Notes on EM and Gaussian Mixtures

Alexander Hoyle

March 2026

1 Overview of the Gaussian distribution and multivariate random variables

Gaussian distribution Let's consider the most important distribution in statistics: the Gaussian.

$$p(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

This is a bit hairy, so we will break it down μ here is the mean of the distribution and σ^2 the variance. If we focus just on that inner part

$$-(x-\mu)^2$$

we can see that we move further from the mean, we have exponentially lower density (many things follow this pattern: e.g., the volume of a noise source decreases exponentially the further you are from it). The variance σ^2 scales this relationship: the lower the variance, the more confidence we have that values should be near the mean. As we move away from it, we rapidly get lower probability density. On the other hand, with large variance, the "cost" of going away from the mean is smaller.

Multivariate Gaussian So far we've dealt with scalar random variables. But in practice, our data points often live in d-dimensional space: a point $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]^\top \in \mathbb{R}^d$.[1] For instance, you might describe a person by their [height, weight]; an entire population of deer by their ages [age of deer 1, age of deer 2, ..., age of deer n]; or the volume of rain that will fall in a grid of square kilometers across Switzerland.

We might want to model the joint distribution over all d dimensions simultaneously, and—to anticipate things a bit—we might expect that these individual points might have systematic relationships with each other.

The multivariate Gaussian generalizes the univariate case to d dimensions. The mean $\boldsymbol{\mu} \in \mathbb{R}^d$ is now a vector, and the variance becomes a $d \times d$ *covariance matrix* $\boldsymbol{\Sigma}$ that captures both the spread along each dimension and the correlations between dimensions:

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

The quantity $(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$ is the *Mahalanobis distance*—a generalization of the squared distance from the mean that accounts for the shape of the distribution. When $\boldsymbol{\Sigma}$ is a full matrix, the resulting density contours are ellipses that can be oriented in any direction (capturing correlations between dimensions). In this lecture, we will work with a simplified *diagonal* covariance $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_d^2)$, which assumes the dimensions are independent. In this case the density reduces to

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\sigma}^2) = \frac{1}{(2\pi)^{d/2}\prod_{j=1}^d \sigma_j} \exp\left(-\frac{1}{2}\sum_{j=1}^d \frac{(x_j-\mu_j)^2}{\sigma_j^2}\right)$$

We can simplify further by assuming the same variance in every dimension, giving $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$. This is the form we will use for the Gaussian Mixture Model, and it means the density contours are spheres (circles in 2D). All the arguments generalize to the full covariance case.

[1] Text in blue is AI-generated.

2 Review: Maximum Likelihood

Say we observe some data $x_i \sim p(x_i \mid \theta)$ for $i = 1, \ldots, n$ for a known parametric distribution p with parameters θ. Maximum likelihood estimation is a procedure to learn the parameters that make the data most probable under this distribution. For instance, say our data is sampled IID from a univariate Gaussian:

$$x_i \sim \mathcal{N}(\mu, \sigma^2)$$

Generally, because each sampling event is independent, we know that we write an IID likelihood as

$$\mathcal{L}(\theta, \boldsymbol{x}) = \prod_{i=1}^{n} p(x_i \mid \theta)$$

So in this case, we have

$$\begin{aligned}
\mathcal{L}(\mu, \sigma^2, \boldsymbol{x}) &= \prod_{i=1}^{n} \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{1}{2\sigma^2}(x_i - \mu)^2\right) \\
&= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2\right) \\
\ell(\mu, \sigma^2, \boldsymbol{x}) &= -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2
\end{aligned}$$

Now, to find a maximum, we differentiate with respect to our parameters and set this equal to zero. We'll just look at μ for now:

$$\begin{aligned}
\frac{\partial \ell}{\partial \mu} &= \frac{1}{\sigma^2}\sum_{i=1}^{n}(x_i - \mu) \\
&= -\frac{n\mu}{\sigma^2} + \frac{1}{\sigma^2}\sum_{i=1}^{n} x_i \\
\implies \hat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} x_i
\end{aligned}$$

3 Motivating the Gaussian Mixture Model

Before introducing the probabilistic model, let's recall the k-means algorithm, which solves a related but simpler problem. Our data points now live in $\mathbb{R}^d$: each x_i is a vector with d components. The (squared) Euclidean distance between two points $\boldsymbol{a}$ and $\boldsymbol{b}$ in this space is

$$||\boldsymbol{a} - \boldsymbol{b}||^2 = \sum_{j=1}^{d}(a_j - b_j)^2$$

This is just the Pythagorean theorem generalized to d dimensions.
We have a number of unlabeled points x_i and a suspicion that they fall into k clusters C_j. Our objective is to minimize the distance between each point and the mean of the cluster to which it belongs.

But we can't do this all at once: we need an iterative algorithm. After randomly initializing the k cluster centers u_j, $j = 1 \ldots k$, the algorithm has two steps that we repeat until convergence:

(1) For each of the n points x_i, we loop through the k centers of our clusters u_j, and calculate the distance

$$d_{ij} = ||x_i - u_j||^2.$$

We deterministically assign the point to the cluster C_j for which this distance is smallest. After this step, each cluster is going to have a new set of points.

(2) We update the centers u_j by calculating the mean of all the points in each cluster:

$$u_j = \frac{1}{|C_j|}\sum_{x_i \in C_j} x_i$$

3.1 Gaussian Mixture Model

In a Gaussian Mixture Model, we have a similar setup, wherein we have a set of points that we believe roughly fall into k clusters (or more precisely, that are generated by k component distributions).[2] The difference is that we now have an explicit *probabilistic model* of our data, a "story" of how our observations came to be. Namely, we believe that each of the n points was generated from one of k component Gaussian distributions:

$$\begin{aligned} \boldsymbol{\phi} = [\phi_1, \ldots, \phi_k],\ ||\boldsymbol{\phi}|| = 1; \quad & \boldsymbol{\mu} = [\mu_1, \ldots, \mu_k]; \quad \boldsymbol{\sigma^2} = [\sigma_1^2, \ldots, \sigma_k^2] \\ & z_i \sim \text{Categorical}(z_i \mid \boldsymbol{\phi}) \\ & x_i \sim \mathcal{N}\left(x_i \mid \mu_{z_i}, \sigma_{z_i}^2 \mathbf{I}\right) \end{aligned}$$

In words, it's fairly straightforward. We have k multivariate Gaussian distributions each with (unknown) mean and covariance[3] μ_j and $\sigma^2\mathbf{I}$, respectively: these are our "clusters". Some clusters are larger than others, and ϕ_j corresponds to the proportion of points generated by component j (so $\sum_j^k \phi_j = 1$).

When generating our data, first we decide which component distribution each point x_i is going to belong to: this is z_i, a positive integer between 1 and k. z_i indexes (or "selects") the parameters of the particular Gaussian from which we will sample x_i. However, z_i are unobserved in reality; they're latent variables.

The difference from k-means is that we can now can label each point with a *probability* that it came from a particular component. Cluster assignments are no longer "hard".

4 EM for Gaussian Mixtures

4.1 An attempt at Maximum Likelihood Estimation

To reiterate, we have a probabilistic model of unknown parameters. We can do the obvious thing, and try maximum likelihood estimation of the *observed* data $\boldsymbol{x}$, $\arg\max_\theta p(\boldsymbol{x} \mid \theta)$. But we could imagine that, if we knew the latent variables, trying to maximize the joint likelihood, $\arg\max_\theta p(\boldsymbol{x}, \boldsymbol{z} \mid \theta)$.

Let's start with the obvious approach, while keeping this other idea in the back of our minds:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) &= \prod_{i=1}^n p(x_i \mid \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \\ \ell(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) &= \sum_{i=1}^n \log p(x_i \mid \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \\ &= \sum_{i=1}^n \log \sum_{z_i \in \{1 \ldots k\}} p(x_i, z_i \mid \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \qquad \left[\text{since } P(A) = \sum_B P(A, B)\right] \quad (1) \\ &= \sum_{i=1}^n \log \sum_{z_i \in \{1 \ldots k\}} \mathcal{N}(x_i \mid \mu_{z_i}, \sigma_{z_i}^2 \mathbf{I}) \text{Cat}(z_i \mid \boldsymbol{\phi}) \qquad \left[\text{since } P(A, B) = P(A \mid B) P(B)\right] \\ &= \sum_{i=1}^n \log \sum_{j=1}^k \mathcal{N}(x_i \mid \mu_j, \sigma_j^2 \mathbf{I}) \phi_j \qquad \left[\text{Cat}(A = j) = P(\mathbb{I}(A = j)) = \phi_j\right] \quad (2) \end{aligned}$$

We can now differentiate the above with respect to the various parameters. Note that the sum inside the log makes things a little tough (what if z were continuous?), but for the current problem it's manageable. Let's start with μ_j.

[2] I use the word "clusters" to relate it to k-means, but a cluster implies that data points are distinguishable, which may not be the case.

[3] For simplicity, we will be working with diagonal covariance, but all arguments generalize to full covariance matrices

$$\frac{\partial \ell}{\partial \mu_j} = \sum_{i=1}^{n} \frac{1}{\sum_{j'=1}^{k} \mathcal{N}(x_i \mid \mu_{j'}, \sigma_{j'}^2 \mathbf{I}) \phi_{j'}} \frac{\partial}{\partial \mu_j} \mathcal{N}(x_i \mid \mu_j, \sigma_j^2 \mathbf{I}) \phi_j \quad \left[\frac{d}{dx} \log(af(x) + bg(y)) = \frac{f'(x)a}{af(x) + bg(y)}\right]$$

$$= \sum_{i=1}^{n} \frac{\mathcal{N}(x_i \mid \mu_j, \sigma_j^2 \mathbf{I}) \phi_j}{\sum_{j'=1}^{k} \mathcal{N}(x_i \mid \mu_{j'}, \sigma_{j'}^2 \mathbf{I}) \phi_{j'}} \frac{2}{2\sigma_j^2}(x_i - \mu_j) \qquad \left[\frac{d}{dx} a e^{f(x)} = a e^{f(x)} f'(x)\right]$$

Take a moment to interpret the left-hand fraction here. The numerator is the probability of a specific point coming from a specific Gaussian, $\mathcal{N}(x_i \mid \mu_j, \sigma_j)$, weighted by the proportion of points in that "cluster", ϕ_j, finally normalized by the probabilities of all the other points. In other words, it's the posterior probability that a point x_i came from component j (called the *responsibility* of component j for point x_i).

Call this probability r_{ij}, equal to $P(z_i = j \mid x_i, \mu_j, \sigma_j, \phi_j)$, and let's forget for a moment that it's a function of our unknown parameters. Setting the above to zero, we have

$$0 = \sum_{i=1}^{n} r_{ij} \frac{1}{\sigma_j^2}(x_i - \mu_j)$$

$$0 = \sum_{i=1}^{n} r_{ij} \frac{1}{\sigma_j^2} x_i - \sum_{i=1}^{n} r_{ij} \frac{1}{\sigma_j^2} \mu_j$$

$$\hat{\mu}_j = \frac{\sum_{i=1}^{n} r_{ij} x_i}{\sum_{i=1}^{n} r_{ij}}$$

You'll note a rough correspondence of this estimate to the MLE for n points sampled from a Gaussian, and to the second step of the k-means algorithm (to see this, imagine that $r_{ij} = 1$ or 0, that is, we know for certain which component each point comes from). The sum over the r_{ij} is the "effective number of points assigned to component j"[3]. In a sense, we have a "soft", version of k-means — we might call them *weighted* means — where each point is weighted by its probability of being in a given cluster.

The other parameters follow more or less the same approach, and lead to similar interpretations. Stating them here without proof[4]:

$$\hat{\phi}_j = \frac{1}{n} \sum_{i=1}^{n} r_{ij}$$

$$\hat{\sigma}_j^2 = \frac{\sum_{i=1}^{n} r_{ij} ||x_i - \mu_j||^2}{\sum_{i=1}^{n} r_{ij}}$$

Of course, there's an issue with our above estimates. They are all functions of themselves and the other parameters (through r_{ij}), meaning that there's no closed-form solution.

4.2 One view of the EM algorithm

4.2.1 Aside: A note on probability

Although we aren't dealing with the fully Bayesian form of Gaussian Mixture Models here, it's still helpful to reflect for a moment on what a probability distribution can be thought to represent. In some sense, they can encode our beliefs about the world — if we think of a Gaussian, a small variance corresponds to a stronger belief.

The benefit of probabilistic models is that, in a very rough sense, they allow us to manipulate *beliefs* about numbers (almost) as if they were numbers themselves. We also can update these beliefs based on the data we observe, thus reducing our uncertainty.[5]

[4] To show $\hat{\phi}$ we need to make use of a Langrange multiplier given the constraint $||\boldsymbol{\phi}|| = 1$

[5] For an example of this principle applied to the problem of finding intelligent extraterrestrial life, see [5])

4.2.2 Overview

What would make our problem easier? Well, if we knew the latent variables z_i, we'd be able to say whether point x_i was generated by Gaussian j, and $r_{ij} = P(z_i = j \mid x_i)$ would either be 1 or 0. We could then easily estimate the parameters of the Gaussians and the proportion of points generated by each — basically, it would amount to a counting problem.

Since we don't know the z_i values, we'll have to make an informed guess. As discussed, we have an belief about what these are, through the posterior $r_{ij} = P(z_i = j \mid x_i, \mu_j, \sigma_j^2, \phi_j)$. A posterior, roughly speaking, can be thought to represent the information we have about an unknown quantity given observed data.

But now we run into the problem from before, namely, that the posterior is a function of unknown parameters (μ_j, σ_j^2), which of course would be easy to compute if we knew the posteriors... [6]

EM presents an intuitive solution to this conundrum:

E-step We'll proceed as if we knew the parameters (starting with some initial guess), fixing them in place, and use them to calculate the posteriors.

M-step We'll use these posteriors to get better estimates of the parameters.

These steps are analogous to the two steps in k-means, the only difference is that we'll be working with "beliefs" about the cluster assignments, represented by probability distributions, rather than fixed numeric values[7].

4.2.3 EM, concretely

Up until now, we've focused on maximizing the log-likelihood of observed data $\log p(x \mid \theta)$. However, as I mentioned at the outset in section 4.1, one option is to act as if the z are observed, and attempt to optimize the "complete" log-likelihood of the data $\log p(x, z)$. As discussed, the posterior $p(z \mid x, \theta)$ conveys what we know about z. The reason we want to shift focus to the complete log-likelihood is because we marginalize over z in eq. 1—there's no way to include the information we know about it without it getting "lost". [8][9]

As discussed, the posterior $p(z \mid x, \theta)$ conveys what we know about z, so we'll use that information to give a best-guess of the complete log-likelihood. Since the complete log-likelihood is a function of a random variable z, we'll take its *expectation* under the posterior of z[10]

$$Q(\theta, \theta^l) = \mathbb{E}_{p(z|x,\theta^l)}[\log p(x, z \mid \theta)] = \sum_z [\log p(x, z \mid \theta)]\, p(z \mid x, \theta^l) \tag{3}$$

Where θ^l is our current best guess for the parameters, and θ is a free variable.

Thus, we have our algorithm:

E-step Use the current best guess of parameters θ^l to evaluate the posterior $p(z \mid x, \theta^l)$

M-step maximize eq. 3 with respect to θ:

$$\theta^{l+1} = \arg\max_\theta Q(\theta, \theta^l)$$

Applying this to GMMs, we'll first write out the joint probability:

[6] Many of the arguments / structure from this section come from [3]

[7] The estimates of the parameters are in fact fixed points, but in the fully Bayesian GMM they too are represented by probability distributions

[8] This is more clear when looking at the form of the objective in eq. 3: if you replace the $\log p(x, z \mid \theta)$ with $\log \sum_{z'} p(x, z' \mid \theta)$, it can be pulled out and the $\sum_z p(z \mid x, \theta)$ will just sum to 1.

[9] It's also more computationally tractable in the case where z is continuous or can take on many discrete values.

[10] Recall that taking the expectation of function of a random variable involves evaluating that function at all possible values of that variable, weighing the outputs by how probable that value is, then taking the sum (thus forming a weighted mean).

$$p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\phi}) = \prod_{i=1}^{n} \prod_{j=1}^{k} p(x_i, z_i = j \mid \mu_j, \sigma_j^2, \phi_j)^{\mathbb{I}(z_i=j)} \qquad \left[\text{where } \mathbb{I}(z_i = j) = 1 \text{ when } z_i = j, \text{ 0 otherwise}\right]$$
$$= \prod_{i=1}^{n} \prod_{j=1}^{k} \left(p(x_i \mid \mu_j, \sigma_j^2) p(z_i = j \mid \phi_j)\right)^{\mathbb{I}(z_i=j)}$$

To see why this is the case, we know that $\mathbb{I}(z_i = j) = 1$ only when x_i came from component j. Now, we can write down the complete likelihood then take the log:

$$\mathcal{L}_c = \prod_{i=1}^{n} \prod_{j=1}^{k} \left(\mathcal{N}(x_i \mid \mu_j, \sigma_j^2 \mathbf{I}) \phi_j\right)^{\mathbb{I}(z_i=j)}$$
$$\ell_c = \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbb{I}(z_i = j) \left(\log \phi_j \mathcal{N}(x_i \mid \mu_j, \sigma_j^2 \mathbf{I})\right)$$

Then, taking the expectation under the posterior (which we'll just write as $p(z \mid x)$, ignoring the parameters for clarity):

$$\mathbb{E}_{p(z|x)}[\ell_c] = E_{p(z|x)} \left[\sum_{i=1}^{n} \sum_{j=1}^{k} \mathbb{I}(z_i = j) \left(\log \mathcal{N}(x_i \mid \mu_j, \sigma_j^2 \mathbf{I}) + \log \phi_j\right) \right]$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{k} E_{p(z|x)} \left[\mathbb{I}(z_i = j)\right] \left(\log \mathcal{N}(x_i \mid \mu_j, \sigma_j^2 \mathbf{I}) + \log \phi_j\right)$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} \left(\log \mathcal{N}(x_i \mid \mu_j, \sigma_j^2 \mathbf{I}) + \log \phi_j\right)$$

The last equality follows because the expectation of $\mathbb{I}(z_i = j)$ under the posterior $p(z_i \mid x_i, \mu_j^l, \sigma_j^{2l}, \phi_j^l)$ is just when $p(z_i = j \mid x_i, \mu_j^l, \sigma_j^{2l}, \phi_j^l) = r_{ij}$

Now we have our $Q(\theta, \theta^l)$, and can apply EM in the previous section as discussed. This amounts to evaluating r_{ij} for the set of fixed parameters in the E-step, then maximizing with respect to the parameters while keeping r_{ij} fixed in the M-step. The estimates from the M-step are the same as those in section 4.1, but are easier to compute now that the log is inside the sum.

5 Toward Variational EM

As mentioned, EM is a general algorithm to maximize the likelihood of observed data for a latent variable model. While the method discussed in the previous section is perfectly valid, it assumes that the form of the true posterior over the latent variables is known. In this section, we introduce a different form of EM that does not make this assumption.

Here, we'll avoid getting too specific about a particular generative model and say that we've sampled a set of IID random variables $x_i \sim P(x_i \mid \theta)$ for $i = 1, \ldots, n$. As before, we'll write out the likelihood

$$\mathcal{L}(\theta) = \prod_{i=1}^{N} p(x_i \mid \theta)$$
$$\ell(\theta) = \sum_{i=1}^{N} \log p(x_i \mid \theta)$$
$$= \sum_{i=1}^{N} \log \int p(x_i, z_i \mid \theta) dz_i$$

This last equality coming from the introduction of latent variables z_i and the law of total probability.[11] Now, for a key trick, we introduce a new distribution over the latent variables parameterized by ϕ, called $q(z_i \mid \phi)$ (this is just multiplying by 1).

$$\ell(\theta) = \sum_{i=1}^{N} \log \int \frac{p(x_i, z_i \mid \theta)}{q(z_i \mid \phi)} q(z_i \mid \phi) dz_i \tag{4}$$

This distribution q can be more or less anything, but typically we select a parametric distribution that we believe is a good candidate for the posterior $p(z \mid x)$. In many cases this may mean giving it the same functional form (in the Gaussian Mixture Model, it was also a Gaussian). The power of this formulation of EM is manifested in the case of intractable or complex posteriors, where q is an *approximation* to the true posterior (and could be, for instance, a neural network or a fully factorized distribution). This is termed *variational* EM.

We could try to maximize equation 4 directly, but the sum inside the logarithm makes life difficult. In general, it will be computationally intractable except for discrete z (and one that only takes on a small number of values, at that). Luckily, it is in the form of an expectation over $q(z_i \mid \phi)$. By Jensen's inequality, we know that

$$\mathbb{E}(f(x)) \leq f(\mathbb{E}(x))$$

for concave f.[12] Since log is concave, we can write the following:

$$\ell(\theta) = \sum_{i=1}^{N} \log \mathbb{E}_{q(z_i|\phi)} \left[\frac{p(x_i, z_i \mid \theta)}{q(z_i \mid \phi)} \right] \geq \sum_{i=1}^{N} \mathbb{E}_{q(z_i|\phi)} \left[\log \frac{p(x_i, z_i \mid \theta)}{q(z_i \mid \phi)} \right] \tag{5}$$

Hence, we have a lower bound on the marginal log-likelihood that we can optimize. This is known as the evidence lower-bound or ELBO. We will maximize this lower-bound in two iterative steps, described below.

It's worth noting that the intuition offered in section 4.1 on EM as applied to Gaussian Mixtures still applies in the general case: we will have posteriors[13] of the latent variables that get updated in the E-step using our best guesses for the parameters θ, and use these posteriors to update estimates of the parameters in the M-step.

5.1 E - Step

Rewriting the above a bit further:

$$\begin{aligned}
\ell(\theta) &\geq \sum_{i=1}^{N} \int \log \frac{p(z_i, x_i \mid \theta)}{q(z_i \mid \phi)} q(z_i \mid \phi) dz_i \\
&= \sum_{i=1}^{N} \int \log \frac{p(z_i \mid \theta, x_i) p(x_i \mid \theta)}{q(z_i \mid \phi)} q(z_i \mid \phi) dz_i \\
&= \sum_{i=1}^{N} \int \log \frac{p(z_i \mid \theta, x_i)}{q(z_i \mid \phi)} q(z_i \mid \phi) dz_i + \int \log p(x_i \mid \theta) q(z_i \mid \phi) dz_i \\
&= -KL\left(q(\boldsymbol{z} \mid \phi) \,||\, p(\boldsymbol{z} \mid \theta, \boldsymbol{x}) \right) + \ell(\theta)
\end{aligned} \tag{6}$$

The right-hand term of eq. 6 follows since $p(x_i \mid \theta)$ is not a function of z and because $\int q(z_i) dz_i = 1$ since it's a probability (rewritten in vectorized form for simplicity)

[11] By indexing by i, I've made an assumption in the model of one latent variable per observation. In principle, there could be greater/fewer and it wouldn't change these derivations.

[12] To restate Jensen, $tf(x_1) + (1-t)f(x_2) \leq f(tx_1 + (1-t)x_2)$ for $t \in [0, 1]$. The Wikipedia page has a good explanation

[13] although not necessarily the "true" posteriors

The left-hand term comes from the definition of Kullback–Leibler divergence, which roughly speaking measures the difference between two distributions.[14] Importantly, it will be zero when the two distributions are the same. As a result, we have induced the E-step of the algorithm:

$$\min_{\phi} \; KL\left(q(\boldsymbol{z} \mid \phi) \,||\, p(\boldsymbol{z} \mid \theta, \boldsymbol{x}) \right) + \ell(\theta) \tag{7}$$

In words, we are making q closer to the posterior p over the latent variables $\boldsymbol{z}$. In general, q can be an approximation to the posterior that has a different functional form (e.g., factorized Gaussians), but in the case of Gaussian Mixture Models, it will be exactly equal to the true posterior p.

5.2 M-step

In the E-step, the parameters θ are fixed while we maximize the q distribution. In the M-step, we consider q fixed and maximize θ. Rewriting eq. 5, we have

$$\begin{aligned}
\ell(\theta) &\geq \sum_{i=1}^{N} \mathbb{E}_{q(z_i|\phi)} \left[\log p(x_i, z_i \mid \theta)\right] - \mathbb{E}_{q(z_i|\phi)} \left[\log q(z_i \mid \phi)\right] \\
&= \sum_{i=1}^{N} \mathbb{E}_{q(z_i|\phi)} \left[\log p(x_i, z_i \mid \theta)\right] + \mathrm{H}(q)
\end{aligned}$$

Where $\mathrm{H}(q)$ is just the entropy of q and can be ignored when maximizing over θ in the M-step:

$$\max_{\theta} \; \sum_{i=1}^{N} \mathbb{E}_{q(z_i|\phi)} \left[\log p(x_i, z_i \mid \theta)\right] \tag{8}$$

So, we can think of q as our best guess, at a particular iteration, for the posterior over latents p that we can take expectations of in order to maximize θ.

References

[1] Scribed Notes from students taking a graphical models course at CMU (Huiting Liu, Yifan Yang) `https://www.cs.cmu.edu/~epxing/Class/10708-17/notes-17/10708-scribe-lecture8.pdf`

[2] Notes from a Professor at the Rotman School of Management (Brian Keng) `http://bjlkeng.github.io/posts/the-expectation-maximization-algorithm/`

[3] Notes from a PhD in Statistics at University of Chicago (Matt Bonakdarpour) `https://stephens999.github.io/fiveMinuteStats/intro_to_em.html`

[4] Notes from a Professor at Saarland University (Bernt Schiele) `https://schraudolph.org/teach/ml03/MLmix.pdf`

[5] Dissolving the Fermi Paradox (Anders Sandberg, Eric Drexler and Toby Ord) `https://arxiv.org/pdf/1806.02404.pdf`

[14] NB that it is not a distance because it is not symmetric and doesn't follow the triangle inequality.