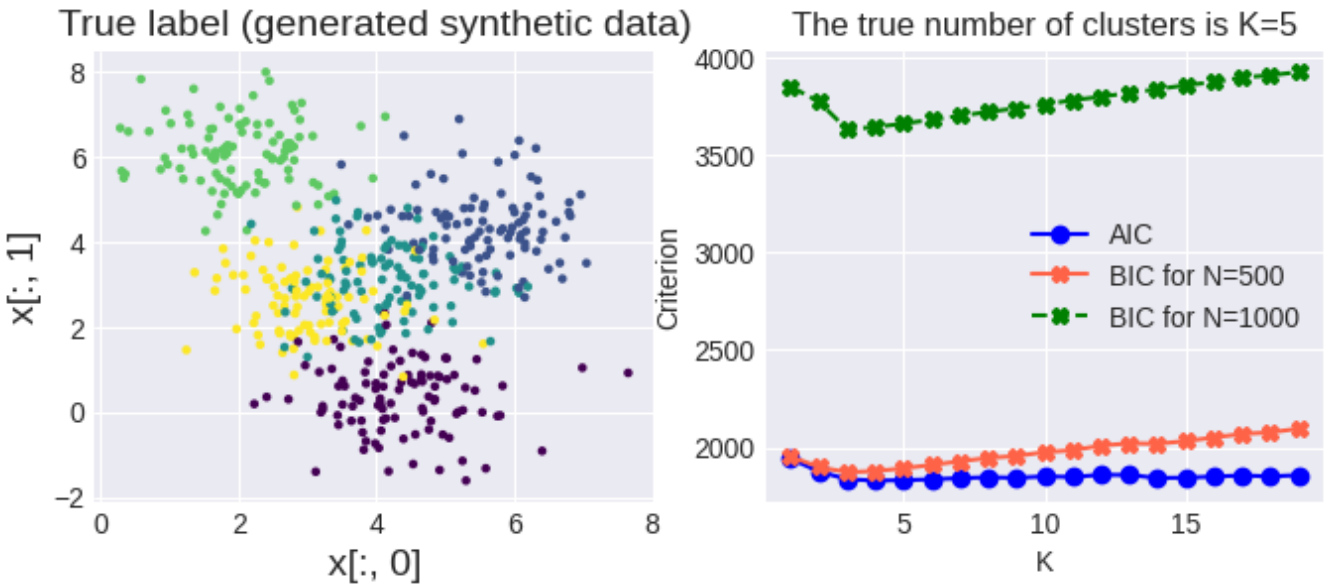
AIC vs BIC

- The idea is to find the best K that balances the "error" and the complexity of the model - Occam's Razor (cf. lecture 5) - if two models explain the data equally well, we choose the simpler one!
- BIC penalizes the complexity more than AIC - BIC increases more as K gets larger
- Example 1: well separated clusters



AIC vs BIC (cont.)

- Example 2: overlapping clusters



High dimensional GMM

- The second problem on page 13 is the high dimensional joint (multivariate) probability distribution of the correlated features
- The EM steps for $d > 1$ is presented as follows

Expectation Step

$$\gamma_{ik} = P(k \mid \boldsymbol{x}_i) = \frac{\pi_k f(\boldsymbol{x}_i \mid k)}{\sum_{c=1}^{K} \pi_c f(\boldsymbol{x}_i \mid c)}, \quad \text{for all } k = 1, \cdots, K$$

Maximization Step

- $\pi_k = P(k) = \frac{N_k}{N}$, where $N_k = \sum_{i=1}^{N} \gamma_{ik}$
- $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} \boldsymbol{x}_i$
- $\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T$ - note that $\boldsymbol{x}_i$ is a column vector here

The covariance matrix $\boldsymbol{\Sigma}_k$ captures the dependence between features

Note: you should be able to calculate the two dimensional case

Recap: GMM

- Gaussian Mixture Models (GMM) is a mixture distribution characterized by a mixture PDF

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f(\boldsymbol{x} \mid k)$$

where each $f(\boldsymbol{x} \mid k)$ is a multivariate Gaussian PDF for each Gaussian component k multivariate because $x \in \mathbb{R}^d$ is a $d \geq 1$ dimensional feature vector

  - $f(\boldsymbol{x})$ is the PDF of the mixture model with **parameters** $\theta = \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \cdots, \boldsymbol{\Sigma}_K\}$
  - $f(\boldsymbol{x})$ is the **likelihood** of θ given data $\boldsymbol{x}$
- **Parameter estimation**: maximum **likelihood** estimation - given $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N$

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \log(f(\boldsymbol{x}_i))$$

  - No closed form solution - iterative algorithm for finding $\hat{\theta}$ - the EM algorithm
- Comparison to other techniques
  - Gaussian naive Bayes classifiers vs GMM
  - K-means vs the EM algorithm for parameter estimation

GMM: pros and cons

- Pros:
  - Relatively simple compared to other mixture models we love Gaussians!
  - Flexible due to the soft clustering criterion
- Cons:
  - Might get stuck on local optimum of the objective function
  - Convergence can be slow
  - Covariance matrix estimate might lead to divergence in case of small data set
  - Need to choose the hyperparameter K manually
  - Gaussian mixture assumption might not be true

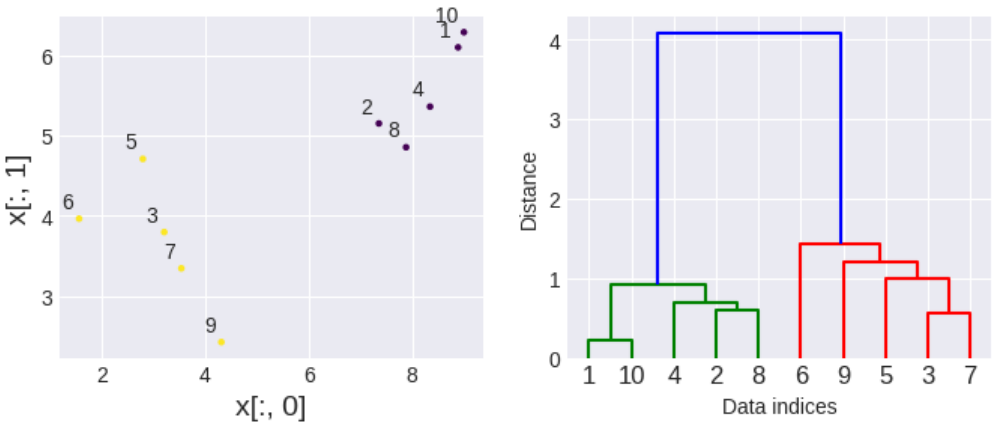
Today

Agglomerative vs divisive

Two types of hierarchical clustering

- Agglomerative (bottom-up):
  - Start with each data point being its own cluster
  - Merge the closest clusters until there is only one cluster
- Divisive (top-down):
  - Start with one cluster
  - Split until each cluster contains only one data point

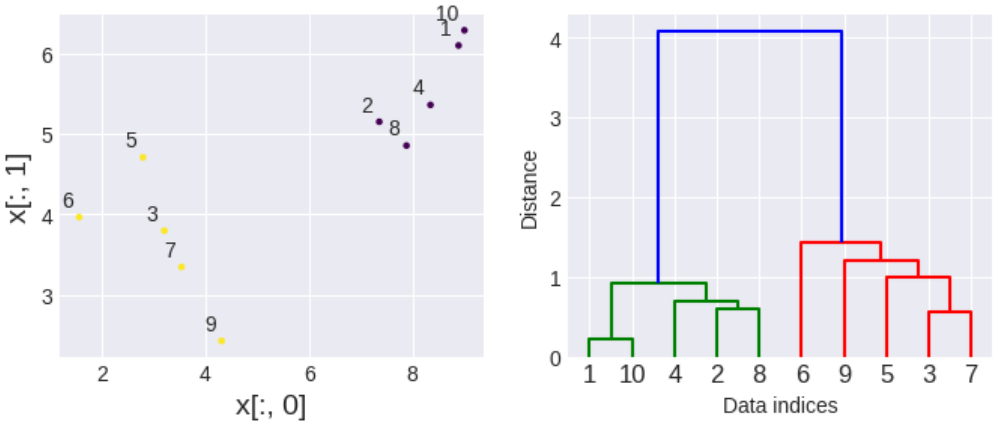
Agglomerative hierarchical clustering using a dendrogram



- Step 1: consider each data point as its own cluster; find the closest clusters - 1 and 10; group 1 and 10 into one cluster; the height of the dendrogram indicates the Euclidean distance - now we have 9 clusters in total
- Step 2: find the closest clusters - 3 and 7 - group 3 and 7 into one cluster
- Step 3: repeat until there is only one cluster left
- Step 4: draw a horizontal line to split data into different clusters

Divisive hierarchical clustering has a similar process but top-down; split can be done using, e.g., K-means

Distance between clusters - alternative linkages



- The height of the dendrogram shows the distance between two clusters
- We need to choose how to compute the distance on a cluster level when there are more than one point in a cluster
- There are different alternatives to defined the distance between two clusters, e.g. the distance between cluster {1, 10} and cluster {4, 2, 8}
  - **Single-linkage**: the distance between the closest pair, i.e. $dist(1, 4)$
  - **Complete-linkage**: the distance between the farthest pair, i.e $dist(10, 8)$
  - **Centroid**: the distance between two centroids, i.e.

$$dist(\text{centroid}(\{1,10\}), \text{centroid}(\{4, 2, 8\}))$$

In this example, $dist(\cdot, \cdot)$ is the Euclidean distance

Hierarchical clustering: pros and cons

- Pros:
  - No need to choose K
  - Easy to implement
  - Might give a meaningful taxonomy
- Cons:
  - Once two clusters are grouped, the action cannot be undone
  - Does not scale well with large data set
  - No well defined objective function; rather heuristic

Today

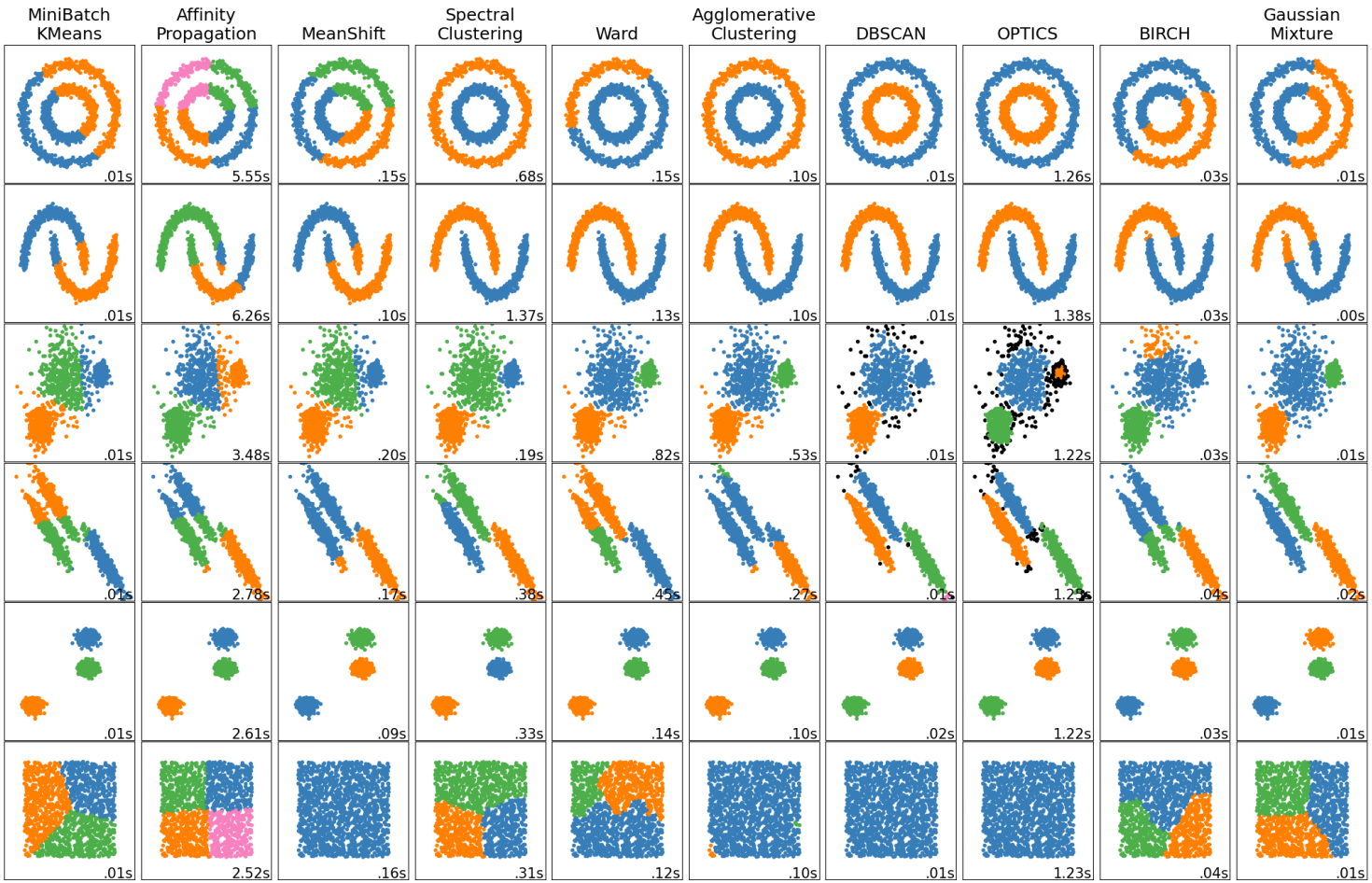
Introduction

- Idea: cluster data based on their closest points, i.e. the neighborhood
- Hyperparameter
  - Radius of the neighborhood ϵ
  - Minimum number of points n in its ϵ neighborhood

Density clustering: pros and cons

- Pros:
  - Handles clusters with arbitrary shapes
  - Handles noise explicitly
- Cons:
  - Sensitive to the sampling technique in the neighborhood
  - Need to choose hyperparameters ϵ and n
  - Not optimal for clusters with varying density

Comparison (from Python scikit-learn)



Today

Validation criteria

Cluster validation is to evaluate the quality of clusters; there are two types of criteria: internal and external

- Internal criteria: unsupervised; cluster labels are unknown
  - Algorithms that assume clusters with spherical shapes
    - Silhouette score
    - SSE
    - Many other indices, e.g. Davies–Bouldin index, Dunn index, etc
  - Algorithms that assume mixture distributions
    - AIC
    - BIC
  - Distance based criteria; more generic
    - Similarity matrix with data ordered by cluster indices
- External criteria: supervised; ground truth labels are given
  - Purity
  - F1-score
  - Entropy and mutual information

Today

Summary

So far:

- Data types and data containers
- Descriptive data analysis: descriptive statistics, visualization
- Probability distributions, events, random variables, PMF, PDF, parameters
- CDF, Q-Q plot, how to compare two distributions (data vs theoretical, data vs data)
- Modeling
- Parameter estimation: maximum likelihood estimation (MLE) and maximum a posteriori estimation (MAP)
- Classification, multinomial naive Bayes classifier, Gaussian naive Bayes classifier
- Central limit theorem, interval estimation
- Clustering, cluster tendency
- Centroid clustering, k-means, parameter estimation, SSE, Silhouette score
- Gaussian Mixture Models, AIC/BIC
- The EM algorithm

Next:

- Recap: probability distributions, likelihood function, MLE, MAP, the Bayes' rule, CLT
- Hypothesis testing