Gaussian Mixture Model and Applications with Expectation-Maximization

KK Thuwajit "KKuRoMa" and Khalid Al-Raisi "Standard Deviation Male"

April 28, 2025

Contents

Preface

We aim to establish the mathematical derivations that will be use throughout the document here.

Theorem 1 (MLE of Categorical Sampling)**.** *Given $y_1, y_2, \ldots, y_n$ are sampled IID from $\{1, 2, \ldots, k\}$ such that $\mathbb{P}(y_i = j) = \phi_j$, $\sum_{j=1}^{k} = \phi_j$, the MLE for ϕ_i is given by*

$$\hat{\phi}_j = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{y_i = j\}$$

Proof. The likelihood of the parameters $\phi_1, \phi_2, \ldots, \phi_k$ given observations $y_1, y_2, \ldots, y_n$ can be expressed as

$$f(\phi_1, \phi_2, \ldots, \phi_k) = \prod_{i=1}^{n} \phi_{y_i} = \prod_{j=1}^{k} \phi_j^{\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{y_i = j\}}$$

and consequently the log-likelihood

$$g(\phi_1, \phi_2, \ldots, \phi_k) = \frac{1}{n} \sum_{j=1}^{k} \log(\phi_j) \left(\sum_{i=1}^{n} \mathbb{I}\{y_i = j\} \right)$$

we are able to solve for the MLE using Lagrangian multipliers. Consider the objective function

$$\mathcal{L}(\phi_1, \phi_2, \ldots, \phi_k, \lambda) = \sum_{j=1}^{k} \log(\phi_j) \left(\sum_{i=1}^{n} \mathbb{I}\{y_i = j\} \right) + \lambda \left(1 - \sum_{j=1}^{k} \phi_j \right)$$

Taking the partial derivative for the gradient

$$\frac{\partial}{\partial \phi_l} \mathcal{L}(\phi_1, \phi_2, \ldots, \phi_k, \lambda) = \frac{\sum_{i=1}^{n} \mathbb{I}\{y_i = l\}}{\phi_l} - \lambda$$

and the second partial derivative for the hessian

$$\frac{\partial^2}{\partial \phi_l^2} \mathcal{L}(\phi_1, \phi_2, \ldots, \phi_k, \lambda) = -\frac{\sum_{i=1}^{n} \mathbb{I}\{y_i = l\}}{\phi_l^2} \text{ and } \frac{\partial}{\partial \phi_l} \frac{\partial}{\partial \phi_m} \mathcal{L}(\phi_1, \phi_2, \ldots, \phi_k, \lambda) = 0$$

We can see that the hessian is negative semi-definite, meaning setting the gradient as $\mathbf{0}$ or equivalently $\phi_l = \frac{\sum_{i=1}^{n} \mathbb{I}\{y_i = l\}}{\lambda}$ yields the maximum. Since

$$1 = \sum_{j=1}^{k} \phi_j = \sum_{j=1}^{k} \frac{\sum_{i=1}^{n} \mathbb{I}\{y_i = j\}}{\lambda} = \frac{n}{\lambda}$$

therefore, $\lambda = n$. Put together, the empirical prediction $\hat{\phi}_j = \frac{\sum_{i=1}^{n} \mathbb{I}\{y_i = j\}}{n}$ is the MLE, completing the proof □

Theorem 2 (MLE of Multivariate Gaussian)**.** *Given $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ are sampled IID from $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, the MLE for $\boldsymbol{\mu}$ and Σ are given by*

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \text{ and } \Sigma = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top$$

Proof. The likelihood of the parameters $\boldsymbol{\mu}$ and Σ given observations $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ can be expressed as

$$f(\boldsymbol{\mu}, \Sigma) = \prod_{i=1}^{n} \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \right)$$

and consequently the log-likelihood

$$g(\boldsymbol{\mu}, \Sigma) = -\frac{n}{2} \log |\Sigma| - \frac{nd}{2} \log(2\pi) - \frac{1}{2} \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu})$$

Fancy footnote hehe

To find the MLE, we maximize $g(\boldsymbol{\mu}, \Sigma)$ with respect to $\boldsymbol{\mu}$ and Σ. First, optimizing with respect to $\boldsymbol{\mu}$, we take the gradient and hessian

$$\frac{\partial}{\partial \boldsymbol{\mu}} g(\boldsymbol{\mu}, \Sigma) = \Sigma^{-1} \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu}) \text{ and } \frac{\partial^2}{\partial \boldsymbol{\mu}^2} g(\boldsymbol{\mu}, \Sigma) = -\Sigma^{-1}$$

Since Σ has to be positive semi-definite to be a covariance, so does Σ^{-1}, which implies the gradient to zero or

$$\Sigma^{-1} \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu}) = 0 \implies \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu}) = 0 \implies n\boldsymbol{\mu} = \sum_{i=1}^{n} \mathbf{x}_i \implies \boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$$

yields the maximum. Next, optimizing with respect to Σ, plugging $\hat{\boldsymbol{\mu}}$ back in, the log-likelihood simplifies to

$$g(\boldsymbol{\mu}, \Sigma) = -\frac{n}{2} \log |\Sigma| - \frac{1}{2} \sum_{i=1}^{n} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top \Sigma^{-1} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})$$

Here, we use the trace trick:

$$-\frac{n}{2} \log |\Sigma| - \frac{1}{2} \sum_{i=1}^{n} \operatorname{Tr}\left((\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top \Sigma^{-1} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}) \right)$$

Taking the derivative with respect to Σ^{-1} using the these identities

$$\frac{\partial}{\partial \Sigma^{-1}} \log |\Sigma| = \Sigma \quad \text{and} \quad \frac{\partial}{\partial \Sigma^{-1}} \operatorname{tr}(A\Sigma^{-1}) = -A$$

we obtain

$$\frac{\partial}{\partial \Sigma^{-1}} g(\boldsymbol{\mu}, \Sigma) = \frac{n}{2} \Sigma - \frac{1}{2} \sum_{i=1}^{n} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top$$

Setting the gradient to zero, we get

$$n\Sigma = \sum_{i=1}^{n} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top \implies \Sigma = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^\top$$

Put together, the empirical predictions are indeed the MLE, thus completing the proof. □

Definition 1 (Kullback–Leibler (KL) Divergence)**.** The Kullback–Leibler (KL) divergence between two probability distributions p and q over a continuous domain is defined as

$$\mathrm{KL}(p \| q) = \mathbb{E}_{\mathbf{x} \sim p} \left[\log p(\mathbf{x}) - \log q(\mathbf{x}) \right] = \int p(\mathbf{x}) \log \left(\frac{p(\mathbf{x})}{q(\mathbf{x})} \right) d\mathbf{x}$$

It measures the extent to which the distribution p differs from the distribution q. It's called a divergence as $\mathrm{KL}(p\|q)$ and $\mathrm{KL}(q\|p)$ are not necessarily equal.

Theorem 3 (KL Divergence of Multivariate Gaussians)**.** *Given two multivariate Gaussian distributions $p = \mathcal{N}(\boldsymbol{\mu}_p, \Sigma_p)$ and $q = \mathcal{N}(\boldsymbol{\mu}_q, \Sigma_q)$, the KL divergence between them is*

$$\mathrm{KL}(p \| q) = \frac{1}{2} \left(\log \frac{|\Sigma_q|}{|\Sigma_p|} - d + \operatorname{Tr}(\Sigma_q^{-1} \Sigma_p) + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^\top \Sigma_q^{-1} (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p) \right)$$

Proof. By definition, the KL divergence is

$$\mathrm{KL}(p \| q) = \mathbb{E}_{\mathbf{x} \sim p} \left[\log p(\mathbf{x}) - \log q(\mathbf{x}) \right]$$

Using the density of a multivariate Gaussian, we have

$$\log p(\mathbf{x}) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_p| - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_p)^\top \Sigma_p^{-1} (\mathbf{x} - \boldsymbol{\mu}_p)$$

and

$$\log q(\mathbf{x}) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log |\Sigma_q| - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_q)^\top \Sigma_q^{-1} (\mathbf{x} - \boldsymbol{\mu}_q)$$

Fancy footnote hehe

Substituting into the KL formula, the constant $-\frac{d}{2}\log(2\pi)$ cancels out, giving

$$\mathrm{KL}(p\|q) = \mathbb{E}_{\mathbf{x}\sim p}\left[-\frac{1}{2}\log|\Sigma_p| + \frac{1}{2}\log|\Sigma_q| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_p)^\top\Sigma_p^{-1}(\mathbf{x}-\boldsymbol{\mu}_p) + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_q)^\top\Sigma_q^{-1}(\mathbf{x}-\boldsymbol{\mu}_q)\right]$$

Grouping constants:

$$\mathrm{KL}(p\|q) = \frac{1}{2}\left(\log\frac{|\Sigma_q|}{|\Sigma_p|} + \mathbb{E}_{\mathbf{x}\sim p}\left[(\mathbf{x}-\boldsymbol{\mu}_q)^\top\Sigma_q^{-1}(\mathbf{x}-\boldsymbol{\mu}_q) - (\mathbf{x}-\boldsymbol{\mu}_p)^\top\Sigma_p^{-1}(\mathbf{x}-\boldsymbol{\mu}_p)\right]\right)$$

Now, expand each quadratic form separately. First,

$$(\mathbf{x}-\boldsymbol{\mu}_q)^\top\Sigma_q^{-1}(\mathbf{x}-\boldsymbol{\mu}_q) = (\mathbf{x}-\boldsymbol{\mu}_p+\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^\top\Sigma_q^{-1}(\mathbf{x}-\boldsymbol{\mu}_p+\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)$$

Expanding the square,

$$= (\mathbf{x}-\boldsymbol{\mu}_p)^\top\Sigma_q^{-1}(\mathbf{x}-\boldsymbol{\mu}_p) + 2(\mathbf{x}-\boldsymbol{\mu}_p)^\top\Sigma_q^{-1}(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q) + (\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^\top\Sigma_q^{-1}(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)$$

Taking expectation under $\mathbf{x}\sim p$, and using that $\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu}_p$ and $\mathbb{E}[(\mathbf{x}-\boldsymbol{\mu}_p)(\mathbf{x}-\boldsymbol{\mu}_p)^\top] = \Sigma_p$, we get

$$\mathbb{E}\left[(\mathbf{x}-\boldsymbol{\mu}_p)^\top\Sigma_q^{-1}(\mathbf{x}-\boldsymbol{\mu}_p)\right] = \mathrm{Tr}(\Sigma_q^{-1}\Sigma_p)$$

and

$$\mathbb{E}\left[2(\mathbf{x}-\boldsymbol{\mu}_p)^\top\Sigma_q^{-1}(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)\right] = 0$$

because the expectation of $\mathbf{x}-\boldsymbol{\mu}_p$ is zero. The third term is constant and remains

$$(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)^\top\Sigma_q^{-1}(\boldsymbol{\mu}_p-\boldsymbol{\mu}_q)$$

Similarly, for the second expectation

$$\mathbb{E}\left[(\mathbf{x}-\boldsymbol{\mu}_p)^\top\Sigma_p^{-1}(\mathbf{x}-\boldsymbol{\mu}_p)\right] = \mathrm{Tr}(\Sigma_p^{-1}\Sigma_p) = d$$

Putting everything together:

$$\mathrm{KL}(p\|q) = \frac{1}{2}\left(\log\frac{|\Sigma_q|}{|\Sigma_p|} - d + \mathrm{Tr}(\Sigma_q^{-1}\Sigma_p) + (\boldsymbol{\mu}_q-\boldsymbol{\mu}_p)^\top\Sigma_q^{-1}(\boldsymbol{\mu}_q-\boldsymbol{\mu}_p)\right)$$

thus completing the proof. □

1. Gaussian Mixture Model

1.1 Defining Gaussian Mixture

Definition 2 (Gaussian Mixture Problem)**.** For $j = 1, 2, \ldots, k$ suppose there exists some fixed, hidden parameters $\phi_j \in \mathbb{R}, \boldsymbol{\mu}_j \in \mathbb{R}^d, \Sigma_j \in \mathbb{R}^{d\times d}$ such that $\sum_{j=1}^k \phi_j = 1$ and Σ_j is a valid covariance matrix, we consider the following procedure

- For $i = 1, 2, \ldots, n$ sample IID z_i from $\{1, 2, \ldots, k\}$ such that $\mathbb{P}(z_i) = \phi_j$
- For $i = 1, 2, \ldots, n$ sample IID $\mathbf{x}_i$ from $\mathcal{N}(\boldsymbol{\mu}_{z_i}, \Sigma_{z_i})$

There are two versions of the problems:

- **Supervised**: for $i = 1, 2, \ldots, n$ we are given $\mathbf{x}_i, z_i$. Find the estimate for $\phi_j, \boldsymbol{\mu}_j, \Sigma_j$
- **Unsupervised**: for $i = 1, 2, \ldots, n$ we are given only $\mathbf{x}_i$. Find the estimate for $\phi_j, \boldsymbol{\mu}_j, \Sigma_j$

This is assuming that there exists k different Gaussian clusters, and $\mathbf{x}_i$ is sampled from one of them with probability specified by ϕ_j. Notice that the supervised version of the problem essentially boils down to running the regular Multivariate Gaussian problem for each class z_i can take. We are more interested in the unsupervised problem, as the MLE could not be solved in a closed form.

1.2 Potential Applications of Gaussian Mixture

We present two applications of the unsupervised version of the Gaussian Mixture defined above. The exact implementations will be explained in later sessions.

- **Unsupervised classification**: the resulting estimate for the Gaussian clusters' mean and covariance allows estimates for the unknown label z_i to be retrieved from $\mathbf{x}_i$.
- **Conditional generation**: the Gaussian Mixture model allows data outside of the given set to be sampled from the resulting Gaussian distributions, conditioned on one of the possible classes.

1.3 Gaussian Mixture as MLE Problem

Given access to only the data vectors $\mathbf{x}_i$ for $i = 1, 2, \ldots, n$, one may model the likelihood of the parameters $\{\phi_j, \boldsymbol{\mu}_j, \Sigma_j\}_{j=1}^k$ as follows:

$$\begin{aligned} f(\{\phi_j, \boldsymbol{\mu}_j, \Sigma_j\}_{j=1}^k) &= \prod_{i=1}^n \sum_{z=1}^k \mathbb{P}(Z = k) p(\mathbf{x}_n; \boldsymbol{\mu}_k, \Sigma_k) \\ &= \prod_{i=1}^n \sum_{z=1}^k \frac{\phi_z}{(2\pi)^{d/2}|\Sigma_z|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_z)^\top \Sigma_z^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_z)\right) \end{aligned}$$

This results in a double summation once a logarithm is taken, particularly with the term inside logarithm being a summation. While a closed form MLE solution doesn't exist for this expression, the use of the EM algorithm explained over the next sections will help us iteratively solve for a good approximate.

2. Expectation-Maximization Algorithms

This section aims to define a class of algorithms known as the Expectation-Maximization (EM) algorithm. These algorithms are designed to provide an approximate solution to MLE problems where, because of an unknown set of "latent" random variables, don't lead to a closed form solution.

Definition 3 (Distribution Learning with Hidden Latent Variables)**.** Suppose there exists

- A distribution $\mathcal{P}$ from $\mathbb{R}^d$ with a joint pdf p and an unknown parameter θ_*
- A distribution $\mathcal{Q}_*$ from $\mathbb{R}^s$ with a pdf q_*.

Note that $\mathcal{P}$ and $\mathcal{Q}_*$ can either be explicitly given or not. Hidden behind the scenes, we sample IID:

- For $i = 1, 2, \ldots, n$, $\mathbf{z}_i \sim \mathcal{Q}_*$ (that is, following the pdf of $q_*(\mathbf{z}_i)$)
- For $i = 1, 2, \ldots, n$, $\mathbf{x}_i|\mathbf{z}_i \sim \mathcal{P}_{\theta_*}$ (that is, following the pdf of $p(\mathbf{x}_i|\mathbf{z}_i; \theta_*)$)

Given access to only $\mathbf{x}_i$ for $i = 1, 2, \ldots, n$, find an estimate for θ_* and q_* (the latter if not explicitly given).

2.1 The Evidence Lower Bound (ELBO)

Like the expression for Gaussian Mixture Models, we may model the likelihood of θ and q as follows:

$$\begin{aligned}
f(\theta, q) &= \prod_{i=1}^{n} \int p(\mathbf{x}_i, \mathbf{z}; \theta) d\mathbf{z} \\
&= \prod_{i=1}^{n} \int q(\mathbf{z}) p(\mathbf{x}_i|\mathbf{z}; \theta) d\mathbf{z} \\
&= \prod_{i=1}^{n} \mathbb{E}_{\mathbf{z}} \left[p(\mathbf{x}_i|\mathbf{z}; \theta) \right] \\
g(\theta, q) &= \log(f(\theta, q)) \\
&= \sum_{i=1}^{n} \log \left(\mathbb{E}_{\mathbf{z}} \left[p(\mathbf{x}_i|\mathbf{z}; \theta) \right] \right) \quad (*) \\
&\geq \sum_{i=1}^{n} \mathbb{E}_{\mathbf{z}} \left[\log \left(p(\mathbf{x}_i|\mathbf{z}; \theta) \right) \right] \quad (**) \\
&= \sum_{i=1}^{n} \int q(\mathbf{z}) \log(p(\mathbf{x}_i|\mathbf{z}; \theta)) d\mathbf{z}
\end{aligned}$$

We're able to go from $(*)$ to $(**)$ using Jensen's inequality, as log is a convex function. This allows us to minimize the lower bound of $g(\theta, q)$ which indirectly optimizes the log-likelihood itself. This lower bound term is therefore dubbed the **evidence lower bound** (ELBO):

$$\text{ELBO}(\mathbf{x}; \theta, q) := \int q(\mathbf{z}) \log(p(\mathbf{x}|\mathbf{z}; \theta)) d\mathbf{z}$$

Finding the optimal prior: For the tightest bound with Jensen inequality to hold, the expectation must be taken over a constant, meaning $p(\mathbf{x}|\mathbf{z}; \theta) = c$ over some constant c. By Bayes' law,

$$\begin{aligned}
p(\mathbf{x}|\mathbf{z}; \theta) &= c \\
\frac{p(\mathbf{x}, \mathbf{z}; \theta)}{q(\mathbf{z})} &= c \\
q(\mathbf{z}) &\propto p(\mathbf{x}, \mathbf{z}; \theta)
\end{aligned}$$

And since $\int_{\mathbf{z}} q(\mathbf{z})d\mathbf{z} = 1$,

$$\begin{aligned} q(\mathbf{z}) &= \frac{q(\mathbf{z})}{\int_{\mathbf{z}} q(\mathbf{z})d\mathbf{z}} \\ &= \frac{p(\mathbf{x}, \mathbf{z}; \theta)}{\int_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}; \theta)} \\ &= \frac{p(\mathbf{x}, \mathbf{z}; \theta)}{p(\mathbf{x}, \theta)} \\ &= p(\mathbf{z}|\mathbf{x}; \theta) \end{aligned}$$

Therefore, $p(\mathbf{z}|\mathbf{x}; \theta)$ is the optimal choice for $q(\mathbf{z})$. Notice that this choice is dependent on $\mathbf{x}$. As such, for each $\mathbf{x}_i$, a choice q_i is made as $p(\mathbf{z}|\mathbf{x}_i; \theta)$.

Definition 4 (EM Learning Objective)**.** We may isolate θ from the expression using the optimal q_i as follows:

$$\ell(\theta) = \sum_{i=1}^{n} g(\theta, q_i) \geq \sum_{i=1}^{n} \text{ELBO}(\mathbf{x}_i; \theta, q_i) = \sum_{i=1}^{n} \int q_i(\mathbf{z}) \log(p(\mathbf{x}|\mathbf{z}; \theta))d\mathbf{z} \quad \text{when} \quad q_i(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}_i; \theta)$$

2.2 Defining EM Algorithms

The EM algorithm aims to minimize the learning objective defined above, which, if we recall, is the lower bound of the log-likelihood of a parameter θ, given q_i are chosen optimally.

Definition 5 (EM Algorithm)**.** The algorithm takes the scenario defined in the "Distribution Learning with Hidden Latent Variables" problem and carries the following steps

- **Expectation (E) Step:** compute $q_i^{(t)}(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}_i; \theta^{(t)})$ for $i = 1, 2, \ldots, n$
- **Maximization (M) Step:** compute

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_{i=1}^{n} \text{ELBO}(\mathbf{x}_i; \theta, q_i^{(t)}) = \arg\max_{\theta} \sum_{i=1}^{n} \int q_i^{(t)}(\mathbf{z}) \log(p(\mathbf{x}|\mathbf{z}; \theta))d\mathbf{z}$$

With an arbitrary initial value for $\theta^{(0)}$, the algorithm repeats the E and M steps until convergence.

A primary concern with such an algorithm is the convergence, which will promptly be proven

Theorem 4 (EM Algorithm Convergence)**.** *The EM Algorithm defined above updates $\ell(\theta^{(t)})$ monotonically, that is, $\ell(\theta^{(t+1)}) \geq \ell(\theta^{(t)})$.*

Proof. First, recall the the ELBO is defined such that $\ell(\theta) \geq \sum_{i=1}^{n} \text{ELBO}(\mathbf{x}_i; \theta, q_i)$ for any prior q_i. Furthermore, the choice for $q_i^{(t)}$ was made to ensure that Jensen's inequality holds with equality, meaning $\ell(\theta^{(t)}) = \sum_{i=1}^{n} \text{ELBO}(\mathbf{x}_i; \theta^{(t)}, q_i^{(t)})$. Therefore,

$$\begin{aligned} \ell(\theta^{(t+1)}) &\geq \sum_{i=1}^{n} \text{ELBO}(\mathbf{x}_i; \theta^{(t+1)}, q_i^{(t)}) && \text{ELBO's definition} \\ &\geq \sum_{i=1}^{n} \text{ELBO}(\mathbf{x}_i; \theta^{(t)}, q_i^{(t)}) && \theta^{(t+1)}\text{'s definition} \\ &= \ell(\theta^{(t)}) && \text{Jensen's Inequality Equality Case} \end{aligned}$$

As desired. □

2.3 EM Algorithm for Gaussian Mixture Models

We revisit the problem of Gaussian Mixture Models, now armed with the knowledge of the EM algorithm. We first note that the Gaussian Mixture Model problem is a specific case of the "Distribution Learning with Hidden Latent Variables" problem defined earlier. In particular, $\theta = \{\boldsymbol{\mu}_j, \Sigma_j\}_{j=1}^{k}$, p is the Gaussian Mixture pdf, and q is the proportion of each possible class of z showing up. The EM Algorithm becomes

- **Expectation (E) Step:** $q_i^{(t)}(z) = p(z|\mathbf{x}_i;\theta^{(t)})$ for $i = 1, 2, \ldots, n$
- **Maximization (M) Step:** We first use Bayes's rule to re-write the terms

$$\begin{aligned}\theta^{(t+1)} &= \arg\max_\theta \sum_{i=1}^n \sum_{z=1}^k q_i^{(t)}(z) \log(p(\mathbf{x}|z;\boldsymbol{\mu}_z,\Sigma_z,\phi_z)) \\ &= \arg\max_\theta \sum_{i=1}^n \sum_{z=1}^k q_i^{(t)}(z) \log\left(\frac{p(\mathbf{x},z;\boldsymbol{\mu}_z,\Sigma_z,\phi_z)}{q_i^{(t)}(z)}\right) \\ &= \arg\max_\theta \sum_{i=1}^n \sum_{z=1}^k q_i^{(t)}(z) \log\left(\frac{p(\mathbf{x}|z;\boldsymbol{\mu}_z,\Sigma_z)p(z,\phi_z)}{q_i^{(t)}(z)}\right) \\ &= \arg\max_\theta \sum_{i=1}^n \sum_{z=1}^k q_i^{(t)}(z) \log\left(\frac{p(\mathbf{x}|z;\boldsymbol{\mu}_z,\Sigma_z)\phi_z}{q_i^{(t)}(z)}\right)\end{aligned}$$

  therefore, taking the log and removing constant terms

$$\theta^{(t+1)} = \arg\max_\theta \sum_{i=1}^n \sum_{z=1}^k q_i^{(t)}(z) \left(-\frac{n}{2}\log|\Sigma_z| - \frac{1}{2}\sum_{i=1}^n (\mathbf{x}_i-\boldsymbol{\mu}_z)^\top \Sigma_z^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_z) + \log(\phi_z)\right)$$

Notice that the optimization terms for $\boldsymbol{\mu}_z$ and Σ_z is a linear combination of several log-likelihood terms of regular Gaussian fitting. Indeed, we are able to use the results from the preface to conclude that for $z = 1, 2, \ldots, k$

$$\boldsymbol{\mu}_z = \frac{\sum_{i=1}^n q_i^{(t)}(z)\mathbf{x}_i}{\sum_{i=1}^n q_i^{(t)}(z)} \text{ and } \Sigma_z = \frac{\sum_{i=1}^n q_i^{(t)}(z)(\mathbf{x}_i-\boldsymbol{\mu}_z)(\mathbf{x}_i-\boldsymbol{\mu}_z)^\top}{\sum_{i=1}^n q_i^{(t)}(z)}$$

Furthermore, the optimization terms ϕ_z are a linear combination of several log-likelihood terms of the categorical sampling. Similarly, we can use the results the preface to conclude that for $z = 1, 2, \ldots, k$

$$\phi_z = \frac{1}{n}\sum_{i=1}^n q_i^{(t)}(z)$$

Conclusion: The Gaussian Mixture problem can be seen as a special case for the previously defined "Distribution Learning with Hidden Latent Variables" problem. Using the EM algorithm, we're able to derive an iterative procedure that approximates $\boldsymbol{\mu}_z, \Sigma_z, \phi_z$ in a closed form.

2.4 Extending the EM Algorithm

The Variational Autoencoder (VAE) is a popular extension of the EM Algorithm. Particularly, the **E-step** of most EM algorithm involves computing $q_i^{(t)}(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}_i;\theta^{(t)})$ for $i = 1, 2, \ldots, n$, which is often an **intractable problem** for most distributions. The VAE replaces the posterior $p(\mathbf{z}|\mathbf{x}_i;\theta^{(t)})$ by a learnable distribution q parametrized by ϕ. Commonly, the two distributions $p(\mathbf{x}|\mathbf{z};\theta)$ (**Decoder**) and $q(\mathbf{z}|\mathbf{x};\phi)$ (**Encoder**) are learned through a neural network with parameters θ and ϕ respectively. Revisiting the ELBO, now reparametrized with θ and ϕ as q is parametrized on the latter,

$$\begin{aligned}\text{ELBO}(\mathbf{x},\theta,\phi) &= \int q(\mathbf{z}|\mathbf{x};\phi)\log\left(\frac{p(\mathbf{x},\mathbf{z};\theta)}{q(\mathbf{z}|\mathbf{x};\phi)}\right)d\mathbf{z} \\ &= \int q(\mathbf{z}|\mathbf{x};\phi)\log\left(\frac{p(\mathbf{x}|\mathbf{z};\theta)p(\mathbf{z};\theta)}{q(\mathbf{z}|\mathbf{x};\phi)}\right)d\mathbf{z} \\ &= \int q(\mathbf{z}|\mathbf{x};\phi)\log(p(\mathbf{x}|\mathbf{z};\theta))d\mathbf{z} - \int q(\mathbf{z}|\mathbf{x};\phi)\log\left(\frac{q(\mathbf{z}|\mathbf{x};\phi)}{p(\mathbf{z};\theta)}\right)d\mathbf{z}\end{aligned}$$

- The first term $\int q(\mathbf{z}|\mathbf{x};\phi)\log(p(\mathbf{x}|\mathbf{z};\theta))d\mathbf{z} = \mathbb{E}_{\mathbf{z}\sim q_\phi}[\log(p(\mathbf{x}|\mathbf{z};\theta))]$ can be derived further under the assumption that $p(\mathbf{x}|\mathbf{z};\theta)$ is Gaussian of some mean $\mu_\theta(\mathbf{z})$ parameterized on θ and a constant diagonal covariance $\sigma^2\mathbf{I}_{d\times d}$

$$\begin{aligned}\log(p(\mathbf{x}|\mathbf{z};\theta)) &= \log\left(\frac{1}{\sigma^d(2\pi)^{d/2}}\exp\left(-\frac{\|\mathbf{x}-\mu_\theta(\mathbf{z})\|_2^2}{2\sigma^2}\right)\right) \\ &= -\|\mathbf{x}-\mu_\theta(\mathbf{z})\|_2^2 + C\end{aligned}$$

Fancy footnote hehe

when the constant terms are grouped to C. As such, optimizing this term is equivalent to minimizing $\mathbb{E}[\|\mathbf{x} - \mu_\theta(\mathbf{z})\|_2^2]$, the euclidean difference between the data point $\mathbf{x}$ and its reconstruction $\mu_\theta(\mathbf{z})$.

- The second term $\int q(\mathbf{z}|\mathbf{x};\phi) \log\left(\frac{q(\mathbf{z}|\mathbf{x};\phi)}{p(\mathbf{z};\theta)}\right) d\mathbf{z}$ is the KL divergence between the two distributions $q(\mathbf{z}|\mathbf{x};\phi)$ and $p(\mathbf{z};\theta)$. We first assume $p(\mathbf{z};\theta)$ follows a standard $s-$dimensional Gaussian pdf. We then assume that $q(\mathbf{z}|\mathbf{x};\phi)$ follows $\mathcal{N}(\mu_\phi(\mathbf{x}), \text{diag}(\sigma_\phi^2(\mathbf{x})))$ (when diag($\mathbf{v}$) takes $\mathbf{v} \in \mathbb{R}^s$ and returns an $s \times s$ diagonal matrix with entries from the basis coefficient of $\mathbf{v}$). Using results from the preface, this KL divergence can be expressed as follows:

$$\text{KL}(q(\mathbf{z}|\mathbf{x};\phi)\|p(\mathbf{z};\theta)) = \frac{1}{2}\sum_{i=1}^{s}\left(\sigma_{\phi,i}^2(\mathbf{x}) + \mu_{\phi,i}^2(\mathbf{x}) - 1 - \log\sigma_{\phi,i}^2(\mathbf{x})\right)$$

  Remark: this is known as the reparametrization trick; the encoder q produces two vectors $\mu_\phi(\mathbf{x})$ and $\sigma_\phi^2(\mathbf{x})$ to parameterize the Gaussian distribution z is then sampled from. Oftentimes, the log-variance is predicted instead to prevent negative-valued predictions.

Conclusion the VAE provides an alternative to EM algorithm's **E-step**, which involves an often intractable term $p(\mathbf{z}|\mathbf{x}_i;\theta^{(t)})$. Instead, the VAE learns two distributions $p(\mathbf{x}|\mathbf{z};\theta)$ (**Decoder**) and $q(\mathbf{z}|\mathbf{x};\phi)$ (**Encoder**) simultaneously to maximize the ELBO. By several assumptions of $p(\mathbf{x}|\mathbf{z};\theta)$, $q(\mathbf{z}|\mathbf{x};\phi)$, and $p(\mathbf{z};\theta)$ normality, the ELBO simplifies to a sum of the reconstruction loss and KL divergence

$$\hat{\theta}, \hat{\phi} = \arg\min_{\theta,\phi} \mathbb{E}[\|\mathbf{x} - \mu_\theta(\mathbf{z})\|_2^2] + \frac{1}{2}\sum_{i=1}^{s}\left(\sigma_{\phi,i}^2(\mathbf{x}) + \mu_{\phi,i}^2(\mathbf{x}) - 1 - \log\sigma_{\phi,i}^2(\mathbf{x})\right)$$

which could be learned through (stochastic) gradient descent. Unlike the EM algorithm, the VAE is only able to improve the lower bound (ELBO) of the log-likelihood, meaning convergence could not be guaranteed.

3. Unsupervised Classification

Definition 6 (Unsupervised Classification)**.** Given data vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ and a set of possible classes $\mathcal{C} = \{1, 2, \ldots, k\}$, we attempt to find $z_1, z_2, \ldots, z_n \in \mathcal{C}$ and parameters θ such that $\prod_{i=1}^{n} p(\mathbf{x}_i, z_i; \theta)$. In other words, the MLE under a known distribution type is maximized.

This definition can seen as a reformulation of the "Distribution Learning with Hidden Latent Variables" where $\mathbf{z}_i$ is limited to a discrete set. By solving for q with EM, we're able to derive the MLE for the classes z_i as follows

$$\begin{aligned}
\hat{z} &= \arg\max_{z \in \mathcal{C}} p(z|\mathbf{x}; \theta) \\
&= \arg\max_{z \in \mathcal{C}} \frac{p(\mathbf{x}|z; \theta)p(z)}{p(\mathbf{x})} && \text{Bayes' Rule} \\
&= \arg\max_{z \in \mathcal{C}} p(\mathbf{x}|z; \theta)p(z) && p(\mathbf{x}) \text{ is constant} \\
&= \arg\max_{z \in \mathcal{C}} p(\mathbf{x}|z; \theta)\hat{q}(z) && \hat{q} \text{ is an estimate of the prior}
\end{aligned}$$

3.1 K-means Clustering

The K-means clustering finds an estimate to the unsupervised classification problem using the following procedure. First assign centroids $\boldsymbol{\mu}_j^{(0)}$ for $j = \{1, 2, \ldots, k\}$ to random points.

-

In this section, we argue that the algorithm above can be seen as a special case of the EM algorithm.

3.2 Gaussian Mixture Classification

3.3 Comparison

4. Conditional Generation

4.1 Pure Gaussian Mixture Models

4.2 Gaussian Mixture Variational Autoencoders

4.3 Comparison