

Deep Generative Models: Variational Inference

Fall Semester 2025

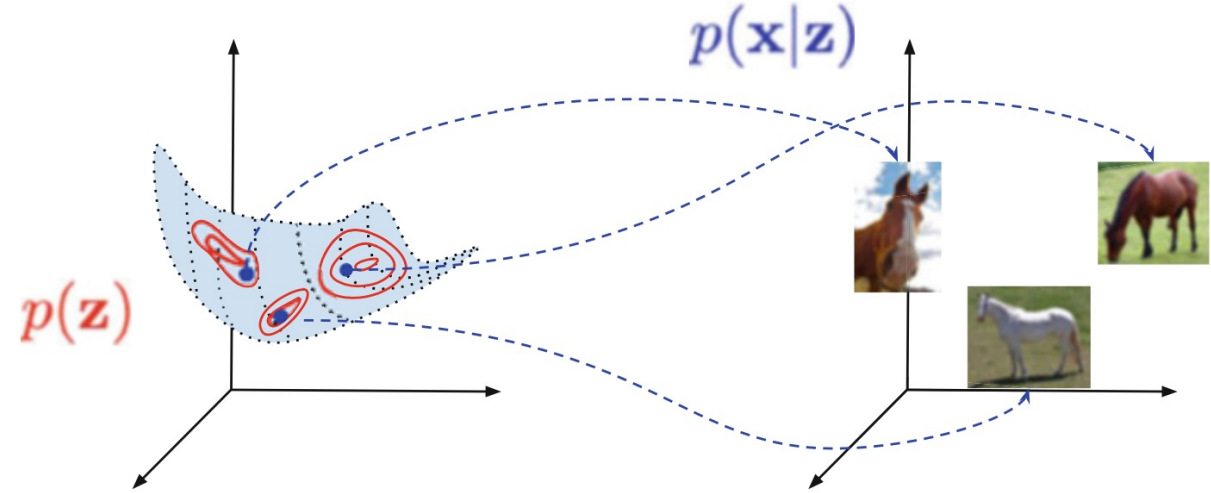
René Vidal

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



Latent Variable Models

- \mathbf{X} = observed variable
- \mathbf{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 $\mathbf{z} \sim \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$ and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{\varepsilon}|\mathbf{0}, \sigma^2 \mathbf{I})$ 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

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

- **Theorem.** Let $\boldsymbol{\mu}_N$ and $\boldsymbol{\Sigma}_N$ be, respectively, the ML estimates for the mean and the covariance of the data. Let U_1 be the matrix with the top d eigenvectors of $\boldsymbol{\Sigma}_N$, Λ_1 be the matrix with the corresponding top d eigenvalues, and λ_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

$$\mathbf{b} = \boldsymbol{\mu}_N, \mathbf{W} = U_1(\Lambda_1 - \sigma^2 \mathbf{I})^{1/2} R \text{ and } \sigma^2 = \frac{1}{D - d} \sum_{i=d+1}^D \lambda_i$$

where $R \in \mathbb{R}^{d \times d}$ 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 $p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$, which gives

$$\max_{\theta} \sum_{i=1}^N \log p_{\theta}(x_i) = \max_{\theta} \sum_{i=1}^N \log \int p_{\theta}(x_i | z)p(z)dz$$

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

$$\max_{\theta} \sum_{i=1}^N \log \sum_j p_{\theta}(x_i | z_j)$$

- 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

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

and the maximizing distribution is given by $p_{\theta}(z|x)$

- **New ML learning objective:**

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

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

Variational Inference

- **New ML learning objective:**

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

- If $q(z|x)$ is easy to evaluate, we can alternate between optimizing w.r.t. θ with $q(z|x)$ fixed and vice versa, leading to the **Expectation Maximization** algorithm
 - Promise: in many cases we will get closed form solutions in each step
- Else, parameterize $q(\cdot |x_i)$ with a NN that takes x_i and outputs a distribution $q_{\phi}(\cdot |x_i)$, where ϕ contains the parameters of the NN, and find (θ, ϕ) via SGD
 - Promise: the output posterior typically has a small variance => Monte Carlo is a good approximation of the integral with respect to z

Variational Inference

- **New ML learning objective:**

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

- 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

$$\log p_{\theta}(x) = \int q(z|x) \log p_{\theta}(x) dz = \int q(z|x) \log \frac{p_{\theta}(x,z)}{p_{\theta}(z|x)} dz$$

$$= \int q(z|x) \log \frac{p_{\theta}(x,z)}{q(z|x)} \frac{q(z|x)}{p_{\theta}(z|x)} dz$$

$$= \boxed{\int q(z|x) \log \frac{p_{\theta}(x,z)}{q(z|x)} dz} + \boxed{\int q(z|x) \log \frac{q(z|x)}{p_{\theta}(z|x)} dz}$$

Evidence Lower Bound (ELBO) $\text{KL}[q(z|x) || p_{\theta}(z|x)]$

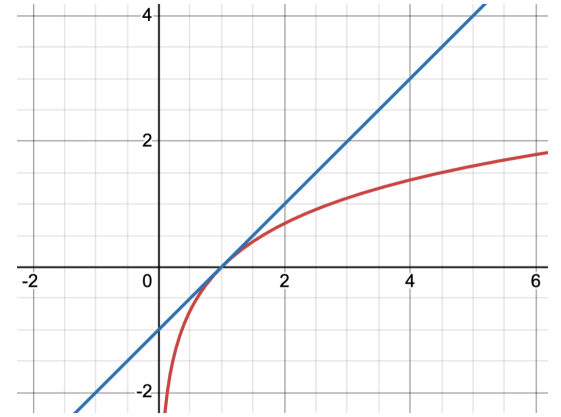
$$\begin{aligned} \text{Thus, } \max_{q(\cdot|x)} \int q(z|x) \log \frac{p_{\theta}(x,z)}{q(z|x)} dz &= \max_{q(\cdot|x)} \log p_{\theta}(x) - \text{KL}[q(z|x) || p_{\theta}(z|x)] \\ &= \log p_{\theta}(x) - \min_{q(\cdot|x)} \text{KL}[q(z|x) || p_{\theta}(z|x)] = \log p_{\theta}(x) \end{aligned}$$

- The last step follows because the $\text{KL}(q || p) \geq 0$ and $\text{KL}(q || p) = 0$ 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: $\text{KL}(q, p) = \int q(x) \log \frac{q(x)}{p(x)} dx$

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

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

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

$$= -1 + 1 = 0$$

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

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

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

- **Expectation Maximization:** If $q(z|x)$ is easy to evaluate, we can alternate between optimizing w.r.t. θ with $q(z|x)$ fixed and vice versa
- **Variational AutoEncoders:** parameterize $q(\cdot|x_i)$ with a NN with parameters ψ that takes x_i and outputs a distribution $q_{\psi}(\cdot|x_i)$, and find (θ, ψ) via SGD

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

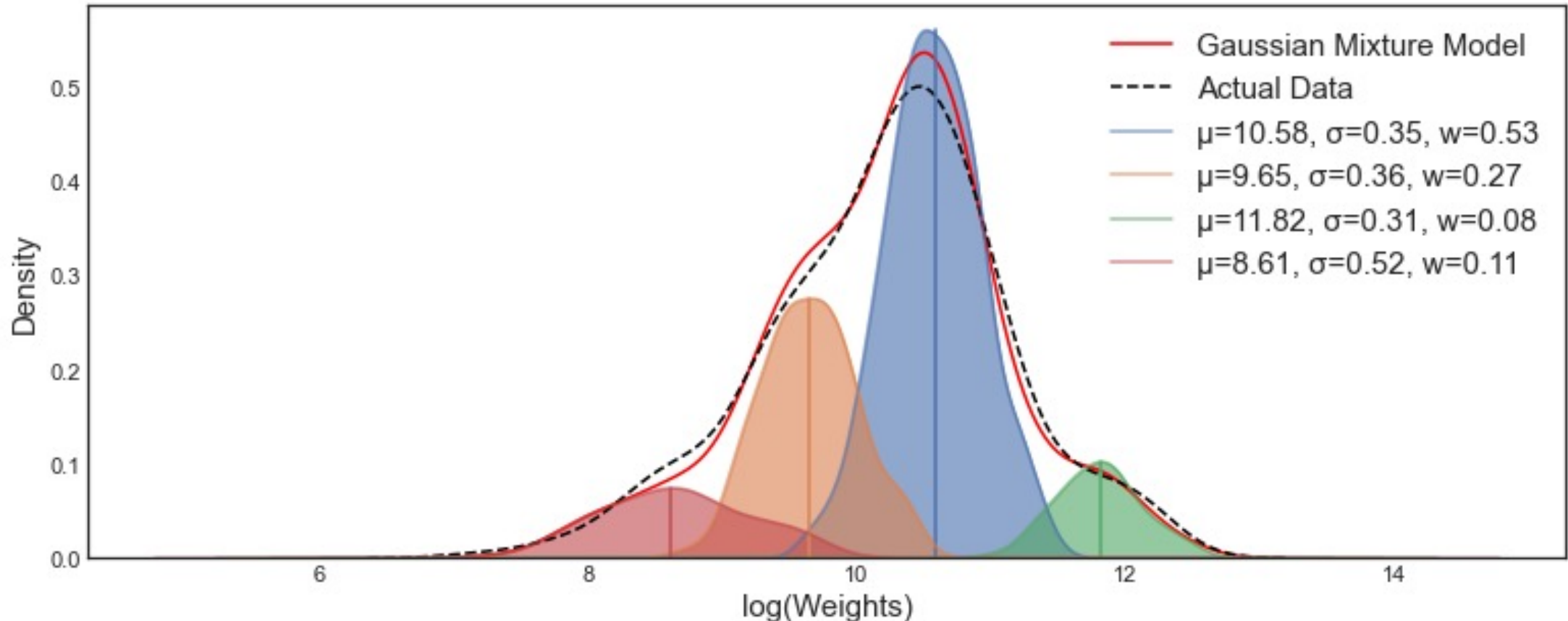
Expectation Maximization

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

- Expectation Maximization alternates between two steps (k : iteration)
- E-step: $q^k(z|x_i) = p_{\theta_k}(z|x_i)$ maximizing w.r.t. q with θ fixed
- M-step: $\theta^{k+1} = \operatorname{argmax}_{\theta} \sum_{i=1}^N \int_{\mathbf{z}} q^k(z|x_i) \log p_{\theta}(x_i, z) dz$ maximizing w.r.t. θ 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:** $p_{\theta}(x) = \sum_{j=1}^n p(x | z = j)p(z = j) = \pi_1 p_{\theta_1}(x) + \pi_2 p_{\theta_2}(x) + \dots + \pi_n p_{\theta_n}(x)$
 - $\pi_j > 0$: prior probability of drawing a point from the j -th model; $\sum_{j=1}^n \pi_j = 1$
 - $p_{\theta_j} = \mathcal{N}(\mu_j, \Sigma_j)$. $\theta_j = (\mu_j, \Sigma_j)$: 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 θ from N i.i.d. samples x_1, \dots, x_N from p_{θ} using EM
 - For $i = 1, \dots, N$, let $z_i = j$ if x_i belongs to class j , and let $q_{ij} = q(z_i = j | x_i)$

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

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

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 θ 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 π_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 θ 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 θ_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: $\Sigma_j = \frac{\sum_{i=1}^N q_{ij}^k (\mathbf{x}_i - \boldsymbol{\mu}_j^{k+1}) (\mathbf{x}_i - \boldsymbol{\mu}_j^{k+1})^{\top}}{\sum_{i=1}^N q_{ij}^k} =: \Sigma_j^{k+1}$