DDA3020 Machine Learning: Lecture 16 Gaussian Mixture Models and Expectation-Maximization Algorithm

Jicong Fan
School of Data Science, CUHK-SZ

18/11/2025

1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

Mixture Models

- We model the joint distribution over $(\mathbf{x}, z)$ as follows

$$p(\mathbf{x}, z) = p(\mathbf{x}|z)p(z),$$

where $\mathbf{x}$ denotes a feature variable, and z denotes the class label variable.

- However, we do not have the class labels z in unsupervised clustering.
- In this case, we can model the marginal distribution over $\mathbf{x}$ as follows

$$p(\mathbf{x}) = \sum_{z} p(\mathbf{x}, z) = \sum_{z} p(\mathbf{x}|z)p(z)$$

- This is called mixture models.

Gaussian Mixture Model (GMM)

The most common mixture model is Gaussian mixture model (GMM).

- A GMM represents a **distribution** as

  $$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$

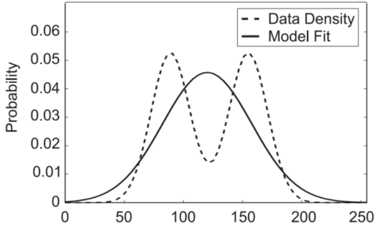
  where π_k are the mixing coefficients, $\sum_{k=1}^{K} \pi_k = 1$, $\pi_k \geq 0, \forall k.$, and

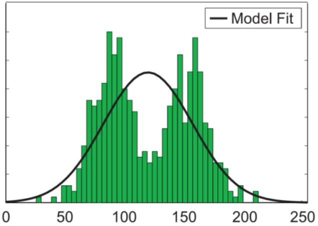
  $$\mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right),$$

  with $|\boldsymbol{\Sigma}_k| = \det(\boldsymbol{\Sigma}_k)$ denotes the determinant of $\boldsymbol{\Sigma}_k$, d indicates the dimension of $\mathbf{x}$.
- GMM is a density estimator. If given enough Gaussian components, GMM is universal approximator of densities.
- In general, mixture models are very powerful, but difficult to optimize

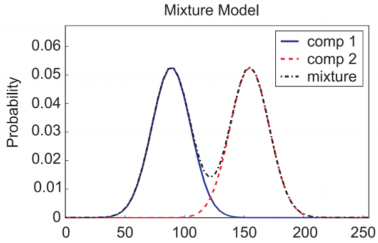
Visualizing A Mixture of Gaussians – 1D Gaussians

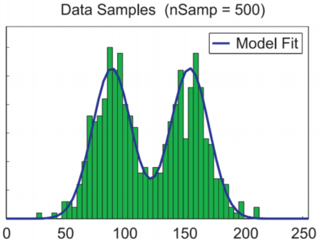
- If you fit a Gaussian to data:





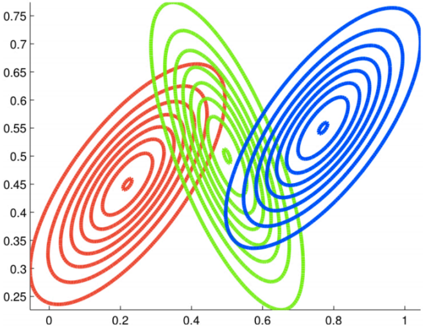
- Now, we are trying to fit a GMM (with $K = 2$ in this example):

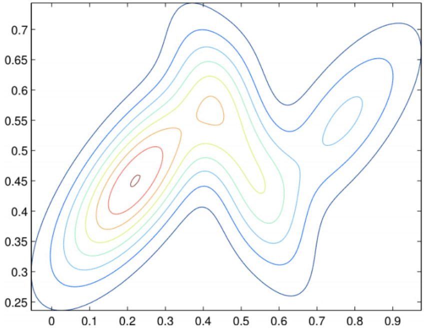


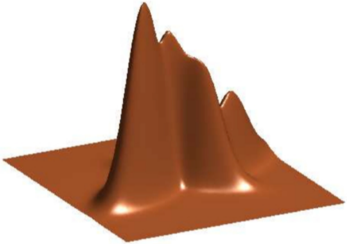


[Slide credit: K. Kutulakos]

Visualizing A Mixture of Gaussians – 2D Gaussians







Fitting GMMs: Maximum Likelihood

- The log-likelihood is

$$\log \mathcal{L}(\boldsymbol{\Theta}) = \ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \right) \right)$$

  where $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$, $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$, and $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K\}$.
- We aim to learn the parameters $\boldsymbol{\Theta}$ by maximizing the above log-likelihood.
- Due to the log-sum-exp operation, we cannot obtain a closed-form solution by setting the derivative to zero.
- Of course you can choose a gradient-based method. However, in the following, we will introduce a more elegant optimization method.

1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

Derivation from MLE

- The log-likelihood is

$$\mathcal{L} := \ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \right) \right)$$

- Let the derivative of $\mathcal{L}$ w.r.t. $\boldsymbol{\mu}_k$ be zero:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = 0 \implies \sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)} | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}^{(n)} | \mu_j, \boldsymbol{\Sigma}_j)}}_{\gamma_k^{(n)}} \Sigma_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) = 0$$

  - Note that $\gamma_k^{(n)}$ is actually a function of $\boldsymbol{\mu_k}$ and remains unknown. But we can estimate it from the previous iteration. The rationale of this operation will be explained later.

Derivation from MLE

- The log-likelihood is

$$\mathcal{L} := \ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right)$$

- Let the derivative of $\mathcal{L}$ w.r.t. $\boldsymbol{\mu}_k$ be zero:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = 0 \implies \sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)} | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}^{(n)} | \mu_j, \boldsymbol{\Sigma}_j)}}_{\gamma_k^{(n)}} \Sigma_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) = 0$$

  - Note that $\gamma_k^{(n)}$ is actually a function of $\boldsymbol{\mu_k}$ and remains unknown. But we can estimate it from the previous iteration. The rationale of this operation will be explained later.

Then $\boldsymbol{\mu_k} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \mathbf{x}^{(n)}$, where $N_k = \sum_{n=1}^{N} \gamma_k^{(n)}$.

  - $\gamma_k^{(n)}$ can be viewed as the **responsibility** of cluster k towards $\mathbf{x}^{(n)}$.
  - N_k denotes the effective number of data points assigned to component k.

Derivation from MLE

- The log-likelihood is

$$\mathcal{L} := \ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \right) \right)$$

- Let the derivative of $\mathcal{L}$ w.r.t. $\boldsymbol{\Sigma}_k$ be zero:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = 0 \implies \boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top$$

Derivation from MLE

- The log-likelihood is

$$\mathcal{L} := \ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \right)$$

- Now compute π_k (should consider the constraint $\sum_{k=1}^{K} \pi_k = 1$)
- Maximize $\tilde{\mathcal{L}} := \mathcal{L} + \lambda(\sum_{k=1}^{K} \pi_k - 1)$, λ: Lagrange multiplier

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \pi_k} = 0 \implies \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}^{(n)} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}^{(n)} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda = 0$$

Derivation from MLE

- The log-likelihood is

$$\mathcal{L} := \ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N} \left(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \right) \right)$$

- Now compute π_k (should consider the constraint $\sum_{k=1}^{K} \pi_k = 1$)
- Maximize $\tilde{\mathcal{L}} := \mathcal{L} + \lambda(\sum_{k=1}^{K} \pi_k - 1)$, λ: Lagrange multiplier

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \pi_k} = 0 \implies \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda = 0$$

- Multiply both sides by π_k and take the sum over all k: $\lambda = -N$
- Recall $\boxed{\begin{array}{l} \gamma_k^{(n)} = \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}^{(n)}|\mu_j, \boldsymbol{\Sigma}_j)} \\ N_k = \sum_{n=1}^{N} \gamma_k^{(n)} \end{array}}$, then $\pi_k = \frac{N_k}{N}$

Algorithm for Fitting GMM

Initialize $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and π_k, $k = 1, \ldots, K$. Iterate until convergence:

- **Step 1**: Evaluate the responsibilities given current parameters

$$\gamma_k^{(n)} = \frac{\pi_k \mathcal{N}\left(\mathbf{x}^{(n)} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x}^{(n)} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}, \quad k = 1, \ldots, K, \quad n = 1, \ldots, N.$$

- **Step 2**: Re-estimate the parameters given current responsibilities

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \mathbf{x}^{(n)},$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k\right) \left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k\right)^\top,$$

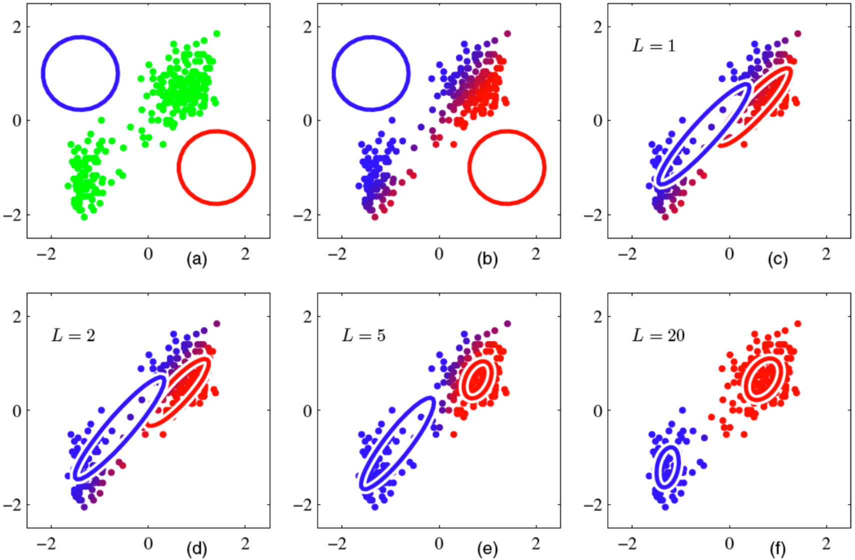
$$\pi_k = \frac{N_k}{N}, \text{ with } N_k = \sum_{n=1}^{N} \gamma_k^{(n)}.$$

- Evaluate the log-likelihood and check for convergence

$$\ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right).$$

This algorithm is actually the well-known Expectation-Maximization (EM) algorithm, where Steps 1 and 2 are called **E-step** and **M-step** respectively.

A 2D Example of Optimization for GMM



1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

Comparison between GMM and K-Means

K-Means:

- **Assignment step**: Assign each data point to the closest cluster, *i.e.*, hard assignment.
- **Refitting step**: Move each cluster center to the center of gravity of the data assigned to it.

GMM:

- **E-step**: Given the current model, compute the posterior probability over z for each data point, like soft assignment.
- **M-step**: Given the posterior probability, update the model parameters by maximizing the expected log-likelihood.

Comparison between GMM and K-Means

- If fixing the covariance matrices $\mathbf{\Sigma}$ as the identity matrix $\mathbf{I}$ for all Gaussian components, EM for GMMs is reduced to a soft version of K-means.
- Instead of hard assignments in the E-step, EM does **soft assignments** based on the softmax of the squared Euclidean distance from each point to each cluster.
- Each center moved by **weighted means** of the data, with weights given by soft assignments.
- In K-means, weights are 0 or 1.
- GMM provides a probabilistic view of clustering - Each cluster corresponds to a different Gaussian component.

1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

Examples of Clustering Results

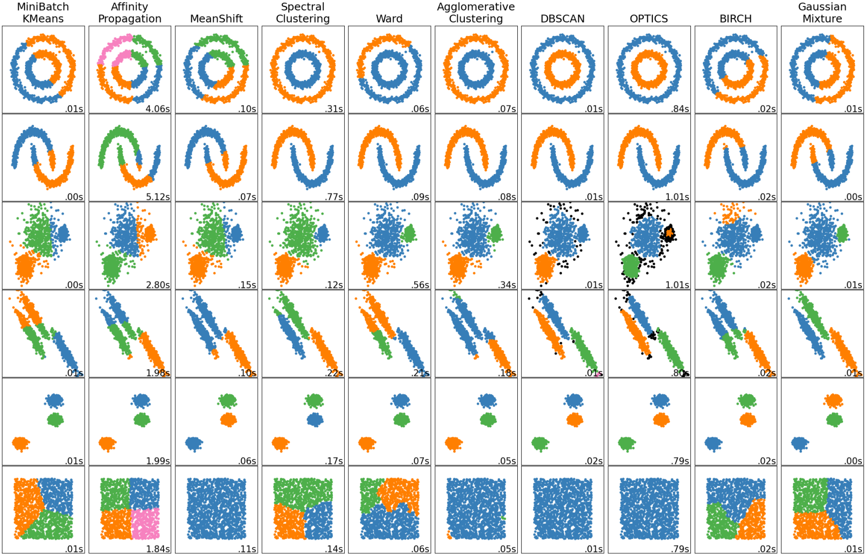


Image source: scikit-learn

Reference

Further readings:

- Chapter 9 in the book "Pattern Recognition and Machine Learning". Link
- Demo with code: https://scikit-learn.org/stable/modules/generated sklearn.mixture.GaussianMixture.html#sklearn.mixture.GaussianMix

1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

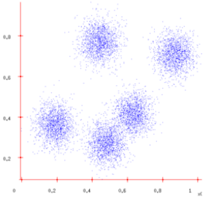
1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

A Latent Variable View of GMM



- We introduce a hidden (latent) variable z, indicating which Gaussian component generates the observation $\mathbf{x}$, with some probability.
- Let $z \sim \text{Categorical}\,(\boldsymbol{\pi})$, where $\boldsymbol{\pi} \geq 0, \quad \sum_k \pi_k = 1$
- Then:

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}, z = k) = \sum_{k=1}^{K} \underbrace{p(z = k)}_{\pi_k} \underbrace{p(\mathbf{x} \mid z = k)}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

- This breaks a complicated distribution into simple components - the price is the hidden variable.

Latent Variable Models

Latent variable model (LVM):

- Definition: A latent variable model is a statistical model that relates a set of observable variables to a set of latent variables.
- Some model variables may be unobserved, either at training or at testing time, or both. Variables which are always unobserved are called **latent variables**, or sometimes **hidden variables**.
- We may want to intentionally introduce latent variables to model complex dependencies between variables – this can actually simplify the model
- According to the type of latent variables, there are two types of LVMs,
  - LVM with continuous latent variables, *e.g.*, factor analysis
  - LVM with discrete latent variables, *e.g.*, mixture models

Reference:
https://en.wikipedia.org/wiki/Latent_variable_model

Back to GMM

- A Gaussian mixture distribution:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$

- We had: $z \sim$ Categorical $(\boldsymbol{\pi})$, *i.e.*, $p(z = k \mid \boldsymbol{\pi}) = \pi_k$, where $\boldsymbol{\pi} \geq 0, \quad \sum_k \pi_k = 1$.
- Joint distribution: $\quad p(\mathbf{x}, z) = p(z) p(\mathbf{x} \mid z)$
- Log-likelihood:

$$\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln p\left(\mathbf{x}^{(n)} \mid \pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$= \sum_{n=1}^{N} \ln \sum_{k=1}^{K} p\left(\mathbf{x}^{(n)}, z^{(n)} = k \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$= \sum_{n=1}^{N} \ln \sum_{k=1}^{K} p\left(\mathbf{x}^{(n)} \mid z^{(n)} = k; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) p(z^{(n)} = k \mid \boldsymbol{\pi})$$

- Note: we have a hidden variable $z^{(n)}$ for every observation $\mathbf{x}^{(n)}$
- If $z^{(n)}$ is known for every $\mathbf{x}^{(n)}$, then the optimization is easy.
- However, $z^{(n)}$ is unknown.

1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

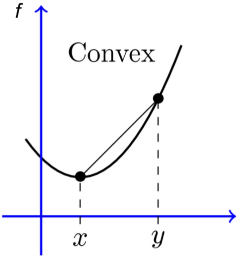
Preliminaries: Convex and Concave Functions

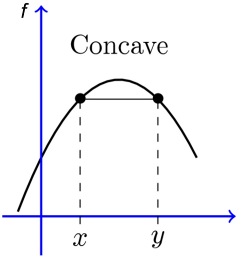
Theorem 1: Suppose f is a convex function, for any two input points x and y, as well as any scalar value $\alpha \in [0, 1]$, we have

$$f\big(\alpha x + (1-\alpha)y\big) \leq \alpha f(x) + (1-\alpha)f(y).$$

Theorem 2: Suppose f is a concave function, for any two input points x and y, as well as any scalar value $\alpha \in [0, 1]$, we have

$$f\big(\alpha x + (1-\alpha)y\big) \geq \alpha f(x) + (1-\alpha)f(y).$$





Preliminaries: Jensen's Inequality

The above theorems can be extended to Jensen's Inequality.

Theorem (Jensen's Inequality)

Suppose f is a convex function, and X is a random variable, then we have

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)].$$

If f is a concave function, then we have

$$f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)].$$

When the equality holds?

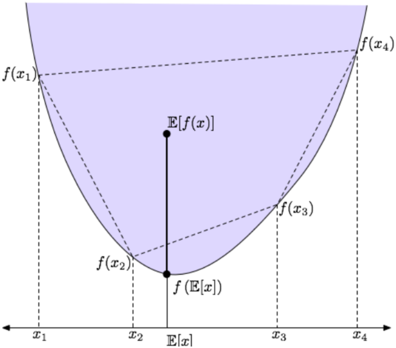
- X has a unique state
- f is not strongly convex/concave (f is affine)

Try to prove the above theorem and claims by yourself. (Hint: using mathematical induction to prove)

Preliminaries: Jensen's Inequality

For example, as shown in the right figure, f is a convex fucntion, and there are four candidate states of X, *i.e.*, x_1, x_2, x_3, x_4. Given any setting of the probability distribution (*i.e.*, $P(X = x_i) = \alpha_i$), it always has

$$f(\sum_{i=1}^{4} \alpha_i x_i) \leq \sum_{i=1}^{4} \alpha_i f(x_i).$$



1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

Notations of Latent Variable Models

- In this lecture, we'll be using $\mathbf{x}$ to denote **observed data** and z to denote the **latent variables**.
- We assume we have an observed dataset $\mathcal{D} = \left\{\mathbf{x}^{(n)}\right\}_{n=1}^{N}$ and would like to fit parameters $\boldsymbol{\theta}$ using maximum log-likelihood:

$$\log p(\mathcal{D}; \boldsymbol{\theta}) = \sum_{n=1}^{N} \log p\left(\mathbf{x}^{(n)}; \boldsymbol{\theta}\right)$$

- To compute $p(\mathbf{x}; \boldsymbol{\theta})$, we have to **marginalize** over z:

$$p(\mathbf{x}; \boldsymbol{\theta}) = \sum_{z} p(z, \mathbf{x}; \boldsymbol{\theta}),$$

  where $p(z, \mathbf{x}; \boldsymbol{\theta})$ denotes the probabilistic model we should define. Note that

  - Anything following a semicolon denotes a parameter of the distribution
  - We're not treating the parameters as random variables

Difficulty of Fitting Latent Variable Models

- Typically, there is no closed form solution to the maximum likelihood problem

$$\log p(\mathcal{D};\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p\left(\mathbf{x}^{(n)};\boldsymbol{\theta}\right) = \sum_{n=1}^{N} \log\left(\sum_{z^{(n)}} p\left(z^{(n)},\mathbf{x}^{(n)};\boldsymbol{\theta}\right)\right).$$

- Key difficulty: once z is marginalized out, $p(\mathbf{x};\theta)$ could be complex (*e.g.*, a mixture distribution).
- As shown in GMM (see last slides), if our objective is in terms of $\log p(z,\mathbf{x};\boldsymbol{\theta})$, which can be fully decomposed, then the optimization is very simple.
- To accomplish this, we need to move the summation outside the log.

Auxiliary Distribution of Latent Variables

- We firstly introduce a new distribution *w.r.t.* each latent variable $z^{(n)}$, denoted as $q_n(z^{(n)})$.
- We assume that the distributions *w.r.t.* different latent variables could be different, and they are independent, *i.e.*,

$$q(\mathbf{z}) = \prod_{n=1}^{N} q_n(z^{(n)}),$$

  where $\mathbf{z} = \{z^{(1)}, z^{(2)}, \ldots, z^{(N)}\}$.
- Note that here we don't specify the parameter value of $q_n(z^{(n)})$, which will be learned later. And, be careful that

$$q_n(z^{(n)}) \neq p(z; \boldsymbol{\pi}).$$

Decomposition of Log Likelihood

- We start from one pair of observed and latent variables, *i.e.*, $\{\mathbf{x}, z\}$. Utilizing $q(z)$, we have

$$\ln p(\mathbf{x};\boldsymbol{\theta}) = \mathbb{E}_{q(z)}\left[\ln\left(\frac{p(\mathbf{x};\boldsymbol{\theta})\cdot q(z)}{q(z)}\right)\right] = \mathbb{E}_{q(z)}\left[\ln\left(\frac{p(\mathbf{x},z;\boldsymbol{\theta})}{q(z)}\cdot\frac{q(z)}{p(z|\mathbf{x};\boldsymbol{\theta})}\right)\right]$$
$$= \mathbb{E}_{q(z)}\left[\ln\left(\frac{p(\mathbf{x},z;\boldsymbol{\theta})}{q(z)}\right)\right] + \mathbb{E}_{q(z)}\left[\ln\left(\frac{q(z)}{p(z|\mathbf{x};\boldsymbol{\theta})}\right)\right]$$

- It is natural to extend the above decomposition to the log likelihood of the whole data set $\mathcal{D}$, *i.e.*,

$$\begin{aligned}\ln p(\mathcal{D};\boldsymbol{\theta}) = & \sum\nolimits_{n=1}^{N}\mathbb{E}_{q_n(z^{(n)})}\left[\ln\left(\frac{p(\mathbf{x}^{(n)},z^{(n)};\boldsymbol{\theta})}{q_n(z^{(n)})}\right)\right] \\ & + \sum\nolimits_{n=1}^{N}\mathbb{E}_{q_n(z^{(n)})}\left[\ln\left(\frac{q_n(z^{(n)})}{p(z^{(n)}|\mathbf{x}^{(n)};\boldsymbol{\theta})}\right)\right] \\ = & \mathcal{L}(\mathbf{q};\boldsymbol{\theta}) + \sum_{n=1}^{N}\mathrm{KL}\big(q_n(z^{(n)})||p(z^{(n)}|\mathbf{x}^{(n)};\boldsymbol{\theta})\big)\end{aligned} \tag{1}$$

Decomposition of Log Likelihood

Theorem

$$\ln p(\mathcal{D};\boldsymbol{\theta}) \geq \mathcal{L}(\mathbf{q};\boldsymbol{\theta}),\ \forall \mathbf{q},\boldsymbol{\theta}.$$

Proof 1: Since $\ln(\cdot)$ is concave, utilizing the Jensen's inequality, we have

$$\begin{aligned}\mathbb{E}_{q(z)}\left[\ln\left(\frac{p(\mathbf{x},\boldsymbol{z};\boldsymbol{\theta})}{q(z)}\right)\right] &\leq \ln \mathbb{E}_{q(z)}\left(\frac{p(\mathbf{x},z;\boldsymbol{\theta})}{q(z)}\right)\\ &= \ln\sum_{k}^{K} q(z=k)\cdot\frac{p(\mathbf{x},z=k;\boldsymbol{\theta})}{q(z=k)} = \ln p(\mathbf{x};\boldsymbol{\theta}).\end{aligned}$$

Then, it is easy to prove the above theorem.

Decomposition of Log Likelihood

Theorem

$$\ln p(\mathcal{D};\boldsymbol{\theta}) \geq \mathcal{L}(\mathbf{q};\boldsymbol{\theta}),\ \forall \mathbf{q},\boldsymbol{\theta}.$$

Proof 2: According to the non-negative property of KL divergence, we have

$$\mathrm{KL}\big(\mathbf{q}(\mathbf{z})||p(\mathbf{z}|\mathcal{D};\boldsymbol{\theta})\big) \geq 0,$$

where the equality holds only when $\mathbf{q}(\mathbf{z}) = p(\mathbf{z}|\mathcal{D};\boldsymbol{\theta})$. Utilizing the decomposition of the log likelihood (*i.e.*, Eq. (1)), we can prove the above theorem.

Maximizing the Lower Bound of Log Likelihood

- Since learning $\boldsymbol{\theta}$ by maximizing $\ln p(\mathcal{D}; \boldsymbol{\theta})$ is difficult, we resort to maximize its lower bound $\mathcal{L}(\mathbf{q}; \boldsymbol{\theta})$ with some auxiliary distribution $\mathbf{q}(\mathbf{z})$, *i.e.*,

$$\max_{\mathbf{q}(\mathbf{z}), \boldsymbol{\theta}} \mathcal{L}(\mathbf{q}; \boldsymbol{\theta}) \equiv \max_{\mathbf{q}(\mathbf{z}), \boldsymbol{\theta}} \sum\nolimits_{n=1}^{N} \mathbb{E}_{q_n(z^{(n)})} \left[\ln \left(\frac{p(\mathbf{x}^{(n)}, z^{(n)}; \boldsymbol{\theta})}{q_n\left(z^{(n)}\right)} \right) \right],$$

  with the constraint $\sum_{z^{(n)}=1}^{K} q_n(z^{(n)}) = 1, \forall n$.
- We adopt the coordinate descent algorithm to solve the above optimization problem, with the following alternative steps:
  - Given $\boldsymbol{\theta}$, update $\mathbf{q}(\mathbf{z})$;
  - Given $\mathbf{q}(\mathbf{z})$, update $\boldsymbol{\theta}$.
- The whole algorithm for fitting the latent variable model is called Expectation Maximization (EM) algorithm.

Expectation Maximization: E step

Given $\boldsymbol{\theta}$, update $\mathbf{q}(\mathbf{z})$ by solving the following sub-problem:

$$
\begin{aligned}
\max_{\mathbf{q}(\mathbf{z})} \mathcal{L}(\mathbf{q};\boldsymbol{\theta}) &\equiv \max_{\mathbf{q}(\mathbf{z})} \sum_{n=1}^{N} \mathbb{E}_{q_n(z^{(n)})}\left[\ln\left(\frac{p(\mathbf{x}^{(n)}, z^{(n)};\boldsymbol{\theta})}{q_n(z^{(n)})}\right)\right] \\
&\equiv \max_{\mathbf{q}(\mathbf{z})} \sum_{n=1}^{N} \mathbb{E}_{q_n(z^{(n)})}\left[\ln\left(\frac{p(z^{(n)}|\mathbf{x}^{(n)};\boldsymbol{\theta}) \cdot p(\mathbf{x}^{(n)};\boldsymbol{\theta})}{q_n(z^{(n)})}\right)\right] \\
&\equiv \max_{\mathbf{q}(\mathbf{z})} \sum_{n=1}^{N} \mathbb{E}_{q_n(z^{(n)})}\left[\ln\left(\frac{p(z^{(n)}|\mathbf{x}^{(n)};\boldsymbol{\theta})}{q_n(z^{(n)})}\right) + \ln p(\mathbf{x}^{(n)};\boldsymbol{\theta})\right] \\
&\equiv \max_{\mathbf{q}(\mathbf{z})} \sum_{n=1}^{N} \mathbb{E}_{q_n(z^{(n)})}\left[\ln\left(\frac{p(z^{(n)}|\mathbf{x}^{(n)};\boldsymbol{\theta})}{q_n(z^{(n)})}\right)\right] + \text{constant} \\
&\equiv \min_{\mathbf{q}(\mathbf{z})} \sum_{n=1}^{N} \mathbb{E}_{q_n(z^{(n)})}\left[\ln\left(\frac{q_n(z^{(n)})}{p(z^{(n)}|\mathbf{x}^{(n)};\boldsymbol{\theta})}\right)\right] \\
&\equiv \min_{\mathbf{q}(\mathbf{z})} \sum_{n=1}^{N} \mathrm{KL}\big(q_n(z^{(n)})||p(z^{(n)}|\mathbf{x}^{(n)};\boldsymbol{\theta})\big),
\end{aligned}
$$

with the constraint $\sum_{k=1}^{K} q_n(z^{(n)} = k) = 1, \forall n$.

Expectation Maximization: E step

- Given $\boldsymbol{\theta}$, update $\mathbf{q}(\mathbf{z})$ by solving the following sub-problem:

$$\min_{\mathbf{q}(\mathbf{z})} \sum_{n=1}^{N} \mathrm{KL}\big(q_n(z^{(n)}) || p(z^{(n)} | \mathbf{x}^{(n)}; \boldsymbol{\theta})\big),$$

  with the constraint $\sum_{k=1}^{K} q_n(z^{(n)} = k) = 1, \forall n.$

- According to the property of KL divergence, it is easy to find the optimal solution, as follows:

$$q_n^*(z^{(n)}) = p(z^{(n)} | \mathbf{x}^{(n)}; \boldsymbol{\theta}).$$

  And this solution also satisfies the equality constraint.

- It is interesting to see that
  - The optimal auxiliary distribution $q_n^*(z^{(n)})$ is exactly the posterior distribution $p(z^{(n)} | \mathbf{x}^{(n)}; \boldsymbol{\theta})$
  - Since $\mathrm{KL}\big(\mathbf{q}^*(\mathbf{z}) || p(\mathbf{z} | \mathcal{D}; \boldsymbol{\theta})\big) = 0$, then

$$\ln p(\mathcal{D}; \boldsymbol{\theta}) = \mathcal{L}(\mathbf{q}^*; \boldsymbol{\theta}).$$

    It means that the gap between $\ln p(\mathcal{D}; \boldsymbol{\theta})$ and its lower bound $\mathcal{L}(\mathbf{q}^*; \boldsymbol{\theta})$ becomes 0, given the current $\boldsymbol{\theta}$.

Expectation Maximization: M step

- Given $\mathbf{q}(\mathbf{z})$, update $\boldsymbol{\theta}$ by solving the following sub-problem:

$$\max_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{q};\boldsymbol{\theta}) \equiv \max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \mathbb{E}_{q_n(z^{(n)})}\left[\ln\left(\frac{p(\mathbf{x}^{(n)}, z^{(n)};\boldsymbol{\theta})}{q_n(z^{(n)})}\right)\right]$$

$$\equiv \max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \mathbb{E}_{q_n\left(z^{(n)}\right)}\left[\log p\left(\mathbf{x}^{(n)}, z^{(n)};\boldsymbol{\theta}\right)\right] - \underbrace{\mathbb{E}_{q_n\left(z^{(n)}\right)}\left[\log q_n\left(z^{(n)}\right)\right]}_{\text{constant w.r.t. } \boldsymbol{\theta}}$$

- Substitute in $q_n\left(z^{(n)}\right) = p\left(z^{(n)} \mid \mathbf{x}^{(n)};\boldsymbol{\theta}^{\text{old}}\right)$:

$$\boldsymbol{\theta}^{\text{new}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{n=1}^{N} \mathbb{E}_{p\left(z^{(n)} | \mathbf{x}^{(n)};\boldsymbol{\theta}^{\text{old}}\right)}\left[\log p\left(z^{(n)}, \mathbf{x}^{(n)};\boldsymbol{\theta}\right)\right]$$

- This is the expected complete data log-likelihood, which is easy to optimize.

Expectation Maximization: Summary

- The EM algorithm alternates between making the bound tight at the current parameter values and then optimizing the lower bound
- If the current parameter value is $\boldsymbol{\theta}^{\text{old}}$:
  - **E-step**: Given $\boldsymbol{\theta}^{\text{old}}$, we update the auxiliary distribution $\mathbf{q}(\mathbf{z})$ to make the bound tight:

    $$\mathbf{q}(\mathbf{z}) = \underset{\mathbf{q}(\mathbf{z})}{\operatorname{argmax}} \mathcal{L}(q, \boldsymbol{\theta}^{\text{old}}). \tag{2}$$

    It leads to $q_n\big(z^{(n)}\big) = p\big(z^{(n)} \mid \mathbf{x}^{(n)}; \boldsymbol{\theta}^{\text{old}}\big), \forall n$, and makes

    $$\log p\left(\mathcal{D}; \boldsymbol{\theta}^{\text{old}}\right) = \mathcal{L}\left(q; \boldsymbol{\theta}^{\text{old}}\right)$$

  - **M-step**: Given $\mathbf{q}(\mathbf{z})$ updated above, we update $\boldsymbol{\theta}$ by optimizing the lower bound:

    $$\begin{aligned}\boldsymbol{\theta}^{\text{new}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \mathcal{L}(q, \boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{n=1}^{N} \mathbb{E}_{q_n\left(z^{(n)}\right)}\left[\log \frac{p\left(z^{(n)}, \mathbf{x}^{(n)}; \boldsymbol{\theta}\right)}{q_n\left(z^{(n)}\right)}\right]\end{aligned}$$

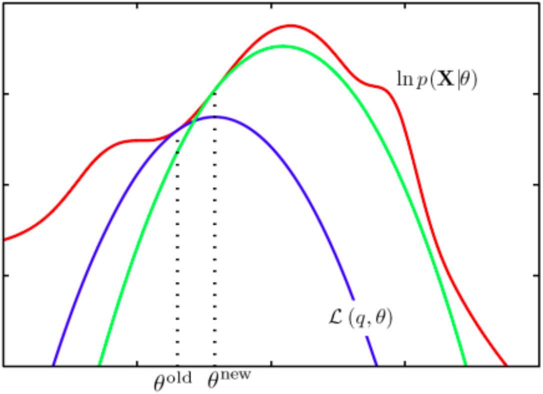
EM Convergence

- We can deduce that an iteration of EM will improve the log-likelihood by using the fact that the bound is tight at $\boldsymbol{\theta}^{\text{old}}$ after the E-step
- Let q denote the q_n after the E-step, *i.e.*, $q_n\left(z^{(n)}\right) = p\left(z^{(n)} \mid \mathbf{x}^{(n)}; \boldsymbol{\theta}^{\text{old}}\right)$

$$\begin{aligned}
\log p\left(\mathcal{D}; \boldsymbol{\theta}^{\text{new}}\right) &\geq \mathcal{L}\left(q, \boldsymbol{\theta}^{\text{new}}\right) && \text{since } \log p(\mathcal{D}; \boldsymbol{\theta}) \geq \mathcal{L}(q, \boldsymbol{\theta}) \text{ always holds} \\
&\geq \mathcal{L}\left(q, \boldsymbol{\theta}^{\text{old}}\right) && \text{since } \boldsymbol{\theta}^{\text{new}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ \mathcal{L}(q, \boldsymbol{\theta}) \\
&= \log p\left(\mathcal{D}; \boldsymbol{\theta}^{\text{old}}\right) && \text{since } \log p\left(\mathcal{D}; \boldsymbol{\theta}^{\text{old}}\right) = \mathcal{L}\left(q; \boldsymbol{\theta}^{\text{old}}\right)
\end{aligned}$$

- It tells that the log likelihood objective keeps increasing after each iteration of EM, until convergence.

EM Visualization



The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values.

1. Introduction to Gaussian Mixture Model
2. An Alternating Updating Algorithm for GMM
    - Derivation from MLE
    - Comparison between GMM and K-Means
    - Examples of Clustering
3. The Expectation-Maximization Algorithm
    - A Latent Variable View of GMM
    - Preliminaries: Jensen's Inequality
    - EM for General Latent Variable Models
    - Application of EM to GMM
    - A Summary of EM Algorithm
4. A Summary for GMM

Revisiting Gaussian Mixture Models

- Let's revisit the Gaussian mixture models from last lecture and derive the updates using our general EM algorithm
- Recall our model was:

$$p(\mathbf{x};\boldsymbol{\theta}) = \sum_z p(\mathbf{x}, z;\boldsymbol{\theta}) = \sum_z p(\mathbf{x}|z;\boldsymbol{\theta})p(z|\boldsymbol{\theta}) \tag{3}$$

$$p(z=k;\boldsymbol{\theta}) = \pi_k,\ \sum_{k=1}^{K}\pi_k = 1. \tag{4}$$

$$p(\mathbf{x} \mid z=k;\boldsymbol{\theta}) = \mathcal{N}\left(\mathbf{x};\boldsymbol{\mu_k},\boldsymbol{\Sigma_k}\right) \tag{5}$$

In this scenario, we have $\boldsymbol{\theta} = \{\boldsymbol{\pi_k}, \boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}\}_{k=1}^{K}$.

E-Step for Gaussian Mixture Models

- Let the current parameters be $\boldsymbol{\theta}^{\text{old}} = \left\{\boldsymbol{\pi}_{\boldsymbol{k}}^{\text{old}}, \boldsymbol{\mu}_{\boldsymbol{k}}^{\text{old}}, \boldsymbol{\Sigma}_{\boldsymbol{k}}^{\text{old}}\right\}_{k=1}^{K}$
- **E-step**: For all n, set $q_n\left(z^{(n)}\right) = p\left(z^{(n)} \mid \mathbf{x}^{(n)}; \boldsymbol{\theta}^{\text{old}}\right)$, *i.e.*,

$$
\begin{aligned}
q_n\left(z^{(n)} = k\right) &= p\left(z^{(n)} = k \mid \mathbf{x}^{(n)}; \theta^{\text{old}}\right) \\
&= \frac{p(z^{(n)} = k) p(\mathbf{x}^{(n)} \mid z^{(n)} = k)}{p(\mathbf{x}^{(n)})} \\
&= \frac{p(z^{(n)} = k) p(\mathbf{x}^{(n)} \mid z^{(n)} = k)}{\sum_{j=1}^{K} p(z^{(n)} = j) p(\mathbf{x}^{(n)} \mid z^{(n)} = j)} \\
&= \frac{\pi_k^{\text{old}} \mathcal{N}\left(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_{\boldsymbol{k}}^{\text{old}}, \boldsymbol{\Sigma}_{\boldsymbol{k}}^{\text{old}}\right)}{\sum_{j=1}^{K} \pi_j^{\text{old}} \mathcal{N}\left(\mathbf{x}^{(n)} \mid \mu_j^{\text{old}}, \Sigma_j^{\text{old}}\right)} \triangleq \gamma_k^{(n)}
\end{aligned}
$$

E-Step for Gaussian Mixture Models

Once we computed $\gamma_k^{(n)} = p\left(z^{(n)} = k \mid \mathbf{x}^{(n)}\right)$, we can compute the expected log likelihood, as follows:

$$
\begin{aligned}
&\sum_n \mathbb{E}_{P(z^{(n)}|\mathbf{x}^{(n)})}\left[\ln\left(P(\mathbf{x}^{(n)}, z^{(n)} \mid \boldsymbol{\theta})\right)\right] \\
=&\sum_n \sum_k \gamma_k^{(n)} \left(\ln\left(P(z^{(n)} = k \mid \boldsymbol{\theta})\right) + \ln\left(P(\mathbf{x}^{(n)} \mid z^{(n)} = k, \boldsymbol{\theta})\right)\right) \\
=&\sum_n \sum_k \gamma_k^{(n)} \left(\ln\left(\pi_k\right) + \ln\left(\mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)\right) \\
=&\sum_n \sum_k \gamma_k^{(n)} \ln\left(\pi_k\right) + \sum_n \sum_k \gamma_k^{(n)} \ln\left(\mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right),
\end{aligned}
$$

where $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}\}$. Note that the above expectation is fully decomposed to each data n and each cluster k, which will facilitate the parameter learning in the following maximization step.

M-Step for Gaussian Mixture Models

- We update the model parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}\}$ by maximizing the expected log likelihood, *i.e.*,

$$\max_{\boldsymbol{\theta}} \sum_{n}^{N} \sum_{k}^{K} \gamma_k^{(n)} \ln\left(\pi_k\right) + \sum_{n}^{N} \sum_{k}^{K} \gamma_k^{(n)} \ln\left(\mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right), \text{ s.t. } \sum_{k}^{K} \pi_k = 1.$$

- Following the derivations introduced in previous slides (see pages 9-11), it is easy to obtain the following solutions:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \mathbf{x}^{(n)}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k\right) \left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k\right)^{\top}$$

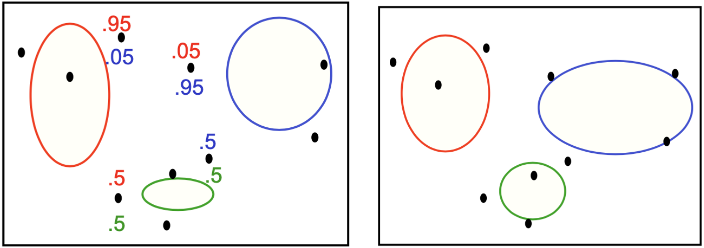
$$\pi_k = \frac{N_k}{N}, \text{ with } N_k = \sum_{n=1}^{N} \gamma_k^{(n)}$$

Note: For instance, on page 9, we have $\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}^{(n)}|\mu_j, \boldsymbol{\Sigma}_j)} \Sigma_k^{-1}(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) = 0$; here we have $\sum_{n=1}^{N} \gamma_k^{(n)} \Sigma_k^{-1}(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) = 0$ directly.

Summarization of EM for GMM

Optimization uses the **Expectation Maximization algorithm**, which alternates between two steps:

- E-step: Compute the posterior probability over z given the current model, *i.e.*, $p(z|\mathbf{x};\boldsymbol{\theta})$, which tells how much do we think each Gaussian generates each data point.
- M-step: Assuming that the data was really generated this way, update the parameters of each Gaussian component to maximize the probability that it would generate the data it is currently responsible for.



Summarization of EM for GMM

Elegant and powerful method for finding maximum likelihood solutions for models with latent variables

- **E-step:**
  - In order to adjust the parameters, we must first solve the inference problem: which Gaussian component generated each datapoint?
  - We cannot ensure, so it's a distribution over all possibilities.

$$\gamma_k^{(n)} = p\left(z^{(n)} = k \mid \mathbf{x}^{(n)}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right).$$

- **M-step:**
  - Each Gaussian gets a certain amount of posterior probability for each data point.
  - We fit each Gaussian to the weighted data points.
  - We can derive closed-form updates for all parameters

1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

A Summary of EM Algorithm

- A general algorithm for optimizing many latent variable models, such as GMMs and mixture of Bernoulli distribution
- Iteratively computes a lower bound and then optimizes it.
- Converges but maybe to a local minimum.
- Can use multiple restarts.
- Can initialize from k-means for mixture models.
- Limitation - need to be able to compute $p(z|\mathbf{x};\boldsymbol{\theta})$, not possible for more complicated models.

References

- Further reading 1: Chapter 9 in the book “Pattern Recognition and Machine Learning”. Link
- Further reading 2: Wikipedia https://en.wikipedia.org/wiki/Expectati E2%80%93maximization_algorithm
- Demo with code: https://www.kaggle.com/code/charel/learn-by-examp notebook

1. Introduction to Gaussian Mixture Model

2. An Alternating Updating Algorithm for GMM
   - Derivation from MLE
   - Comparison between GMM and K-Means
   - Examples of Clustering

3. The Expectation-Maximization Algorithm
   - A Latent Variable View of GMM
   - Preliminaries: Jensen's Inequality
   - EM for General Latent Variable Models
   - Application of EM to GMM
   - A Summary of EM Algorithm

4. A Summary for GMM

Advantages of GMMs

- Flexibility in Data Representation: Gaussian Mixture Models (GMMs) skillfully capture intricate data structures.
- Probabilistic Approach: GMMs proffer a probabilistic framework for cluster allocation, enabling a more delicate analysis of data.
- Soft Clustering: GMMs offer probabilistic cluster assignments, allowing for more nuanced data analysis.
- Effective in Overlapping Clusters: They accurately model data with overlapping clusters.
- Density Estimation Capabilities: Useful in understanding the underlying distribution of data.
- Handling Missing Data: GMMs are capable of parameter estimation despite the presence of incomplete datasets.
- Outlier Detection: Identifying data points that deviate from the general pattern.

Disadvantages of GMMs

- Difficulty in Determining Component Number (K): An incorrect assessment of K may result in overfitting or underfitting.
- Initialization Sensitivity: The initial parameter settings influence the outcome a lot.
- Assumption of Gaussian Distribution: Not always applicable if data do not follow Gaussian distributions.
- Curse of Dimensionality: Not effective in handling high-dimensional data.
- Non-singular requirement: It requires that covariance matrices are invertible.