

# Deep Generative Models: Variational Inference

Fall Semester 2025

René Vidal

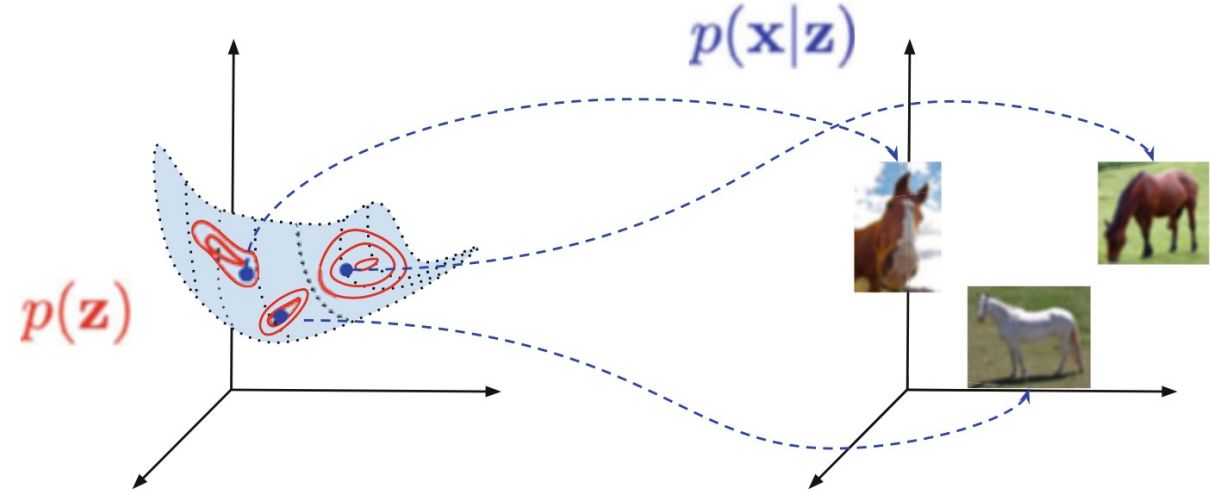
Director of the Center for Innovation in Data Engineering and Science (IDEAS),  
Rachleff University Professor, University of Pennsylvania  
Amazon Scholar & Chief Scientist at NORCE

# Outline

- Latent Variable Models
  - Probabilistic PCA
  - Variational Inference
- Variational Inference
  - Principle
  - Derivation
- Expectation Maximization
  - Derivation
  - EM for a Mixture of Gaussians

# Latent Variable Models

- $X$  = observed variable
- $Z$  = latent variable
- $\mathbf{z} \sim p(\mathbf{z})$
- $\mathbf{x} \sim p(\mathbf{x} | \mathbf{z})$



A latent variable model and a generative process. Note the low-dimensional manifold (here 2D) embedded in the high-dimensional space (here 3D)

- Factorization of the joint model

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

- Marginalization of the model

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

# Probabilistic Principal Component Analysis (PPCA)

- Let  $\mathbf{z} \sim \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{I})$  and  $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon} \mid \mathbf{0}, \sigma^2 \mathbf{I})$  be independent.
- In PPCA, the dependency between  $\mathbf{x} \in \mathbb{R}^D$  and  $\mathbf{z} \in \mathbb{R}^d$ ,  $d \ll D$ , is defined by a linear Gaussian additive model

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mathbf{b} + \boldsymbol{\epsilon}$$

- **Theorem.** Let  $\boldsymbol{\mu}_N$  and  $\boldsymbol{\Sigma}_N$  be, respectively, the ML estimates for the mean and the covariance of the data. Let  $\mathbf{U}_1$  be the matrix with the top  $d$  eigenvectors of  $\boldsymbol{\Sigma}_N$ ,  $\boldsymbol{\Lambda}_1$  be the matrix with the corresponding top  $d$  eigenvalues, and  $\lambda_i$  be the  $i$ th largest eigenvalue of  $\boldsymbol{\Sigma}_N$ . The ML estimates for the PPCA parameters  $(\mathbf{b}, \mathbf{W}, \sigma)$  is given by

$$\mathbf{b} = \boldsymbol{\mu}_N, \mathbf{W} = \mathbf{U}_1(\boldsymbol{\Lambda}_1 - \sigma^2 \mathbf{I})^{1/2} \mathbf{R} \text{ and } \sigma^2 = \frac{1}{D - d} \sum_{i=d+1}^D \lambda_i$$

where  $\mathbf{R} \in \mathbb{R}^{d \times d}$  is an arbitrary orthogonal matrix.

# What about Latent Variable Models other than PPCA

- We would like to learn the parameters of the model via Maximum Likelihood.
- Since  $\mathbf{z}$  is latent, we need to marginalize  $p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{x} | \mathbf{z})p(\mathbf{z})d\mathbf{z}$ , i.e.

$$\max_{\theta} \sum_{i=1}^N \log p_{\theta}(\mathbf{x}_i) = \max_{\theta} \sum_{i=1}^N \log \int p_{\theta}(\mathbf{x}_i | \mathbf{z})p(\mathbf{z})d\mathbf{z}$$

- For PPCA, we could compute  $p(\mathbf{x})$  in closed form and solve for  $\theta$  analytically.
- In general, we need many samples of  $\mathbf{z}$  for each  $\mathbf{x}_i$  to approximate the integral.

$$\max_{\theta} \sum_{i=1}^N \log \sum_j p_{\theta}(\mathbf{x}_i | \mathbf{z}_j)$$

- We address this challenge using **Variational Inference**, which we describe next.

# Variational Inference

- **Old ML learning objective:**  $\max_{\theta} \sum_{i=1}^N \log p_{\theta}(x_i) = \sum_{i=1}^N \log \int p_{\theta}(x_i | z) p(z) dz$

- **Theorem:** the log likelihood can be written as

$$\log p_{\theta}(\mathbf{x}) = \max_{q(\cdot|\mathbf{x}): q(\cdot|\mathbf{x}) \geq 0, \int q(z|\mathbf{x}) dz = 1} \int q(z | \mathbf{x}) \log \frac{p_{\theta}(\mathbf{x}, z)}{q(z|\mathbf{x})} dz.$$

and the maximizing distribution is given by  $q^*(\mathbf{z} | \mathbf{x}) = p_{\theta}(\mathbf{z} | \mathbf{x})$

- **New ML learning objective:**

$$\max_{\theta} \max_{q(\cdot|\mathbf{x}_i), \forall i} \sum_{i=1}^N \int q(\mathbf{z} | \mathbf{x}_i) \log \frac{p_{\theta}(\mathbf{x}_i, \mathbf{z})}{q(\mathbf{z} | \mathbf{x}_i)} dz$$

- Before going through the derivation, what is the gain here?

# Variational Inference

- **New ML learning objective:**

$$\max_{\theta} \max_{q(\cdot | \mathbf{x}_i), \forall i} \sum_{i=1}^N \int q(\mathbf{z} | \mathbf{x}_i) \log \frac{p_{\theta}(\mathbf{x}_i, \mathbf{z})}{q(\mathbf{z} | \mathbf{x}_i)} d\mathbf{z}$$

- **Expectation Maximization:**

- If finding  $q^*$  given  $\theta$  is easy, then we can alternate between finding  $q$  given  $\theta$  and vice versa
- Here  $q^*(\mathbf{z} | \mathbf{x}) = p_{\theta}(\mathbf{z} | \mathbf{x})$ . Thus, if the integral w.r.t.  $\mathbf{z}$  is easy to evaluate for a fixed  $\theta$ , we can alternate between computing the integral (E-step) and maximizing w.r.t.  $\theta$  (M-step).

- **Variational AutoEncoders:** parameterize  $q(\cdot | x_i)$  with a NN with parameters  $\psi$  that takes  $x_i$  and outputs a distribution  $q_{\psi}(\cdot | x_i)$ , and find  $(\theta, \psi)$  via SGD

$$\max_{\theta} \max_{\psi} \sum_{i=1}^N \int q_{\psi}(\mathbf{z} | \mathbf{x}_i) \log \frac{p_{\theta}(\mathbf{x}_i, \mathbf{z})}{q_{\psi}(\mathbf{z} | \mathbf{x}_i)} d\mathbf{z}$$

# Variational Inference

- **New ML learning objective:**

$$\max_{\theta} \max_{q(\cdot|\mathbf{x}_i), \forall i} \sum_{i=1}^N \int q(\mathbf{z} | \mathbf{x}_i) \log \frac{p_{\theta}(\mathbf{x}_i, \mathbf{z})}{q(\mathbf{z} | \mathbf{x}_i)} d\mathbf{z}$$

- We will use VI for many latent variable models
  - Mixtures of Gaussians (a.k.a. Gaussian Mixture Models) -> EM
  - Probabilistic Principal Component Analysis (PPCA) -> EM
  - Mixtures of PPCA -> EM
  - Variational Auto-Encoders (VAE) -> VI
  - Diffusion models -> VI
  - ...



# Variational Inference: Derivation

- Proof: Let  $q(z|x)$  be the variational distribution. Observe that

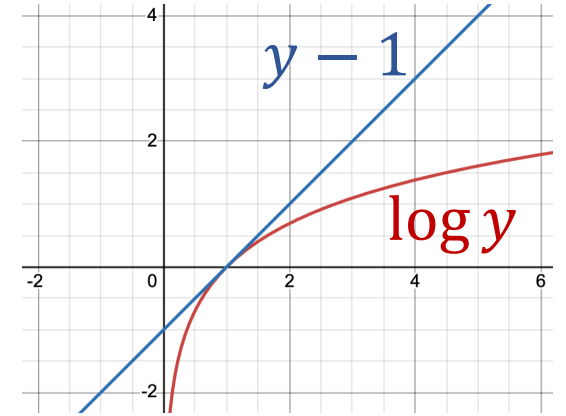
$$\begin{aligned}\log p_{\theta}(x) &= \int q(\mathbf{z} | \mathbf{x}) \log p_{\theta}(\mathbf{x}) d\mathbf{z} = \int q(\mathbf{z} | \mathbf{x}) \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{p_{\theta}(\mathbf{z} | \mathbf{x})} d\mathbf{z} \\ &= \int q(\mathbf{z} | \mathbf{x}) \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} | \mathbf{x})} \frac{q(\mathbf{z} | \mathbf{x})}{p_{\theta}(\mathbf{z} | \mathbf{x})} d\mathbf{z} \\ &= \boxed{\int q(\mathbf{z} | \mathbf{x}) \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} | \mathbf{x})} d\mathbf{z}} + \boxed{\int q(\mathbf{z} | \mathbf{x}) \log \frac{q(\mathbf{z} | \mathbf{x})}{p_{\theta}(\mathbf{z} | \mathbf{x})} d\mathbf{z}} \\ &\quad \text{Evidence Lower Bound (ELBO)} \quad \text{KL}[q(\mathbf{z} | \mathbf{x}) || p_{\theta}(\mathbf{z} | \mathbf{x})]\end{aligned}$$

- Thus,  $\max_{q(\cdot|x)} \int q(\mathbf{z} | \mathbf{x}) \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q(\mathbf{z} | \mathbf{x})} d\mathbf{z} = \max_{q(\cdot|x)} \log p_{\theta}(\mathbf{x}) - \text{KL}[q(\mathbf{z} | \mathbf{x}) || p_{\theta}(\mathbf{z} | \mathbf{x})]$   
 $= \log p_{\theta}(\mathbf{x}) - \min_{q(\cdot|x)} \text{KL}[q(\mathbf{z} | \mathbf{x}) || p_{\theta}(\mathbf{z} | \mathbf{x})] = \log p_{\theta}(\mathbf{x})$
- The last step follows because the  $KL(q || p) \geq 0$  and  $KL(q || p) = 0$  iff  $p = q$ .

# Variational Inference: Derivation

- **Proposition:**  $\text{KL}(q \parallel p) \geq 0$ . Further,  $\text{KL}(q \parallel p) = 0$  if and only if  $p = q$ .

- **Lemma:**  $\log y \leq y - 1$ , equality holds if and only if  $y = 1$ 
  - This is by the concavity of  $\log(\cdot)$



- **Proof of the Proposition:**  $\text{KL}(q, p) = \int q(x) \log \frac{q(x)}{p(x)} dx$

$$= -\int q(x) \log \frac{p(x)}{q(x)} dx$$

$$\geq -\int q(x) \left( \frac{p(x)}{q(x)} - 1 \right) dx \quad (\text{by the lemma})$$

$$= -\int p(x) dx + \int q(x) dx$$

$$= -1 + 1 = 0$$

# Deep Generative Models: Expectation Maximization

Fall Semester 2025

René Vidal

Director of the Center for Innovation in Data Engineering and Science (IDEAS),  
Rachleff University Professor, University of Pennsylvania  
Amazon Scholar & Chief Scientist at NORCE



# Variational Inference

- **Theorem:** the log likelihood criterion can be written in variational form as

$$\max_{\theta} \sum_{i=1}^N \log p_{\theta}(\mathbf{x}_i) \equiv \max_{\theta} \max_{q(\cdot | \mathbf{x}_i), \forall i} \sum_{i=1}^N \int q(\mathbf{z} | \mathbf{x}_i) \log \frac{p_{\theta}(\mathbf{x}_i, \mathbf{z})}{q(\mathbf{z} | \mathbf{x}_i)} d\mathbf{z}$$

and the maximizing distribution is given by  $q^*(\mathbf{z} | \mathbf{x}) = p_{\theta}(\mathbf{z} | \mathbf{x})$

- **Expectation Maximization:** since we know  $q^*$ , if the integral w.r.t.  $\mathbf{z}$  is easy to compute for a fixed  $\theta$ , we can alternate between computing the integral (E-step) and maximizing w.r.t.  $\theta$  (M-step).
- **Variational AutoEncoders:** parameterize  $q(\cdot | x_i)$  with a NN with parameters  $\psi$  that takes  $x_i$  and outputs a distribution  $q_{\psi}(\cdot | x_i)$ , and find  $(\theta, \psi)$  via SGD

$$\max_{\theta} \max_{\psi} \sum_{i=1}^N \int q_{\psi}(\mathbf{z} | x_i) \log \frac{p_{\theta}(x_i, \mathbf{z})}{q_{\psi}(\mathbf{z} | x_i)} d\mathbf{z}$$

# Expectation Maximization

- ML objective

$$\max_{\theta} \sum_{i=1}^N \log p_{\theta}(\mathbf{x}_i) \equiv \max_{\theta} \max_{q(\cdot|\mathbf{x}_i), \forall i} \sum_{i=1}^N \int q(\mathbf{z} | \mathbf{x}_i) \log \frac{p_{\theta}(\mathbf{x}_i, \mathbf{z})}{q(\mathbf{z} | \mathbf{x}_i)} d\mathbf{z}$$

- Expectation Maximization alternates between two steps ( $k$ : iteration)

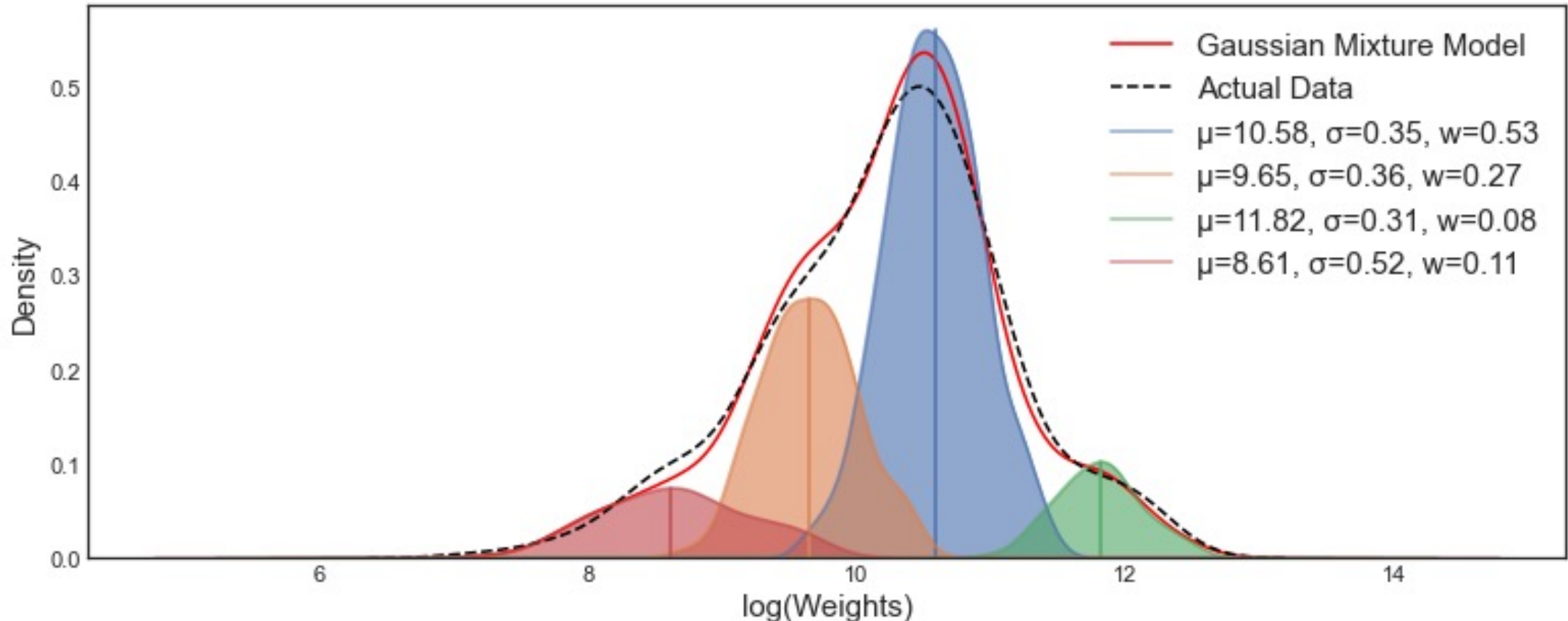
- **E-step:**  $q^k(\mathbf{z}|\mathbf{x}_i) = p_{\theta_k}(\mathbf{z}|\mathbf{x}_i)$  maximize w.r.t.  $q$  with  $\theta$  fixed
- **E-step:**  $Q(\theta, \theta_k) = \sum_{i=1}^N \int_{\mathbf{z}} q^k(\mathbf{z}|\mathbf{x}_i) \log p_{\theta}(\mathbf{x}_i, \mathbf{z}) d\mathbf{z}$  integrate w.r.t.  $q$  with  $\theta$  fixed
- **M-step:**  $\theta^{k+1} = \operatorname{argmax}_{\theta} Q(\theta, \theta_k)$  maximize w.r.t.  $\theta$  with  $q$  fixed

- Examples

- For a mixture of Gaussians, E & M steps are closed-form (next slide)
- Often E-step can be done by sampling (MCMC) and M-step can be done by optimization (SGD)

# E.g.: EM for Gaussian Mixture Model

- **Mixture Model:**  $p_{\theta}(x) = \sum_{j=1}^n p(x | z = j)p(z = j) = \pi_1 p_{\theta_1}(x) + \pi_2 p_{\theta_2}(x) + \dots + \pi_n p_{\theta_n}(x)$ 
  - $\pi_j > 0$ : prior probability of drawing a point from the  $j$ -th model;  $\sum_{j=1}^n \pi_j = 1$
  - $p_{\theta_j} = \mathcal{N}(\mu_j, \Sigma_j)$ .  $\theta_i = (\mu_j, \Sigma_j)$ : mean and covariance of the  $j$ -th Gaussian distribution
  - $\theta = (\theta_1, \dots, \theta_n, \pi_1, \dots, \pi_n)$ : the parameters of the mixture model



# E.g.: EM for Gaussian Mixture Model

- **Mixture Model:**  $p_{\theta}(\mathbf{x}) = \sum_{j=1}^n p(\mathbf{x} \mid z = j)p(z = j) = \pi_1 p_{\theta_1}(\mathbf{x}) + \pi_2 p_{\theta_2}(\mathbf{x}) + \dots + \pi_n p_{\theta_n}(\mathbf{x})$ 
  - $\pi_j > 0$ : prior probability of drawing a point from the  $j$ -th model;  $\sum_{j=1}^n \pi_j = 1$
  - $p_{\theta_j} = \mathcal{N}(\mu_j, \Sigma_j)$ .  $\theta_j = (\mu_j, \Sigma_j)$ : mean and covariance of the  $j$ -th Gaussian distribution
  - $\theta = (\theta_1, \dots, \theta_n, \pi_1, \dots, \pi_n)$ : the parameters of the mixture model
- **Goal:** estimate  $\theta$  from  $N$  i.i.d. samples  $x_1, \dots, x_N$  from  $p_{\theta}$  using EM
  - For  $i = 1, \dots, N$ , let  $z_i = j$  if  $x_i$  belongs to class  $j$ , and let  $q_{ij} = q(z_i = j \mid x_i)$

$$\max_{\theta} \sum_{i=1}^N \log p_{\theta}(x_i) = \max_{\theta} \max_{q_{ij} \geq 0: \sum_j q_{ij} = 1, \forall i} \sum_{i=1}^N \sum_{j=1}^n q_{ij} \log \frac{p_{\theta}(x_i, z_i = j)}{q_{ij}}$$

- **E-step:** compute  $q_{ij}^k = p_{\theta^k}(z_i = j \mid x_i) = \frac{p_{\theta^k}(x_i \mid z_i = j)p_{\theta^k}(z_i = j)}{p_{\theta^k}(x_i)} = \frac{p_{\theta_j^k}(x_i)\pi_j^k}{\sum_{j=1}^n p_{\theta_j^k}(x_i)\pi_j^k}$

# E.g.: EM for Gaussian Mixture Model

- **M-step:**  $\theta^{k+1} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^N \sum_{j=1}^n q_{ij}^k \log \frac{p_{\theta}(x_i, z_i=j)}{q_{ij}^k} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^N \sum_{j=1}^n q_{ij}^k \log p_{\theta}(x_i, z_i = j)$ 
  - If we spell out  $\theta$  and  $p_{\theta}(x_i, z_i = j)$ , the above becomes  $\underset{\{\theta_j\}_{j=1}^n, \{\pi_j\}_{j=1}^n}{\operatorname{argmax}} \sum_{i=1}^N \sum_{j=1}^n q_{ij}^k \log (\pi_j \cdot p_{\theta_j}(x_i))$
- For  $\pi_j$ 's:  $\underset{\{\pi_j\}_{j=1}^n: \sum_{j=1}^n \pi_j = 1}{\operatorname{max}} \sum_{i=1}^N \sum_{j=1}^n q_{ij}^k \log(\pi_j)$
- To analyze the solution of a constrained optimization problem, we use the method of Lagrange multipliers:

$$L(\{\pi_j\}_j, \lambda) = \sum_{i=1}^N \sum_{j=1}^n q_{ij}^k \log(\pi_j) - \lambda(\sum_{j=1}^n \pi_j - 1)$$

- $\frac{\partial L}{\partial \lambda} = 0 \Leftrightarrow \sum_{j=1}^n \pi_j = 1$
- $\forall j: \frac{\partial L}{\partial \pi_j} = 0 \Leftrightarrow \sum_{i=1}^N q_{ij}^k \frac{1}{\pi_j} - \lambda = 0$

- The solution is  $\lambda = \sum_{i=1}^N \sum_{j=1}^n q_{ij}^k, \pi_j = \frac{\sum_{i=1}^N q_{ij}^k}{\sum_{i=1}^N \sum_{j=1}^n q_{ij}^k} =: \pi_j^{k+1}$



# E.g.: EM for Gaussian Mixture Model

- **M-step:**  $\theta^{k+1} = \operatorname{argmax}_{\theta} \sum_{i=1}^N \sum_{j=1}^n q_{ij}^k \log \frac{p_{\theta}(x_i, z_i=j)}{q_{ij}^k} = \operatorname{argmax}_{\theta} \sum_{i=1}^N \sum_{j=1}^n q_{ij}^k \log p_{\theta}(x_i, z_i = j)$ 
  - If we spell out  $\theta$  and  $p_{\theta}(x_i, z_i = j)$ , the above becomes  $\operatorname{argmax}_{\{\theta_j\}_{j=1}^n, \{\pi_j\}_{j=1}^n} \sum_{i=1}^N \sum_{j=1}^n q_{ij}^k \log (\pi_j \cdot p_{\theta_j}(x_i))$
- For  $\theta_j$ 's:  $\max_{\mu_j, \Sigma_j} \left[ \sum_{i=1}^N q_{ij}^k \left( -\frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_j)^{\top} \Sigma_j^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_j) - \frac{1}{2} \log \det(\Sigma_j) \right) \right] =: \mathcal{L}(\boldsymbol{\mu}_j, \Sigma_j)$
- $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_j} = \sum_{i=1}^N q_{ij}^k \Sigma_j^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_j) = \Sigma_j^{-1} \left( \sum_{i=1}^N q_{ij}^k \mathbf{x}_i - \left( \sum_{i=1}^N q_{ij}^k \right) \boldsymbol{\mu}_j \right).$
- Setting it to 0  $\Rightarrow \boldsymbol{\mu}_j = \frac{\sum_{i=1}^N q_{ij}^k \mathbf{x}_i}{\sum_{i=1}^N q_{ij}^k} =: \boldsymbol{\mu}_j^{k+1}$
- $\mathcal{L}(\boldsymbol{\mu}_j, \Sigma_j) = -\frac{1}{2} \operatorname{tr} \left( \Sigma_j^{-1} \sum_{i=1}^N q_{ij}^k (\mathbf{x}_i - \boldsymbol{\mu}_j) (\mathbf{x}_i - \boldsymbol{\mu}_j)^{\top} \right) - \frac{1}{2} \left( \sum_{i=1}^N q_{ij}^k \right) \log \det \Sigma_j$
- Reusing our derivation in MLE for Gaussian:  $\Sigma_j = \frac{\sum_{i=1}^N q_{ij}^k (\mathbf{x}_i - \boldsymbol{\mu}_j^{k+1}) (\mathbf{x}_i - \boldsymbol{\mu}_j^{k+1})^{\top}}{\sum_{i=1}^N q_{ij}^k} =: \Sigma_j^{k+1}$