Lecture 9: Gaussian Mixture Model and EM Algorithm

Shukai Gong

1 Generative Model and Discriminative Model

Denote X as the random variable of samples, optionally Y as the random variable of labels. First, we introduce the differences between the *generative model* and *discriminative model*.

	Generative Model	Discriminative Model
Functionality	Capture the mechanism of generating data	Capture the differences between different data points
Principle	Model the data distribution $p(X)$ or the joint distribution $p(X, Y)$ Data distribution $\Rightarrow$ Resample for new data	Model the conditional distribution $p(Y\|X)$
Example	K-means Clustering	Linear/Logistic Regression

Table 1: Generative Model versus Discriminative Model

2 Gaussian Mixture Model

There is another way of thinking about clustering the data points into K clusters: for each cluster $\mathcal{C}_k$, the data points within the cluster are generated from a certain distribution, and there are K distributions in total.

One common modeling measure is to model the distribution as Gaussian distribution. As is shown in Figure 1, we can model the data points from two clusters as a mixture of two **Gaussian** distributions. Each data point $\boldsymbol{x}_i$ from a certain cluster can have both a probability γ_{i1} of belonging to cluster 1 and a probability γ_{i2} of belonging to cluster 2.

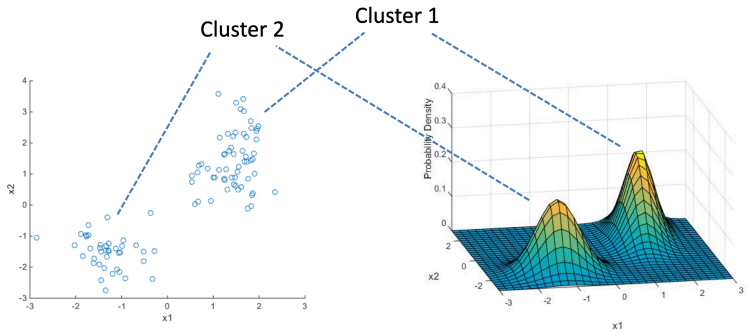

Figure 1: Modelling datapoints from two clusters as a mixture of two Gaussian distributions

In this sense, Gaussian Mixture Model (GMM) provides a **parametric generative model** for data with clustering structure: we not only cares about which cluster does a data point $\boldsymbol{x}_i$ belong to, but also the probability of $\boldsymbol{x}_i$ being generated from each cluster.

Gaussian Mixture Model (GMM)

Suppose that there are K Gaussian distributions defined on the sample space $\mathcal{X} \in \mathbb{R}^D$ corresponding to K clusters. The probability distribution of a random data point $\boldsymbol{x}$ under GMM is given by

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} p(\mathcal{C}_k)p(\boldsymbol{x}|\mathcal{C}_k) = \sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where $w_k = p(\mathcal{C}_k) \geq 0$ is the probability of choosing the k-th distribution, $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = p(\boldsymbol{x}|\mathcal{C}_k)$ is the conditional probability of choosing $\boldsymbol{x}$ from the cluster $\mathcal{C}_k$, modeled as a Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. w_k can be considered as the **mixture coefficient** of the k-th Gaussian distribution satisfying $\sum_{k=1}^{K} w_k = 1$. Our goal is to learn the parameters $\boldsymbol{\Theta} = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ from the data.

2.1 Solving GMM Parameters by MLE

Given an i.i.d. sample of $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, the likelihood function of the dataset is

$$L(\boldsymbol{\Theta}) = p(\boldsymbol{X}|\boldsymbol{\Theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\Theta}) = \prod_{i=1}^{N} \sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\Rightarrow l(\boldsymbol{\Theta}) = \log L(\boldsymbol{\Theta}) = \sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)$$

Our goal is

$$\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \underbrace{\sum_{i=1}^{N} \log \left(\sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right)}_{l(\boldsymbol{\Theta})}$$

However, it's hard to optimize the log-likelihood function directly due to the sum inside the logarithm. Alternatively, we adopt the **Expectation-Maximization (EM) Algorithm** to solve the optimization problem.

2.2 Solving GMM Parameters by Expectation-Maximization (EM) Algorithm

In practice, EM Algorithm is adopted to learn GMM parameters:

EM Algorithm

- **Initialization**: Initialize the parameters $\boldsymbol{\Theta} = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ randomly.
- **E-Step**: Given current parameters $\{w_k^{(t)}, \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}\}_{k=1}^{K}$, calculate **responsibility**, i.e. the probability of the i-th data point $\boldsymbol{x}_i$ belonging to the k-th cluster $\mathcal{C}_k$, denoted as $\gamma_{ik}^{(t)}$

$$\gamma_{ik}^{(t)} = \frac{w_k^{(t)} \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)})}{\sum_{j=1}^{K} w_j^{(t)} \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)})}, \forall n = 1, 2, \ldots, N, k = 1, 2, \ldots, K$$

- **M-Step**: Update the model parameters $\boldsymbol{\Theta} = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ by

$$w_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik}^{(t)}$$

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{i=1}^{N} \gamma_{ik}^{(t)} \boldsymbol{x}_i}{\sum_{i=1}^{N} \gamma_{ik}^{(t)}}$$

$$\boldsymbol{\Sigma}_k^{(t+1)} = \frac{\sum_{i=1}^{N} \gamma_{ik}^{(t)} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t+1)})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(t+1)})^\top}{\sum_{i=1}^{N} \gamma_{ik}^{(t)}}$$

For the explanation of E-step, first, we can introduce a binary latent variable z_k ($z_k = 1$ means $\boldsymbol{x}_i$ belongs to cluster $\mathcal{C}_k$) to represent the cluster that a data point $\boldsymbol{x}_i$ really belongs to. By Bayes Rule, γ_{ik} is essentially the **posterior distribution of cluster $\mathcal{C}_k$ given the data point $\boldsymbol{x}_i$:**

$$\gamma_{ik} = p(z_k = 1|\boldsymbol{x}_i) = \frac{p(z_k = 1)p(\boldsymbol{x}_i|z_k = 1)}{p(\boldsymbol{x}_i)} = \frac{w_k\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} w_j\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Then, instead of maximizing $l(\boldsymbol{\Theta})$, we choose to minimize its **lower bound $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)})$ instead**:

$$\begin{aligned}
l(\boldsymbol{\Theta}) &= \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} w_k\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) \\
&= \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \gamma_{ik}^{(t)} \frac{w_k\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\gamma_{ik}^{(t)}}\right) \\
&\geq \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \log\left(\frac{w_k\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\gamma_{ik}^{(t)}}\right) \equiv Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)}) \qquad \textbf{(by Jensen's Inequality, } \log \textbf{ is concave)}
\end{aligned}$$

Now the optimization problem at the t-th iteration is

$$\left(\boldsymbol{\Theta}^{(t+1)}\right)^* = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)}) = \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \log\left(\frac{w_k\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\gamma_{ik}^{(t)}}\right)$$

Then, in the M-step, we can update the parameters by taking the partial derivative of $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)})$ with respect to $w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ respectively. Specifically, to solve for $w_k^{(t+1)}$,

$$\begin{aligned}
&\max_{\boldsymbol{\Theta}} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \log\left(\frac{w_k\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\gamma_{ik}^{(t)}}\right), \text{ s.t. } \sum_{k=1}^{K} w_k = 1 \\
\iff &\max_{\boldsymbol{\Theta}} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \log\left(w_k\right), \text{ s.t. } \sum_{k=1}^{K} w_k = 1 \\
&\mathcal{L} = \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \log\left(w_k\right) + \lambda\left(1 - \sum_{k=1}^{K} w_k\right) \\
\Rightarrow &\begin{cases} \dfrac{\partial\mathcal{L}}{\partial w_k} = \sum\limits_{i=1}^{N} \dfrac{\gamma_{ik}^{(t)}}{w_k} - \lambda = 0 \\ \sum\limits_{k=1}^{K} w_k = 1 \end{cases} \Rightarrow w_k^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N} \gamma_{ik}^{(t)}
\end{aligned}$$

Indicating that the mixture coefficient w_k is the average of the responsibilities of all data points to cluster $\mathcal{C}_k$. For the update of $\boldsymbol{\mu}_k^{(t+1)}$,

$$\begin{aligned}
&\max_{\boldsymbol{\Theta}} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \log\left(\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) = \max_{\boldsymbol{\Theta}} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \log\left(\frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right)\right) \\
= &\max_{\boldsymbol{\Theta}} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k\right)^\top \boldsymbol{\Sigma}_k^{-1} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_k\right) \\
\Rightarrow &\frac{\partial\mathcal{L}}{\partial\boldsymbol{\mu}_k} = \sum_{i=1}^{N} \gamma_{ik}^{(t)} \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) = 0 \\
\Rightarrow &\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{i=1}^{N} \gamma_{ik}^{(t)} \boldsymbol{x}_i}{\sum_{i=1}^{N} \gamma_{ik}^{(t)}}
\end{aligned}$$

For the update of $\boldsymbol{\Sigma}_k^{(t+1)}$,

$$\max_{\boldsymbol{\Theta}} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \log\left(\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) = \max_{\boldsymbol{\Theta}} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \log\left(\frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)\right)\right)$$

$$= \max_{\boldsymbol{\Theta}} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \left(-\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)\right)$$

$$\Rightarrow \frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = \sum_{i=1}^{N} \gamma_{ik}^{(t)} \left(-\frac{1}{2}\boldsymbol{\Sigma}_k^{-1} + \frac{1}{2}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}\right) = 0$$

$$\Rightarrow \boldsymbol{\Sigma}_k^{(t+1)} = \frac{\sum_{i=1}^{N} \gamma_{ik}^{(t)} (\boldsymbol{x}_i-\boldsymbol{\mu}_k^{(t+1)})(\boldsymbol{x}_i-\boldsymbol{\mu}_k^{(t+1)})^\top}{\sum_{i=1}^{N} \gamma_{ik}^{(t)}}$$

2.3 Mechanism of EM Algorithm

Essentially, the EM algorithm is used to find local maximum likelihood parameters of a statistical model that involve **unobserved latent variables**. In GMMs, the latent variables are the real cluster assignments of the data points. To illustrate the mechanism of EM Algorithm, denote $q(\boldsymbol{z})$ as the distribution of the latent variables $\boldsymbol{z}$, and we can decompose $l(\boldsymbol{\Theta}) = \log p(\boldsymbol{X}|\boldsymbol{\Theta})$ as

$$\begin{aligned}
l(\boldsymbol{\Theta}) = \log p(\boldsymbol{X}|\boldsymbol{\Theta}) &= \int_{\boldsymbol{z}} q(\boldsymbol{z}) \log p(\boldsymbol{X}|\boldsymbol{\Theta}) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q(\boldsymbol{z}) \log\left(\frac{p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\Theta})}{p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta})}\right) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q(\boldsymbol{z}) \log\left(\frac{p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\Theta})}{q(\boldsymbol{z})} \cdot \frac{q(\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta})}\right) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} q(\boldsymbol{z}) \log\left(\frac{p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\Theta})}{q(\boldsymbol{z})}\right) d\boldsymbol{z} + \int_{\boldsymbol{z}} q(\boldsymbol{z}) \log\left(\frac{q(\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta})}\right) d\boldsymbol{z} \\
&= \mathcal{L}(q, \boldsymbol{\Theta}) + KL(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}))
\end{aligned}$$

The first term $\mathcal{L}(q, \boldsymbol{\Theta})$ is the **evidence lower bound (ELBO)** of the log-likelihood function $l(\boldsymbol{\Theta})$; the second term $KL(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}))$ is the **Kullback-Leibler (KL) divergence** between the distribution of latent variables $q(\boldsymbol{z})$ and the posterior distribution of latent variables $p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta})$.

Obviously, $\log p(\boldsymbol{X}|\boldsymbol{\Theta}) = \mathcal{L}(q, \boldsymbol{\Theta}) + KL(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta})) \geq \mathcal{L}(q, \boldsymbol{\Theta})$, shown as Figure 2 (a).

KL Divergence

The Kullback-Leibler (KL) divergence measures the difference between two probability distributions $p(\boldsymbol{z})$ and $q(\boldsymbol{z})$:

$$KL(p||q) = \int p(\boldsymbol{z}) \log\left(\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})}\right) d\boldsymbol{z}$$

The KL divergence is always non-negative, and it is equal to 0 if and only if $p(\boldsymbol{z}) = q(\boldsymbol{z})$.

E-step: By setting $\gamma_{ik}^{(t)} = p(z_k = 1|\boldsymbol{x}_i)$ as the distribution of latent variables, i.e. $q(\boldsymbol{z}) = p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta})$, the KL divergence term is minimized to 0, making the ELBO equals to the log-likelihood function, shown as Figure 2 (b).

M-step: After obtaining the distribution of latent variables $q(\boldsymbol{z}) = p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)})$, we plug it in the ELBO

$\mathcal{L}(q, \boldsymbol{\Theta})$

$$
\begin{aligned}
\mathcal{L}(q, \boldsymbol{\Theta}) &= \int_{\boldsymbol{z}} q(\boldsymbol{z}) \log \left(\frac{p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\Theta})}{q(\boldsymbol{z})} \right) d\boldsymbol{z} = \int_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)}) \log \left(\frac{p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\Theta})}{p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)})} \right) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)}) \log p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\Theta}) d\boldsymbol{z} - \int_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)}) \log p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)}) d\boldsymbol{z} \\
&= \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)})} [\log p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\Theta})] - \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)})} [\log p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}^{(t)})] \\
&= Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)}) + \text{const.}
\end{aligned}
$$

In M-step, the maximzation of $Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(t)})$ is essentially lifting the ELBO of the log-likelihood function $l(\boldsymbol{\Theta})$, shown as Figure 2 (c).

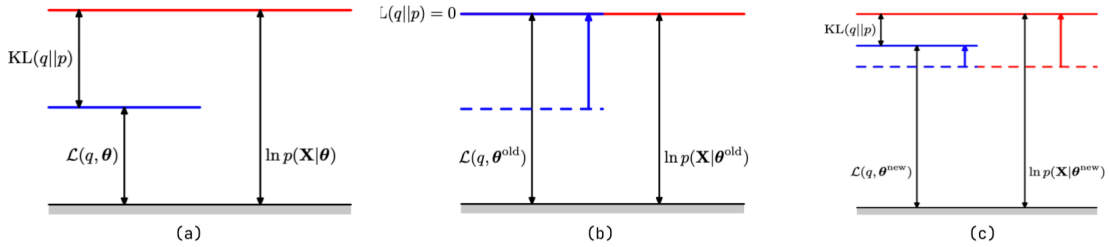

Figure 2: Mechanism of EM Algorithm

To summarize, E-step fixes the distribution of latent variables and makes ELBO equal to the log-likelihood; M-step lift the log-likelihood function by maximizing ELBO. The EM Algorithm iterates between E-step and M-step until convergence, shown as Figure 3.

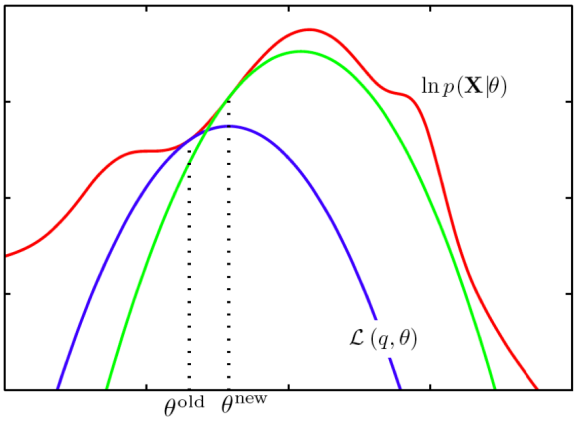

Figure 3: The alternative iterations between E-step and M-step

3 Revisiting K-means Clustering from the Viewpoint of EM Algorithm

K-means can be considered as a variant of GMM optimized with EM Algorithm. As is shown in Table 2, K-means clustering is a special case of GMM with the following properties *i*) Spherical clusters, *ii*) Hard assignment of data points to clusters.

	K-means Clustering	Gaussian Mixture Model
Shape of Cluster	Spherical	Elliptical
Cluster Assignment	Hard Assignment (Each data point is assigned to only one cluster)	Soft Assignment (Each data point is assigned to each cluster with a probability)

Table 2: K-means Clustering versus Gaussian Mixture Model

[Note]: The preassumed distribution of GMM can be set as other distributions, such as Bernoulli, Uniform, etc. The key point is to model a mixture of different distributions.

For convenience, here we still assume that the samples are generated by Gaussian distributions. In K-means Clustering, some parameters in GMM can be simplified:

- Mixing coefficient: $w_k = \dfrac{|\mathcal{C}_k|}{N}$
- Covariance matrix $\boldsymbol{\Sigma}_k = \epsilon \boldsymbol{I} \Rightarrow p(\boldsymbol{x}|\mathcal{C}_k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k = \epsilon \boldsymbol{I})$

So the only thing left for optimizing is the centroid $\boldsymbol{\mu}_k$ (The weighting coefficient w_k is always set as the proportion of data points in cluster k to all data points, and the covariance matrix $\boldsymbol{\sigma}_k$ is always a scalar matrix since the cluster blob is spherical). The E-step and M-step in K-means Clustering are conducted as follows:

- **E-step:** for each data point $\boldsymbol{x}_i$, when $\varepsilon \to 0$, the responsibility γ_{ik} is calculated by

$$\gamma_{ik}^{(t)} = \frac{w_k^{(t)} p(\boldsymbol{x}_i|\mathcal{C}_k)}{\sum_{j=1}^{K} w_j^{(t)} p(\boldsymbol{x}_i|\mathcal{C}_j)} = \frac{w_k^{(t)} \exp\{\dfrac{\|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2}{2\epsilon}\}}{\sum_{j=1}^{K} w_j^{(t)} \exp\{\dfrac{\|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|^2}{2\epsilon}\}} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\boldsymbol{x}_i - \boldsymbol{\mu}_j\|_2^2 \\ 0 & \text{otherwise} \end{cases}, \varepsilon \to 0$$

- **M-step:** for each cluster $\mathcal{C}_k$, the centroid $\boldsymbol{\mu}_k$ is updated by maximizing $Q(\boldsymbol{\mu}_k, \boldsymbol{\mu}_k^{(t)})$:

$$\begin{aligned} \boldsymbol{\mu}_k^{(t+1)} &= \min_{\boldsymbol{\mu}_k} Q(\boldsymbol{\mu}_k, \boldsymbol{\mu}_k^{(t)}) = \min_{\boldsymbol{\mu}_k} -\sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \log\left(\frac{1}{(2\pi)^{D/2}\epsilon^{D/2}} \exp\left(-\frac{1}{2\epsilon}\|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2\right)\right) \\ &= \min_{\boldsymbol{\mu}_k} \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_{ik}^{(t)} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2 \end{aligned}$$

This is exactly the optimization goal of K-means clustering itself. To summarize, K-means clustering is a special example of EM-optimized GMM where the distributions of samples under each cluster are Gaussian distributions with 0 covariance.

References

- Statistical Machine Learning 18: Gaussian Mixture Model and EM Algorithm
- EM Algorithm — From Latent Variable to Maximizing Evidence Lower Bound
- CSC 311: Introduction to Machine Learning — Lecture 11 k-Means and EM Algorithm