Lecture 10: Clustering Part II

Statistical Methods for Data Science

Yinan Yu

Department of Computer Science and Engineering

December 7, 2020

Today

1. Clustering - recap
2. Centroid clustering
3. Distribution clustering
   - Gaussian Mixture Models (GMM)
   - One dimensional GMM
4. Summary

Learning outcome

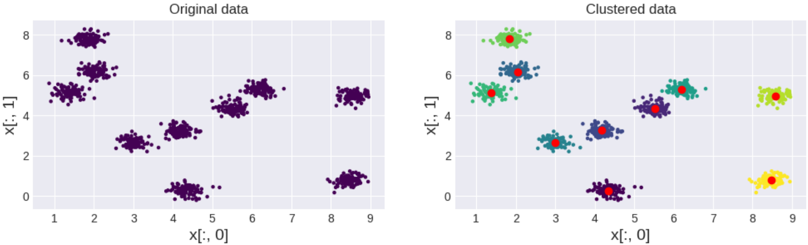
- Be able to explain the within-cluster sum of squared error (SSE) and the Silhouette score; be able to determine K and the best initial guesses using SSE and the Silhouette score
- Be able to explain the difference between Gaussian naive Bayes classifier and GMM in terms of parameter estimation
- Be 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

Today

1. Clustering - recap
2. Centroid clustering
3. Distribution clustering
4. Summary

Recall: clustering

- We start with blobs of data



- We assign some semantics to each of these data points
- Each of these semantics is called a **cluster**
- The process of finding clusters is called **clustering**

Recall: clustering (cont.)

Four categories of clustering models

- Centroid clustering
- Distribution clustering
- Hierarchical clustering
- Density clustering

Recall: models in this course

In this course, we focus on two clustering models

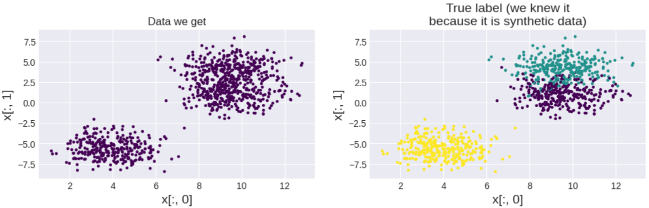
- K-means
  - **Parameters**: K centroids
  - **Hyperparameters**: K
  - **Parameter estimation**: an iterative method to update the centroids until convergence (simplified Expectation-Maximization algorithm)
- Gaussian mixture models
  - **Parameters**: K priors, K Gaussian likelihood (the big two!)
  - **Hyperparameters**: K
  - **Parameter estimation**: the Expectation-Maximization algorithm

Today

1. Clustering - recap
2. Centroid clustering
3. Distribution clustering
4. Summary

Recall: K-means

- **Data** x: d dimensional feature vector $\boldsymbol{x}$



- **Target** y:

$$y = \arg\min_{k\in\{1,\cdots,K\}} dist(\boldsymbol{x}, \boldsymbol{\mu}_k)$$

  where $dist(\cdot,\cdot)$ is a distance measure; in this course, we use the Euclidean distance (cf. lecture 9)

- **Parameters**: K centroids $\boldsymbol{\mu}_k$
- **Hyperparameters**: K
- **Parameter estimation**: an iterative method to update the centroids until convergence
- It is a **hard clustering** technique - one data point is assigned to only one cluster

Two challenges for K-means

- **Challenges**:
  - How to choose the hyperparameter K?
  - K-means is sensitive to the initialization of $\boldsymbol{\mu}_k$ for $k = 1, \cdots, K$
- **Solution**:
  - Choose a range of **candidate values**, e.g. for the first problem, we can choose $K \in \{1, \cdots, 10\}$; for the second problem, we can randomly select 100 different initial guesses for $\boldsymbol{\mu}_k$
  - For each of these **candidate values**, we run the K-means algorithm to estimate the parameters and evaluate the **quality** of the clusters produced by these parameters
  - Choose the **candidate value** that gives the best **quality**
- **Quality** evaluation criteria
  - Within-cluster sum of squared errors (SSE)
  - Silhouette score

Cluster quality evaluation criteria 1: SSE

Two commonly used alternative evaluation criteria

1. **Within-cluster sum of squared errors (SSE)**: defied as the summation of the distances from all the data points to their closest centroid

$$SSE = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} dist(\boldsymbol{x}, \boldsymbol{\mu}_k)^2 \tag{1}$$

where C_k denote cluster k; $dist(\cdot, \cdot)$ is a distance measure (**Euclidean distance**: $dist(\boldsymbol{x}, \boldsymbol{\mu}_k)^2 = (\boldsymbol{x} - \boldsymbol{\mu}_k)^T(\boldsymbol{x} - \boldsymbol{\mu}_k)$ for column vectors $\boldsymbol{x}$ and $\boldsymbol{\mu}_k$)

Cluster quality evaluation criteria 1: SSE

Two commonly used alternative evaluation criteria

1. **Within-cluster sum of squared errors (SSE)**: defied as the summation of the distances from all the data points to their closest centroid

$$SSE = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} dist(\boldsymbol{x}, \boldsymbol{\mu}_k)^2 \tag{1}$$

where C_k denote cluster k; $dist(\cdot, \cdot)$ is a distance measure (**Euclidean distance**: $dist(\boldsymbol{x}, \boldsymbol{\mu}_k)^2 = (\boldsymbol{x} - \boldsymbol{\mu}_k)^T(\boldsymbol{x} - \boldsymbol{\mu}_k)$ for column vectors $\boldsymbol{x}$ and $\boldsymbol{\mu}_k$)
Example:

- Given $\boldsymbol{x}_1 = [x_1^1, x_2^1]$, $\boldsymbol{x}_2 = [x_1^2, x_2^2]$, $\boldsymbol{x}_3 = [x_1^3, x_2^3]$, $\boldsymbol{x}_4 = [x_1^4, x_2^4]$, where $\boldsymbol{x}_1, \boldsymbol{x}_2 \in$ cluster 1 with centroid $\boldsymbol{\mu}_1 = [\mu_1^1, \mu_2^1]$; $\boldsymbol{x}_3, \boldsymbol{x}_4 \in$ cluster 2 with centroid $\boldsymbol{\mu}_2 = [\mu_1^2, \mu_2^2]$
- The SSE is computed as

$$\begin{aligned} SSE &= \text{distance in cluster } 1 + \text{distance in cluster } 2 \\ &= \underbrace{(x_1^1 - \mu_1^1)^2 + (x_2^1 - \mu_2^1)^2}_{dist(\boldsymbol{x_1}, \boldsymbol{\mu_1})^2} + \underbrace{(x_1^2 - \mu_1^1)^2 + (x_2^2 - \mu_2^1)^2}_{dist(\boldsymbol{x_2}, \boldsymbol{\mu_1})^2} \\ &\quad + \underbrace{(x_1^3 - \mu_1^2)^2 + (x_2^3 - \mu_2^2)^2}_{dist(\boldsymbol{x_3}, \boldsymbol{\mu_2})^2} + \underbrace{(x_1^4 - \mu_1^2)^2 + (x_2^4 - \mu_2^2)^2}_{dist(\boldsymbol{x_4}, \boldsymbol{\mu_2})^2} \end{aligned}$$

Cluster quality evaluation criteria 1: SSE (cont.)

1. **Within-cluster sum of squared errors (SSE)** (cont.):

$$SSE = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} dist(\boldsymbol{x}, \boldsymbol{\mu}_k)^2$$

- SSE is essentially an error term: we want SSE to be small - choose the K value that minimizes SSE?

Cluster quality evaluation criteria 1: SSE (cont.)

1. **Within-cluster sum of squared errors (SSE)** (cont.):

$$SSE = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} dist(\boldsymbol{x}, \boldsymbol{\mu}_k)^2$$

- SSE is essentially an error term: we want SSE to be small - choose the K value that minimizes SSE?
- Note that $SSE \to 0$ for $K \to N$, i.e. when every data point is their own centroid, $SSE = 0$, which is not optimal - we can't simply choose the K value that corresponds to the smallest SSE

Cluster quality evaluation criteria 1: SSE (cont.)

1. **Within-cluster sum of squared errors (SSE)** (cont.):

$$SSE = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} dist(\boldsymbol{x}, \boldsymbol{\mu}_k)^2$$

- SSE is essentially an error term: we want SSE to be small - choose the K value that minimizes SSE?
- Note that $SSE \to 0$ for $K \to N$, i.e. when every data point is their own centroid, $SSE = 0$, which is not optimal - we can't simply choose the K value that corresponds to the smallest SSE
- Instead, the best K is defined as the **elbow** point of the SSE, i.e. the point with the maximum curvature
- The elbow can be computed using, e.g. the kneed library in Python
- This method is also called the **elbow method**

Cluster quality evaluation criteria 2: Silhouette score

2. **Silhouette score** S: the idea is that a good clustering should end up with **compact clusters** with **large separation between different clusters**. This is characterized by the **within-cluster distance** and **between-cluster distance**

Cluster quality evaluation criteria 2: Silhouette score (cont.)

2. **Silhouette score** S (cont.):
   Example: given data $x_1, x_2, x_3 \in$ cluster 1, $x_4, x_5 \in$ cluster 2, $x_6, x_7 \in$ cluster 3; $K = 3$

Cluster quality evaluation criteria 2: Silhouette score (cont.)

2. **Silhouette score S** (cont.):
Example: given data $x_1, x_2, x_3 \in$ cluster 1, $x_4, x_5 \in$ cluster 2, $x_6, x_7 \in$ cluster 3; $K = 3$
   - **Within-cluster distance**: measures how data points scatter in relation to $\boldsymbol{x}_i$ within its own cluster; let $\boldsymbol{x}_i$ be a data point from cluster k,

$$a_i = \frac{1}{|C_k| - 1} \sum_{\boldsymbol{x}_j \in C_k \text{ and } j \neq i} dist(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

In this example, let $i = 1$, $x_1 \in C_1$; there are $|C_1| = 3$ data points in cluster 1

$$a_1 = \frac{1}{3-1}\big(dist(x_1, x_2) + dist(x_1, x_3)\big)$$

Cluster quality evaluation criteria 2: Silhouette score (cont.)

2. **Silhouette score S** (cont.):
Example: given data $x_1, x_2, x_3 \in$ cluster 1, $x_4, x_5 \in$ cluster 2, $x_6, x_7 \in$ cluster 3; $K = 3$

- **Within-cluster distance**: measures how data points scatter in relation to $\boldsymbol{x}_i$ within its own cluster; let $\boldsymbol{x}_i$ be a data point from cluster k,

$$a_i = \frac{1}{|C_k| - 1} \sum_{\boldsymbol{x}_j \in C_k \text{ and } j \neq i} dist(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

In this example, let $i = 1$, $x_1 \in C_1$; there are $|C_1| = 3$ data points in cluster 1

$$a_1 = \frac{1}{3-1}\left(dist(x_1, x_2) + dist(x_1, x_3)\right)$$

- **Between-cluster distance**: measures how data points scatter in relation to $\boldsymbol{x}_i$ when these data points are from other clusters

$$b_i = \min_{k' \neq k, k' \in \{1, \cdots, K\}} \frac{1}{N_{k'}} \sum_{\boldsymbol{x}_j \in C_{k'}} dist(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

In the example, $N_2 = N_3 = 2$

$$b_1 = \min\left(\frac{1}{2}\left(dist(x_1, x_4) + dist(x_1, x_5)\right), \frac{1}{2}\left(dist(x_1, x_6) + dist(x_1, x_7)\right)\right)$$

Cluster quality evaluation criteria 2: Silhouette score (cont.)

2. **Silhouette score** S (cont.):
   - Silhouette score for one data point:

$$S_i = \begin{cases} \frac{b_i - a_i}{\max(a_i, b_i)}, & \text{if } |C_k| > 1 \\ 0, & \text{if } N = 1 \end{cases}$$

     A large S_i indicates a compact cluster k in relation to $\boldsymbol{x}_i$ and a large distance from $\boldsymbol{x}_i$ to clusters other than k
   - Silhouette score for the data set

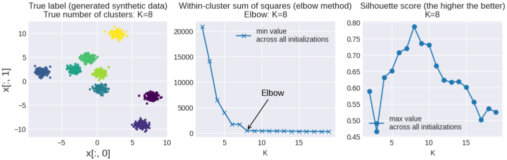
$$S = \frac{1}{N} \sum_{i=1}^{N} S_i, \quad S \in [-1, 1]$$

   - A large Silhouette score indicates a good clustering quality

Example - choose K

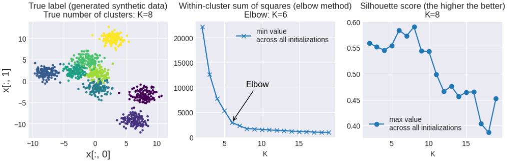
Clusters with equal variance ($K = 8$)

- SSE: $K = 8$
- Silhouette score: $K = 8$



Overlapping clusters with unequal variances ($K = 8$)

- SSE: $K = 6$
- Silhouette score: $K = 8$; but $K = 6$ and $K = 8$ have similar Silhouette scores

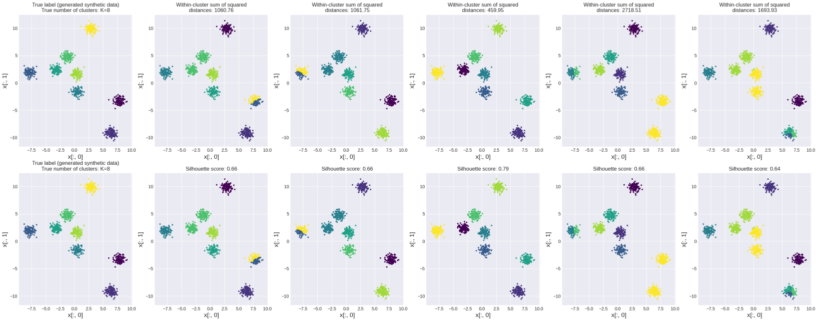


Example - choose initial guess

- Each column corresponds to a different initialization
- For a given K, choose the initialization that gives the smallest SSE or the largest Silhouette score

Clusters with equal variance ($K = 8$)

- SSE: $K = 8$
- Silhouette score: $K = 8$



True label (generated synthetic data)
True number of clusters: K=8

Within-cluster sum of squared distances: 1060.76

Within-cluster sum of squared distances: 1061.75

Within-cluster sum of squared distances: 459.95

Within-cluster sum of squared distances: 2718.51

Within-cluster sum of squared distances: 1693.93

True label (generated synthetic data)
True number of clusters: K=8

Silhouette score: 0.66

Silhouette score: 0.66

Silhouette score: 0.79

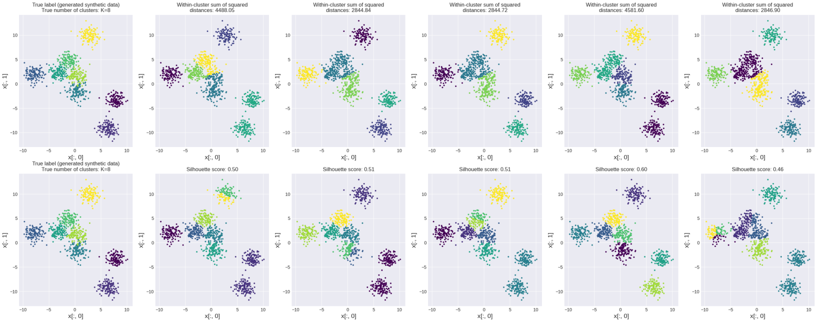
Silhouette score: 0.66

Silhouette score: 0.64

Example - choose initial guess (cont.)

Overlapping clusters with unequal variances ($K = 8$)

- SSE: $K = 6$
- Silhouette score: $K = 8$



True label (generated synthetic data)
True number of clusters: K=8

Within-cluster sum of squared distances: 4488.05

Within-cluster sum of squared distances: 2844.84

Within-cluster sum of squared distances: 2844.72

Within-cluster sum of squared distances: 4581.60

Within-cluster sum of squared distances: 2846.90

True label (generated synthetic data)
True number of clusters: K=8

Silhouette score: 0.50

Silhouette score: 0.51

Silhouette score: 0.51

Silhouette score: 0.60

Silhouette score: 0.46

Today

1. Clustering - recap
2. Centroid clustering
3. Distribution clustering
   - Gaussian Mixture Models (GMM)
   - One dimensional GMM
4. Summary

Gaussian Mixture Models (GMM)

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

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

- Gaussian Mixture Model:

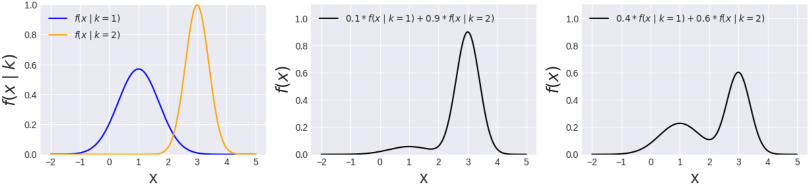
$$f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f(\boldsymbol{x} \mid k)$$

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

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

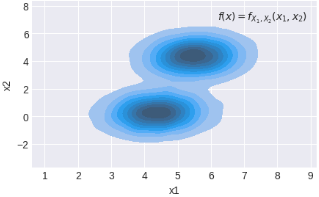


Gaussian Mixture Models (GMM) - overview (cont.)

- 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 $\boldsymbol{\Sigma}_k \in \mathbb{R}^{2\times2}$ 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 $k = 1, \cdots, K$

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

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 of the mixture distribution $f(\boldsymbol{x})$
  - The parameters for each Gaussian likelihood $f(\boldsymbol{x} \mid k)$
  - The prior $P(k)$, typically denoted as π_k

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**!
  - 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 $k = 1, \cdots, K$	
Labels for data set $\mathcal{X}$	known	unknown
	Hard assignment (one label for each $\boldsymbol{x}_i$)	Soft assignment (K probabilities for each $\boldsymbol{x}_i$)
Assumption	$\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ independent for $i \neq j$ (i.i.d.)	
	x^i_m and x^i_n independent for $m \neq n$ (**NAIVE!**)	x^i_m and x^i_n **NOT** independent for $m \neq n$

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. Now 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**; 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 i.i.d.!

Parameter estimation for GMM (cont.)

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

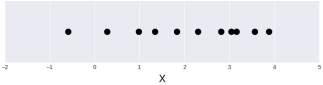
1. We **do not have the labels** - we cannot easily estimate $P(k)$ and $f(\boldsymbol{x} \mid k)$
2. Now the distribution $f(\boldsymbol{x} \mid k)$ is a **multivariate Gaussian PDF** and the features are **not necessarily independent** - now we need to explicitly work with **joint probability distributions**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

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

One dimensional GMM

Parameter estimation for one dimensional GMM

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



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

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

The joint probability distribution of all data points is defined as

$$f_{X_1,\cdots,X_N}(x_1, \cdots, x_N) \overset{i.i.d.}{=} \prod_{i=1}^{N} f(x_i) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k f(x_i \mid k) \qquad (2)$$

This is the **likelihood** of the **mixture distribution** given data $x_1, \cdots, x_N$.

- **Parameter of interest**: π_k, 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. (2) on page 28) is defined as:

$$
\begin{aligned}
Q(\theta) &= \log L(\theta \mid x_1, \cdots, x_N) = \log f_{X_1,\cdots,X_N}(x_1, \cdots, x_N) \\
&= \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} \pi_k f(x_i \mid k) \right) \qquad (3)
\end{aligned}
$$

  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** (explained in the next lecture)

How to choose hyperparameter K

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

How to choose hyperparameter K (cont.)

- **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)**

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

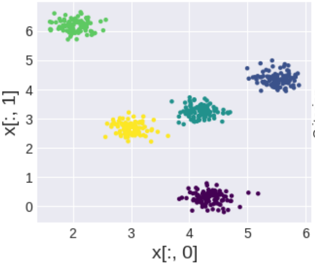
where c_K is the number of parameters to be estimated:

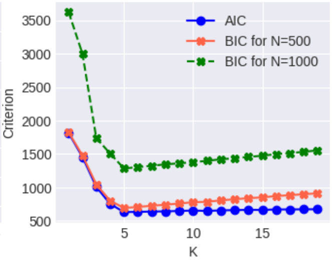
$$c_K = \overbrace{K \times d \times (d+1)/2}^{\text{covariance matrices}} + \overbrace{(K-1)}^{\text{priors}} + \overbrace{d \times K}^{\text{means}}$$

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

AIC vs BIC

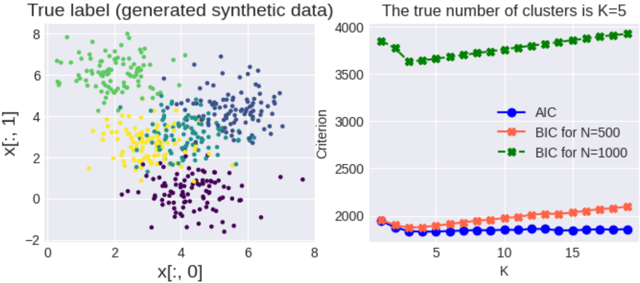
- The idea is to find the best K that balances the “error” and the complexity of the model - Occam’s Razor (cf. lecture 5) - if two models explain the data equally well, we choose the simpler one!
- BIC penalizes the complexity more than AIC - BIC increases more as K gets larger
- Example 1: well separated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


True label (generated synthetic data)


The true number of clusters is K=5

AIC vs BIC (cont.)

- Example 2: overlapping clusters



Today

1. Clustering - recap
2. Centroid clustering
3. Distribution clustering
4. Summary

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
- Hypothesis tests, comparison of two classifiers
- Clustering, cluster tendency, k-means
- SSE and Silhouette score for cluster evaluation, one dimensional Gaussian Mixture Models, AIC/BIC

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
- Hypothesis tests, comparison of two classifiers
- Clustering, cluster tendency, k-means
- SSE and Silhouette score for cluster evaluation, one dimensional Gaussian Mixture Models, AIC/BIC

Next:

- The EM algorithm in detail, hierarchical clustering, density clustering

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
- Hypothesis tests, comparison of two classifiers
- Clustering, cluster tendency, k-means
- SSE and Silhouette score for cluster evaluation, one dimensional Gaussian Mixture Models, AIC/BIC

Next:

- The EM algorithm in detail, hierarchical clustering, density clustering

Before next lecture:

- GMM clustering model