EM Algorithm Clear Explained

Hanyi Mao

March 2025

This vignette is inspired by an excellent textbook [1]. Section **2** and **3** are based on section **I.8.7** of [1], explained in a hopefully more understandable way.

1 Notation

Here are some symbols and conventions we will use in this vignette.

Notation	Description and Conventions
$\boldsymbol{\theta}$	The full parameter set of a given model (e.g., $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$).
D	Domain of definition.
t	Timestep.
n	Index of datapoints, $n = 1, 2, ..., N$.
$\boldsymbol{\theta}^t$	Estimation of $\boldsymbol{\theta}$ at timestep t.
$\boldsymbol{x}$	The full observed data, $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N)$; not using a matrix notation as different datapoints may have different length
$\boldsymbol{x}_n$	A data point, treated as column vectors.
$\boldsymbol{x}_n^{\mathrm{T}}$	The transpose of a vector or a matrix.
$\boldsymbol{z}$	Latent variable(s).
π_k, γ_{nk}	Some scalar parameters.
$\int_z p(z)dz$	Can also mean $\sum_z p(z)$ if z is a discrete variable.

Table 1: Notations and conventions used in this vignette.

2 Bound Optimization

The EM algorithm utilizes the idea of **bound optimization** (also known as **MM algorithms**), so we first look at this to have some intuitions in mind. Assume we want to *maximize* a function:

$$l(\boldsymbol{\theta}) : D \to \mathbb{R}$$

In many cases, we can't do this maximization directly because l is probably intractable (meaning it's hard to compute). Now suppose we get some function

$$g(\boldsymbol{\theta}; \boldsymbol{\theta}^t) : D \to \mathbb{R},\ t = 0, 1, 2, ...$$

in which the first $\boldsymbol{\theta} \in D$ is the independent variable and the second $\boldsymbol{\theta}^t \in D$ is the parameter of this function. (Notice that in many other literature, this function is denoted as $Q(\boldsymbol{\theta}^t; \boldsymbol{\theta}^t)$. We choose g rather than Q here in case people confuse this Q with the variational distribution q which we will see later.) It has following properties:

$$\forall \boldsymbol{\theta}\ g(\boldsymbol{\theta}; \boldsymbol{\theta}^t) \leq l(\boldsymbol{\theta}) \tag{1}$$

$$g(\boldsymbol{\theta}^t; \boldsymbol{\theta}^t) = l(\boldsymbol{\theta}^t) \tag{2}$$

which means $g(\boldsymbol{\theta}; \boldsymbol{\theta}^t)$ is a lower bound of $l(\boldsymbol{\theta})$ and it's tight at $\boldsymbol{\theta}^t$. Now we can do some interesting things with this.
To formalize, we start at a random $\boldsymbol{\theta}^0$. We update the parameter at each timestep t:

$$\boldsymbol{\theta}^{t+1} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, g(\boldsymbol{\theta}; \boldsymbol{\theta}^t) \tag{3}$$

We can prove that this new $\boldsymbol{\theta}^{t+1}$ **improves** the value of l, because

$$l(\boldsymbol{\theta}^{t+1}) \geq g(\boldsymbol{\theta}^{t+1}; \boldsymbol{\theta}^t) \geq g(\boldsymbol{\theta}^t; \boldsymbol{\theta}^t) = l(\boldsymbol{\theta}^t) \tag{4}$$

The first inequality follows Equation (1), the second inequality follows Equation (3), and the third inequality follows Equation (2). Then we can use $g(\boldsymbol{\theta}; \boldsymbol{\theta}^{t+1})$ to find the next $\boldsymbol{\theta}^{t+2}$. Repeating this until convergence, we can maximize $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. This process is visualized in figure 1. An animated version of visualization can be found in this project repository `https://github.com/echeloni11/EM_vignette/`, filename **BoundOptimization.mp4**.

3 The EM Algorithm

3.1 Derive a lower bound for log likelihood

In most cases of statistics/machine learning, our goal is to provide a maximum likelihook estimation of $\boldsymbol{\theta}$ given data $\boldsymbol{x}$.

$$\hat{\boldsymbol{\theta}}_{MLE} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\boldsymbol{x}; \boldsymbol{\theta}) \tag{5}$$

Sometimes we can't directly compute the likelihood as there are some unobserved variables in our modeling. The assumption here is that if we observed the latent variable $\boldsymbol{z}$, we can compute $p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta})$.

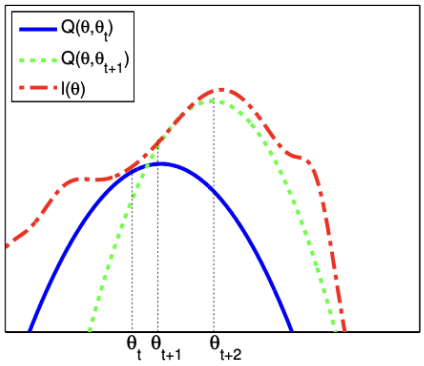

Figure 1: Visualization of Bound optimization. Adapted from [1]

With that in mind, let's see how to derive a tractable lower bound for log likelihood, so that we can apply the idea of bound optimization we saw (notice that maximizing likelihood is equal to maximizing log likelihood).
Given observed data $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_N\}$,

$$\log p(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n; \boldsymbol{\theta}) \tag{6}$$

$$= \sum_{n=1}^{N} \log \int_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n; \boldsymbol{\theta}) d\boldsymbol{z}_n \tag{7}$$

The log operation outside of integral is hard to track. We use a trick called **Jensen's Inequality Trick** (which is the core of **Variational Inference (VI)** methods). The trick works by first plugging in an arbitrary auxiliary distribution (or **variational distribution**) q_n for each n, then use Jensen's Inequality (that is, $f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$ for concave function f, and notice that log is concave)

to push the log operation into the integral.

$$\log p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n;\boldsymbol{\theta}) \tag{8}$$

$$= \sum_{n=1}^{N} \log \int_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n;\boldsymbol{\theta}) dz_n \tag{9}$$

$$= \sum_{n=1}^{N} \log \int_{\boldsymbol{z}_n} q_n(\boldsymbol{z}_n) \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n;\boldsymbol{\theta})}{q_n(\boldsymbol{z}_n)} d\boldsymbol{z}_n \tag{10}$$

$$= \sum_{n=1}^{N} \log \mathbb{E}_{q_n(\boldsymbol{z}_n)}[\frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n;\boldsymbol{\theta})}{q_n(\boldsymbol{z}_n)}] \tag{11}$$

$$\geq \sum_{n=1}^{N} \mathbb{E}_{q_n(\boldsymbol{z}_n)}[\log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n;\boldsymbol{\theta})}{q_n(\boldsymbol{z}_n)}] \tag{12}$$

$$=: g(\boldsymbol{\theta}; \{q_1, q_2, ..., q_N\}) \tag{13}$$

Now $g(\boldsymbol{\theta}; \{q_1, q_2, ..., q_N\})$ is a lower bound of log likelihood. This is also called the **Evidence Lower Bound** or **ELBO**, as the log marginal likelihood $p(\boldsymbol{x};\boldsymbol{\theta})$ is also known as the evidence. Notice that ELBO is a function with independent variable $\boldsymbol{\theta}$.

3.2 Find the q to make the bound tight

Recall that in bound optimization, we need the lower bound $g(\boldsymbol{\theta};\boldsymbol{\theta}^t)$ to take $\boldsymbol{\theta}^t$ as parameter and be tight at $\boldsymbol{\theta}^t$. So what q should we choose such that $g(\boldsymbol{\theta}; \{q_1, q_2, ..., q_N\})$ has such properties? With some algebraic manipulation, we can see that

$$g(\boldsymbol{\theta}; \{q_1, q_2, ..., q_N\}) \coloneqq \sum_{n=1}^{N} \mathbb{E}_{q_n(\boldsymbol{z}_n)}[\log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n;\boldsymbol{\theta})}{q_n(\boldsymbol{z}_n)}] \tag{14}$$

$$= \sum_{n=1}^{N} \mathbb{E}_{q_n(\boldsymbol{z}_n)}[\log \frac{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta})p(\boldsymbol{x}_n;\boldsymbol{\theta})}{q_n(\boldsymbol{z}_n)}] \tag{15}$$

$$= \sum_{n=1}^{N} \mathbb{E}_{q_n(\boldsymbol{z}_n)}[\log \frac{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta})}{q_n(\boldsymbol{z}_n)} + \log p(\boldsymbol{x}_n;\boldsymbol{\theta})] \tag{16}$$

$$= \sum_{n=1}^{N} \left[\mathbb{E}_{q_n(\boldsymbol{z}_n)}[\log \frac{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta})}{q_n(\boldsymbol{z}_n)}] + \log p(\boldsymbol{x}_n;\boldsymbol{\theta})\right] \tag{17}$$

$$= \sum_{n=1}^{N} [-D_{\mathbb{KL}}(q_n(\boldsymbol{z}_n))||p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}))] + \log p(\boldsymbol{x};\boldsymbol{\theta}) \tag{18}$$

where $D_{\mathbb{KL}}(p(\boldsymbol{z})||q(\boldsymbol{z})) \coloneqq \mathbb{E}_q[\log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z})}]$ is the Kullback-Leibler divergence which is a metric that measures the distance between two distributions. The key takeaway is that $D_{\mathbb{KL}}(q(\boldsymbol{z})||p(\boldsymbol{z})) \geq 0$, and $D_{\mathbb{KL}}(q(\boldsymbol{z})||p(\boldsymbol{z})) = 0$ if and only if $q = p$.
Thus, from Equation (18) we can see that $g(\boldsymbol{\theta}; \{q_1, q_2, ..., q_N\}) = \log p(\boldsymbol{x}; \boldsymbol{\theta})$ when $q_n(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta})$ for all n. This means that the **ELBO** is a tight lower bound when the variational distribution q is the posterior distribution of $\boldsymbol{z}$ given $\boldsymbol{x}$.
Now if we set $q_n(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)$, we have that

$$g(\boldsymbol{\theta}; \{p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)|n = 1, 2, ..., N\}) \leq \log p(\boldsymbol{x}; \boldsymbol{\theta}) \tag{19}$$

and

$$g(\boldsymbol{\theta}^t; \{p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)|n = 1, 2, ..., N\}) = \log p(\boldsymbol{x}; \boldsymbol{\theta}^t) \tag{20}$$

3.3 EM is Bound Optimization

We are almost there! Now let's put everything together and see how to use the idea of bound optimization to derive our EM algorithm. Given all above, it's natural to define that

$$l(\boldsymbol{\theta}) \coloneqq \log p(\boldsymbol{x}; \boldsymbol{\theta}) \tag{21}$$

$$g(\boldsymbol{\theta}; \boldsymbol{\theta}^t) \coloneqq g(\boldsymbol{\theta}; \{p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)|n = 1, 2, ..., N\}) \tag{22}$$

$$= \sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)}[\log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n; \boldsymbol{\theta})}{p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)}] \tag{23}$$

Following the idea of bound optimization, all we need to do is to randomly pick a starting $\boldsymbol{\theta}^0$ and use Equation (3) to recursively update $\boldsymbol{\theta}^t$! Now we can see the process of EM algorithm and why it is named this way.

3.3.1 E (Expectation) step

During this step, all we need is to compute $g(\boldsymbol{\theta}; \boldsymbol{\theta}^t)$. From above we have

$$g(\boldsymbol{\theta}; \boldsymbol{\theta}^t) = \sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)}[\log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n; \boldsymbol{\theta})}{p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)}] \tag{24}$$

$$= \sum_{n=1}^{N} \left[\mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n; \boldsymbol{\theta})] - \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)} p(\boldsymbol{z}_n|\boldsymbol{x}_n; \boldsymbol{\theta}^t)]\right] \tag{25}$$

from which we can see that the second term is a constant term with respect to $\boldsymbol{\theta}$. Thus, we throw the second term and only need

$$\sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n; \boldsymbol{\theta})] \tag{26}$$

which is the sum of N **expectations** of log joint probabilities (or **complete-data likelihood**) over the posterior distributions.

3.3.2 M (Maximization) step

During this step, we want to compute

$$\boldsymbol{\theta}^{t+1} = \operatorname*{argmax}_{\boldsymbol{\theta}} g(\boldsymbol{\theta}; \boldsymbol{\theta}^t) = \operatorname*{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n; \boldsymbol{\theta})] \tag{27}$$

This computation can be non-trivial. Luckily, for exponential family, we can solve this maximization by matching the moments of sufficient statistics. We will see more details in Example 4.3 and Appendix B.

4 Example in Biology: scRNA-Seq

4.1 Background Description and Problem Formalization

Single-cell RNA sequencing (scRNA-seq) allows us to profile gene expression at the resolution of individual cells. In our modeling framework, we consider a dataset $\boldsymbol{x} \in \mathbb{R}^{N \times d}$ where each row $\boldsymbol{x}_n \in \mathbb{R}^d$ corresponds to the gene expression profile of a single cell (with d genes measured). Although the true distribution of gene expression is complex, we assume for simplicity that each cell is drawn from a multivariate Gaussian distribution. This assumption, while a simplification, provides a tractable setting to illustrate the EM algorithm.

4.2 Clustering

In this subsection, we consider the problem of clustering cells into distinct groups. We assume a simplified situation in which the data are generated from a mixture of K multivariate Gaussian distributions with covariance matrices equal to the identity, i.e.,

$$\boldsymbol{\Sigma}_k = I \quad \text{for all } k = 1, \ldots, K.$$

Let $z_n \in \{1, 2, \ldots, K\}$ denote the latent variable indicating the cluster assignment for cell n, and let π_k be the prior probability of $z_n = k$. Under this model,

the complete-data log likelihood for cell n is given by

$$\log p(\boldsymbol{x}_n, z_n = k; \boldsymbol{\theta}) = \log \pi_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, I) \tag{28}$$

$$= \log \pi_k - \frac{1}{2}\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 + \text{const}, \tag{29}$$

where $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k \mid k = 1, 2, \ldots, K\}$.

EM Algorithm Derivation:

1. **E-step:** We first compute the posterior probabilities (also called *responsibilities*), which indicate how much the cell $\boldsymbol{x}_n$ is "responsible" for cluster k:

$$\gamma_{nk}^t \coloneqq p(z_n = k \mid \boldsymbol{x}_n; \boldsymbol{\theta}^t) = \frac{\pi_k^t \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k^t, I)}{\sum_{j=1}^K \pi_j^t \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_j^t, I)}.$$

   Then, we compute the expectation of the complete-data log likelihood (ignoring constant terms):

$$g(\boldsymbol{\theta}; \boldsymbol{\theta}^t) = \sum_{n=1}^N \mathbb{E}_{p(z_n \mid \boldsymbol{x}_n; \boldsymbol{\theta}^t)} \left[\log p(\boldsymbol{x}_n, z_n; \boldsymbol{\theta})\right]$$

$$= \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk}^t \left[\log \pi_k - \frac{1}{2}\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2\right]. \tag{30}$$

2. **M-step:** Differentiating $g(\boldsymbol{\theta}; \boldsymbol{\theta}^t)$ with respect to $\boldsymbol{\mu}_k$ and setting the derivative to zero yields

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{n=1}^N \gamma_{nk}^t \boldsymbol{x}_n}{\sum_{n=1}^N \gamma_{nk}^t},$$

   and similarly, the update for the mixing coefficients is

$$\pi_k^{(t+1)} = \frac{1}{N} \sum_{n=1}^N \gamma_{nk}^t.$$

At the end of the EM algorithm, we obtain estimates $\{\hat{\boldsymbol{\mu}}_k, \hat{\pi}_k\}$ along with the posterior distribution indicating the probability of each data point belonging to each cluster:

$$p(z_n = k \mid \boldsymbol{x}_n; \boldsymbol{\theta}) = \gamma_{nk} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k, I)}{\sum_{j=1}^K \pi_j \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_j, I)}. \tag{31}$$

Assigning a Definite Class: What if we want to assign an exact class to each cell? We can do so by taking the class with the highest posterior probability.

In other words, we assign

$$\hat{z}_n = \arg\max_k p(z_n = k \mid \boldsymbol{x}_n; \boldsymbol{\theta}) = \arg\max_k \gamma_{nk}.$$

In fact, we can incorporate this hard assignment directly into the EM algorithm. That is, instead of computing the expectation in each E-step, we approximate it using the mode. This method is known as **Hard EM**. In our example, the hard EM process is to set

$$z_n^t = \arg\max_k \gamma_{nk}^t,$$

so that the expected complete-data log likelihood is approximated by

$$g(\boldsymbol{\theta}; \boldsymbol{\theta}^t) \approx \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{I}_{\{z_n^t = k\}} \left[\log \pi_k - \frac{1}{2} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \right], \tag{32}$$

where $\mathbb{I}_{\{z_n^t = k\}}$ is the indicator function that equals 1 if $z_n^t = k$ and 0 otherwise. In fact, if we assume further that the mixing coefficients are uniform, i.e., $\pi_k = \frac{1}{K}$ for all k, what we get is surprisingly the famous **K-Means** algorithm. The proof of this is left as practice; you can find the answer at Appendix A.

4.3 Missing Data

Now consider a different scenario where each cell is assumed to be generated from the same multivariate Gaussian distribution with parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \Sigma)$, but only a subset of the entries in $\boldsymbol{x}_n$ is observed. In other words, some entries in the data matrix $\boldsymbol{x}$ are missing at random (MAR). We denote by $\boldsymbol{x}_n^{\text{obs}}$ and $\boldsymbol{x}_n^{\text{mis}}$ the observed and missing parts of $\boldsymbol{x}_n$, respectively. The complete-data likelihood is given by

$$p(\boldsymbol{x}_n^{\text{obs}}, \boldsymbol{x}_n^{\text{mis}}; \boldsymbol{\theta}) = \mathcal{N}\left(\begin{pmatrix} \boldsymbol{x}_n^{\text{obs}} \\ \boldsymbol{x}_n^{\text{mis}} \end{pmatrix} \,\middle|\, \begin{pmatrix} \boldsymbol{\mu}^{\text{obs}} \\ \boldsymbol{\mu}^{\text{mis}} \end{pmatrix}, \Sigma \right).$$

As before, our goal is to maximize the expected complete-data log-likelihood. We define

$$g(\boldsymbol{\theta}; \boldsymbol{\theta}^t) = \sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{x}_n^{\text{mis}} \mid \boldsymbol{x}_n^{\text{obs}}; \boldsymbol{\theta}^t)} \left[\log p(\boldsymbol{x}_n^{\text{obs}}, \boldsymbol{x}_n^{\text{mis}}; \boldsymbol{\theta}) \right]. \tag{33}$$

This computation can be non-trivial. However, if the complete-data probability is in the exponential family (e.g., Gaussian and many other common distributions), then we can exploit its structure to simplify the M-step. Unfortunately, the computation is still not easy. The basic idea is to use the result of conditional probability of multivariate Gaussians to impute the missing data $\boldsymbol{x}_n^{\text{mis}}$ as well as the expectation of second moment $\left(\boldsymbol{x}_n^{\text{mis}}, \boldsymbol{x}_n^{\text{obs}}\right)\left(\boldsymbol{x}_n^{\text{mis}}, \boldsymbol{x}_n^{\text{obs}}\right)^{\mathrm{T}}$ at each

step. A detailed introduction to exponential family and how do we utilize its property for EM can be found at Appendix B.

References

[1] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL: `http://probml.github.io/book1`.

A Proof that Hard EM Reduces to K-Means

Now the mixing coefficients are uniform, i.e., $\pi_k = \frac{1}{K}$ for all k. Under this condition, $\log \pi_k$ is constant across all clusters and can be dropped from the maximization. The denominator can also be dropped as it is shared across different k. Then, the responsibility in the E-step becomes

$$z_n^t = \arg\max_k \gamma_{nk}^t = \arg\max_k \mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k^t, I).$$

Recalling that the Gaussian density with identity covariance is given by

$$\mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k^t, I) \propto \exp\left(-\frac{1}{2}\|\boldsymbol{x}_n - \boldsymbol{\mu}_k^t\|^2\right),$$

maximizing $\mathcal{N}(\boldsymbol{x}_n \mid \boldsymbol{\mu}_k^t, I)$ is equivalent to minimizing the squared Euclidean distance:

$$z_n^t = \arg\min_k \|\boldsymbol{x}_n - \boldsymbol{\mu}_k^t\|^2.$$

This is exactly the assignment rule in K-Means.

In the M-step, with hard assignments, the update for $\boldsymbol{\mu}_k$ becomes

$$\boldsymbol{\mu}_k^{(t+1)} = \frac{\sum_{n=1}^N \mathbb{I}_{\{z_n^t = k\}}\, \boldsymbol{x}_n}{\sum_{n=1}^N \mathbb{I}_{\{z_n^t = k\}}}.$$

This is the standard update in K-Means, where each cluster mean is recalculated as the average of all points assigned to that cluster.

Thus, under the assumptions of identity covariance and uniform mixing coefficients, the Hard EM procedure for mixture of Gaussians is equivalent to the standard K-Means algorithm:

- **E-step:** Assign each data point to the cluster with the nearest mean (minimum Euclidean distance).
- **M-step:** Update each cluster mean as the average of the data points assigned to that cluster.

B Matching the sufficient statistics for Exponential Family

In general, a distribution in the exponential family can be written as

$$p(\boldsymbol{x};\boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\Big(\boldsymbol{\theta}^{\mathrm{T}}\mathcal{T}(\boldsymbol{x}) - A(\boldsymbol{\theta})\Big), \tag{34}$$

where:

- $\boldsymbol{\theta}$ is the *natural parameter*,
- $\mathcal{T}(\boldsymbol{x})$ is the *sufficient statistic*,
- $A(\boldsymbol{\theta})$ is the *log-partition function* (or cumulant generating function),
- $h(\boldsymbol{x})$ is the base measure.

A key property of the log-partition function is that its first derivative with respect to $\boldsymbol{\theta}$ equals the expectation of the sufficient statistic:

$$\frac{\partial A(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathbb{E}_{p(\boldsymbol{x};\boldsymbol{\theta})}\big[\mathcal{T}(\boldsymbol{x})\big].$$

Exploiting this property, we can rewrite the expected complete-data log-likelihood as

$$g(\boldsymbol{\theta};\boldsymbol{\theta}^t) = \sum_{n=1}^{N} \Big(\mathbb{E}_{q_n(\boldsymbol{z}_n;\boldsymbol{\theta}^t)}\big[\mathcal{T}(\boldsymbol{x}_n,\boldsymbol{z}_n)\big]^{\mathrm{T}}\boldsymbol{\theta} - A(\boldsymbol{\theta})\Big), \tag{35}$$

where $q_n(\boldsymbol{z}_n;\boldsymbol{\theta}^t) = p(\boldsymbol{z}_n \mid \boldsymbol{x}_n^{\mathrm{obs}};\boldsymbol{\theta}^t)$ is the posterior of the latent variables given the observed data under the current parameter estimates.

Taking the derivative of $g(\boldsymbol{\theta};\boldsymbol{\theta}^t)$ with respect to $\boldsymbol{\theta}$ and setting it to zero yields

$$\frac{\partial g(\boldsymbol{\theta};\boldsymbol{\theta}^t)}{\partial \boldsymbol{\theta}} = \sum_{n=1}^{N} \left(\mathbb{E}_{q_n(\boldsymbol{z}_n;\boldsymbol{\theta}^t)}\big[\mathcal{T}(\boldsymbol{x}_n,\boldsymbol{z}_n)\big] - \frac{\partial A(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right) = 0.$$

Using the property of $A(\boldsymbol{\theta})$, this condition is equivalent to

$$\frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{q_n(\boldsymbol{z}_n;\boldsymbol{\theta}^t)}\big[\mathcal{T}(\boldsymbol{x}_n,\boldsymbol{z}_n)\big] = \mathbb{E}_{p(\boldsymbol{x};\boldsymbol{\theta})}\big[\mathcal{T}(\boldsymbol{x})\big]. \tag{36}$$

For a multivariate Gaussian distribution, the sufficient statistics are

$$\mathcal{T}(\boldsymbol{x}) = (\boldsymbol{x},\ \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}})$$

Plugging this to the right side Equation (36), we have that

$$\mathbb{E}_{p(\boldsymbol{x};\boldsymbol{\theta})}\big[\mathcal{T}(\boldsymbol{x})\big] = \mathbb{E}_{p(\boldsymbol{x};\boldsymbol{\theta})}[(\boldsymbol{x},\ \boldsymbol{x}\boldsymbol{x}^{\mathrm{T}})] = (\boldsymbol{\mu}, \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}}) \tag{37}$$

Now we need to compute the left side of Equation (36) to update $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We need to use the result of the conditional probability for a multivariate Gaussian. Assume

$$p\Big(\boldsymbol{x}_n^{\mathrm{mis}}, \boldsymbol{x}_n^{\mathrm{obs}} \mid \boldsymbol{\theta}^t\Big) = \mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_t^{\mathrm{mis}}\\ \boldsymbol{\mu}_t^{\mathrm{obs}}\end{pmatrix}, \begin{pmatrix}\boldsymbol{\Sigma}_t^{mm} & \boldsymbol{\Sigma}_t^{mo}\\ \boldsymbol{\Sigma}_t^{om} & \boldsymbol{\Sigma}_t^{oo}\end{pmatrix}\right). \tag{38}$$

Then, by standard results on the conditional distribution of a multivariate Gaussian, we have

$$p\Big(\boldsymbol{x}_n^{\mathrm{mis}} \mid \boldsymbol{x}_n^{\mathrm{obs}}, \boldsymbol{\theta}^t\Big) = \mathcal{N}\big(\boldsymbol{m}_n^t, \boldsymbol{V}_n^t\big) \tag{39}$$

$$\boldsymbol{m}_n^t \coloneqq \boldsymbol{\mu}_t^{\mathrm{mis}} + \boldsymbol{\Sigma}_t^{mo}\left(\boldsymbol{\Sigma}_t^{oo}\right)^{-1}\left(\boldsymbol{x}_n^{\mathrm{obs}} - \boldsymbol{\mu}_t^{\mathrm{obs}}\right) \tag{40}$$

$$\boldsymbol{V}_n^t \coloneqq \boldsymbol{\Sigma}_t^{mm} - \boldsymbol{\Sigma}_t^{mo}\left(\boldsymbol{\Sigma}_t^{oo}\right)^{-1}\boldsymbol{\Sigma}_t^{om} \tag{41}$$

Now we can compute the left side of (36)

$$\sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{x}_n^{\mathrm{mis}}|\boldsymbol{x}_n^{\mathrm{obs}};\boldsymbol{\theta}^t)}\Big[\mathcal{T}(\boldsymbol{x}_n^{\mathrm{mis}}, \boldsymbol{x}_n^{\mathrm{obs}})\Big] \tag{42}$$

$$= \sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{x}_n^{\mathrm{mis}}|\boldsymbol{x}_n^{\mathrm{obs}};\boldsymbol{\theta}^t)}\left[\Big(\big(\boldsymbol{x}_n^{\mathrm{mis}}, \boldsymbol{x}_n^{\mathrm{obs}}\big),\ \big(\boldsymbol{x}_n^{\mathrm{mis}}, \boldsymbol{x}_n^{\mathrm{obs}}\big)\big(\boldsymbol{x}_n^{\mathrm{mis}}, \boldsymbol{x}_n^{\mathrm{obs}}\big)^{\mathrm{T}}\Big)\right] \tag{43}$$

$$= \sum_{n=1}^{N}\left(\big(\boldsymbol{m}_n^t, \boldsymbol{x}_n^{\mathrm{obs}}\big),\ \begin{pmatrix}\boldsymbol{V}_n^t + \boldsymbol{m}_n^t(\boldsymbol{m}_n^t)^{\mathrm{T}} & \boldsymbol{m}_n^t(\boldsymbol{x}_n^{\mathrm{obs}})^{\mathrm{T}}\\ \boldsymbol{x}_n^{\mathrm{obs}}(\boldsymbol{m}_n^t)^{\mathrm{T}} & \boldsymbol{x}_n^{\mathrm{obs}}(\boldsymbol{x}_n^{\mathrm{obs}})^{\mathrm{T}}\end{pmatrix}\right) \tag{44}$$

Thus, to update $\boldsymbol{\theta}^{t+1} = (\boldsymbol{\mu}^{t+1}, \boldsymbol{\Sigma}^{t+1})$, we have

$$\boldsymbol{\mu}^{t+1} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{Reordered}(\boldsymbol{m}_n^t, \boldsymbol{x}_n^{\mathrm{obs}}), \tag{45}$$

$$\boldsymbol{\Sigma}^{t+1} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{Reordered}\begin{pmatrix}\boldsymbol{V}_n^t + \boldsymbol{m}_n^t(\boldsymbol{m}_n^t)^{\mathrm{T}} & \boldsymbol{m}_n^t(\boldsymbol{x}_n^{\mathrm{obs}})^{\mathrm{T}}\\ \boldsymbol{x}_n^{\mathrm{obs}}(\boldsymbol{m}_n^t)^{\mathrm{T}} & \boldsymbol{x}_n^{\mathrm{obs}}(\boldsymbol{x}_n^{\mathrm{obs}})^{\mathrm{T}}\end{pmatrix} - \boldsymbol{\mu}^{t+1}(\boldsymbol{\mu}^{t+1})^{\mathrm{T}}. \tag{46}$$

The Reordered operation here is to make sure that variables are in the same order because different n may have different set of missing variables.

For more details on the exponential family, see `https://en.wikipedia.org/wiki/Exponential_family` or Section I.3.4 of [1].