CS 7140: Advanced Machine Learning

Lecture 8: Expectation Maximization (Wed 7 Feb 2018)

Instructor

Jan-Willem van de Meent (j.vandemeent@northeastern.edu)

Scribes

Sabbir Ahmad (ahmad.sub@husky.neu.edu)
Bahar Azari (azari@ece.neu.edu)

1 Expectation Maximization

1.1 Gaussian Mixtures: Gibbs Sampling

In the previous lectures, we have seen sampling methods in Generative Models to estimate the parameters for data clustering, where data is assumed to be generated from Gaussian Mixtures. For example, the Generative Model and the Gibbs Sampler updates for K Gaussian mixtures are specified as follows.

Generative Model	**Gibbs Sampler Updates**
$\mu_k, \Sigma_k \sim p(\mu, \Sigma)$	1. $z_n \sim p(z_n \vert y_n, \mu, \Sigma)$
$z_n \sim \text{Discrete}(\pi_1, \ldots, \pi_K)$	2. $\mu_k, \Sigma_k \sim p(\mu_k, \Sigma_k \vert y, z)$
$y_n \mid z_n = k \sim \mathcal{N}(\mu_k, \Sigma_k)$	

1.2 Gaussian Mixtures: Maximum Likelihood / MAP

We can do Maximum Likelihood (ML) or Maximum a Posteriori (MAP) to estimate the parameter θ of Gaussian Mixtures. Note that, here $\theta := \{\pi, \mu_{1:K}, \Sigma_{1:K}\}$ for a mixture of K Gaussians. The ML/MAP estimation problem in this setting amounts to having to estimate K multivariate Gaussians with mixing weights $\pi_1, \ldots, \pi_K$.

Now we can write for Maximum Likelihood estimation:

$$\theta = \underset{\theta}{\text{argmax}}\, p(y; \theta) = \underset{\theta}{\text{argmax}} \int p(y, z; \theta)\, dz \tag{1}$$

Here, $\int p(y, z; \theta) dz$ is the marginal probability of y over z, where z represents the cluster assignment for every point in the dataset. As we have seen before, computing the integral is often intractable and difficult to calculate. For maximum a Posteriori (MAP) estimation, we can write:

$$\theta = \underset{\theta}{\text{argmax}}\, p(y, \theta) = \underset{\theta}{\text{argmax}} \int p(y, z, \theta)\, dz. \tag{2}$$

1.3 Generalized hard K-means

ML and MAP estimation for Gaussian mixtures is complicated by the fact the we need to marginalize over z. Performing optimization conditioned on z is relatively straightforward. In this case we can solve

for the derivative

$$\nabla_\theta \log p(y, z; \theta) = \sum_{n=1}^{N} \nabla_\theta \log p(y_n, z_n; \theta) = 0. \tag{3}$$

As we will see below, this type of problem is analogous to the one that we already solved in the context of linear regression. It turns out that we can solve this problem in closed form for any likelihood $p(y \mid z, \theta^y)$ that is in the exponential family. When we marginalize over z, we obtain a different problem,

$$\nabla_\theta \log p(y; \theta) = \nabla_\theta \sum_{n=1}^{N} \log p(y_n; \theta), = \sum_{n=1}^{N} \frac{\nabla_\theta p(y_n; \theta)}{p(y_n; \theta)} \tag{4}$$

$$= \sum_{n=1}^{N} \frac{\nabla_\theta \sum_{k=1}^{K} p(y_n, z_n = k; \theta)}{p(y_n; \theta)} \tag{5}$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\nabla_\theta p(y_n, z_n = k; \theta)}{p(y_n; \theta)} = 0. \tag{6}$$

It so happens that this problem is generally not tractable in closed form (we will see further on why). For this reason, we will begin by solving a simpler problem, inspired by Gibbs sampling. In this problem we alternate between two maximization steps

$$\begin{array}{lll} \text{Step 1}: & z_n & = \text{argmax}_{z_n} p(y_n, z_n; \theta) \\ \text{Step 2}: & \mu_k, \Sigma_k & = \text{argmax}_{\mu_k, \Sigma_k} p(y_n, z_n, \theta) \quad [MAP] \\ & & = \text{argmax}_{\mu_k, \Sigma_k} p(y_n, z_n; \theta) \quad [ML] \end{array} \tag{7}$$

In Step 2, the mean and covariance are optimized by MAP or ML. Note that, the notation in MAP estimation, $p(y_n, z_n, \theta) = p(y_n|z_n, \theta)p(z_n|\theta)p(\theta)$, whereas, the ML estimation doesn't have any prior distribution on parameter. So for ML, $p(y_n, z_n; \theta) = p(y_n|z_n; \theta)p(z_n; \theta)$.

1.3.1 Updates for the parameters

Step 1 is trivial to implement (we can simply enumerate over all choices $z_n = k$). In order to implement Step 2, we will need to write down the the joint density, $p(y_n, z_n; \theta) = p(y_n|z_n; \theta)p(z_n; \theta)$, which is

$$p(y, Z; \theta) = \prod_{n=1}^{N} p(y_n, z_n; \theta) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(y_n, z_n = k; \theta)^{\mathbb{I}[z_n = k]}. \tag{8}$$

Here, $\mathbb{I}[z_n = k]$ is the indicator function, which is defined as,

$$\mathbb{I}[z_n = k] = \begin{cases} 1, & \text{if } z_n = k, \\ 0, & \text{if } z_n \neq k. \end{cases} \tag{9}$$

We can express the joint $p(y_n, z_n = k; \theta)$ in terms of the likelihood $p(y_n|z_n = k, \mu, \Sigma)$ and the cluster assignment prior $p(z_n = k; \pi)$,

$$p(y_n|z_n = k, \mu, \Sigma) = \mathcal{N}(y_n; \mu_k, \Sigma_k), \tag{10}$$

$$p(z_n = k; \pi) = \pi_k. \tag{11}$$

To compute the ML update for the mean, we now solve for the derivative with respect to μ_l, which must be 0 at the optimum

$$\frac{\partial}{\partial \mu_l} \log p(y, z; \theta) = \frac{\partial}{\partial \mu_l} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{I}[z_n = k] \log p(y_n, z_n = k; \theta), \tag{12}$$

$$= \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{I}[z_n = k] \frac{\partial}{\partial \mu_l} [\log \mathcal{N}(y_n; \mu_k, \Sigma_k) + \log \pi_k], \tag{13}$$

$$= \sum_{n=1}^{N} \mathbb{I}[z_n = l] \frac{\partial}{\partial \mu_l} \log \mathcal{N}(y_n; \mu_l, \Sigma_l), \tag{14}$$

$$= -\sum_{n=1}^{N} \mathbb{I}[z_n = l](y_n - \mu_l)\Sigma_l^{-1} = 0. \tag{15}$$

If we multiply on the right by $-\Sigma^l$, then we see that the optimum satisfies the condition

$$0 = \sum_{n=1}^{N} \mathbb{I}[z_n = l](y_n - \mu_l). \tag{16}$$

Solving for μ_l now gives us the update

$$\mu_l = \frac{1}{N_l} \sum_{n=1}^{N} \mathbb{I}[z_n = l] y_n, \qquad N_l = \sum_{n=1}^{N} \mathbb{I}[z_n = l]. \tag{17}$$

In other words, the optimal value μ_l simply the mean of the points y_n that are assigned to the cluster l.

We can derive updates for Σ_l and π_l in an analogous manner, which yields

$$\Sigma_l = \left(\frac{1}{N_l} \sum_{n=1}^{N} \mathbb{I}[z_n = l] y_n y_n^\top \right) - \mu_l \mu_l^\top, \qquad \pi_l = \frac{N_l}{N}. \tag{18}$$

1.3.2 Algorithm

Now that we have derived the closed-form solutions for updating the parameters for each iteration, we can formally present the algorithm. We call the Generalized version Hard K-means, as the assignment of each point to a cluster is absolute. That means, we are certain that a point belongs to a cluster completely at a given point in the algorithm. This is called the Hard assignment.

Algorithm: Hard K-means (Generalized version)

- Initialize: π, μ, Σ
- Repeat until z_n unchanged:
  1. For n in $1, \ldots, N$:
     $z_n = \text{argmax}_\mu \log \mathcal{N}(y_n; \mu_k, \Sigma_k) + \log \pi_k$
     $= \text{argmax}_\mu \, p(z_n = k | y; \theta)$
  2. For k in $1, \ldots, N$:
     $\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \mathbb{I}[z_n = k] y_n$
     $\Sigma_k = \frac{1}{N_l} \sum_{n=1}^{N} \mathbb{I}[z_n = k] y_n y_n^\top - \mu_k \mu_k^\top$
     $\pi_k = \frac{N_k}{N} \quad N_k = \sum_{n=1}^{N} \mathbb{I}[z_n = k]$

1.4 Soft K-means (Expectation Maximization)

Instead of hard assignment of the points to a cluster, if we are uncertain about the data points where they belong, we can assign a probability to each point to determine the feasibility of its belonging to a cluster. This type of algorithm is sometimes called called soft K-means, since it performs a "soft" assignment.

Algorithm: Soft K-means (Expectation Maximization)

- Initialize: π, μ, Σ
- Repeat until convergence:
  1. For n in $1, \ldots, N$:
     $\gamma_{nk} := \mathbb{E}_{z_n \sim p(z_n|y_n;\theta)}[\mathbb{I}[z_n = k]]$
     $N_k = \sum_{n=1}^{N} \gamma_{nk}$
  2. For k in $1, \ldots, N$:
     $\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} y_n$
     $\Sigma_k = \frac{1}{N_l} \sum_{n=1}^{N} \gamma_{nk} y_n y_n^\top - \mu_k \mu_k^\top$
     $\pi_k = \frac{N_k}{N}$

This expectation maximization (EM) algorithm iterates between two steps

1. **Expectation step:** Given the current parameters θ, we compute the responsibility γ_{nk} of the cluster for each point.
2. **Maximization step:** Given the current responsibilities γ_{nk}, we compute new values for the parameters θ.

Operationally this algorithm differs from hard K-means in that we replace $I[z_n = k]$ with the responsibilities γ_{nk} in the maximization step, and compute $\gamma_n k$ instead of the maximum posterior value for z_n in the expectation step.

1.4.1 Example

Figure 1 illustrates the EM algorithm for a Gaussian Mixture data with three clusters. The degree of redness indicates the degree to which the point belongs to the red cluster, and similarly for blue and green. As the iteration goes on, each point is assigned to one cluster with more possibility of being in that cluster. The last iteration can be seen from Figure 2(a).

1.4.2 Issues

One caveat of this algorithm is, it can converge to local optima. Random restart can be a solution for this problem, but this does not ensure reaching the global optima.

There can be another issue regarding implementation which is having an empty cluster. Figure 2(b) shows that the algorithm can reach to a local optima with one cluster having all the points while the other two clusters are empty.

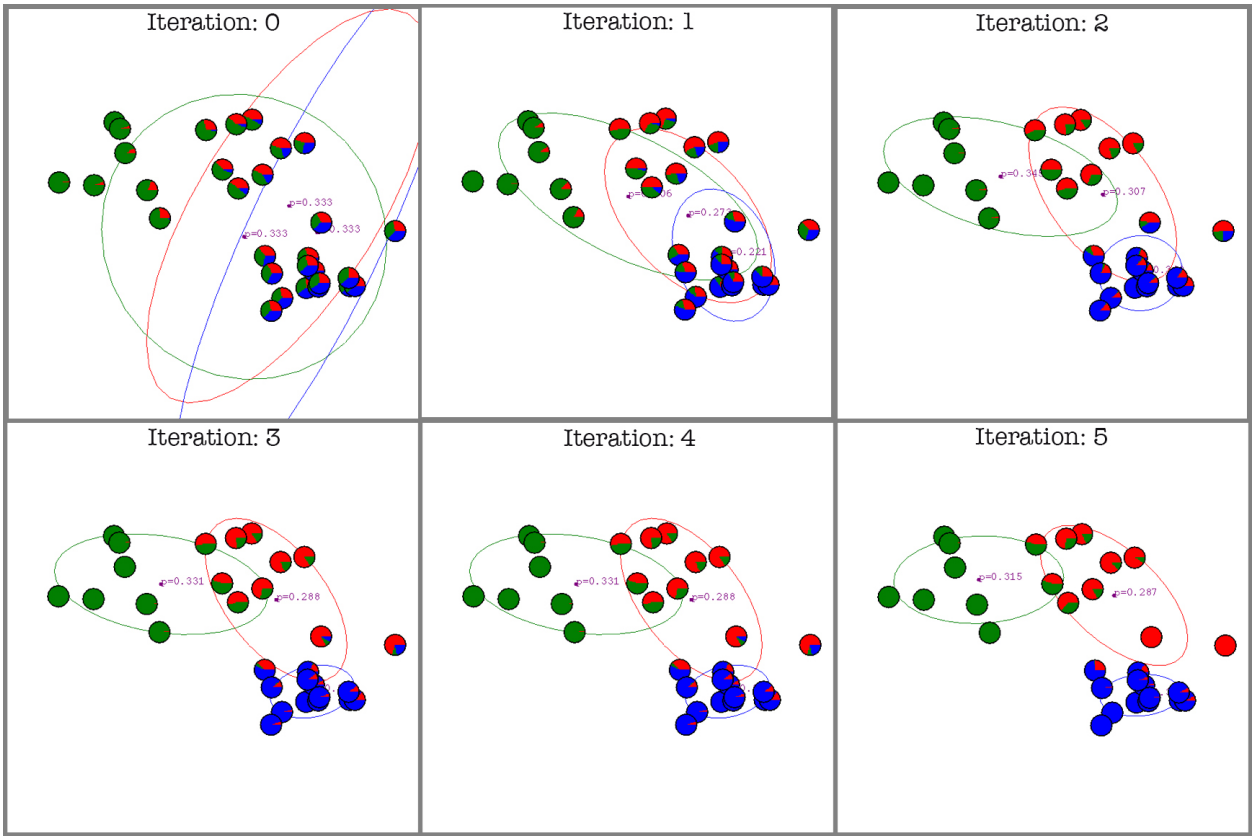

Figure 1: Iterations of Expectation Maximization

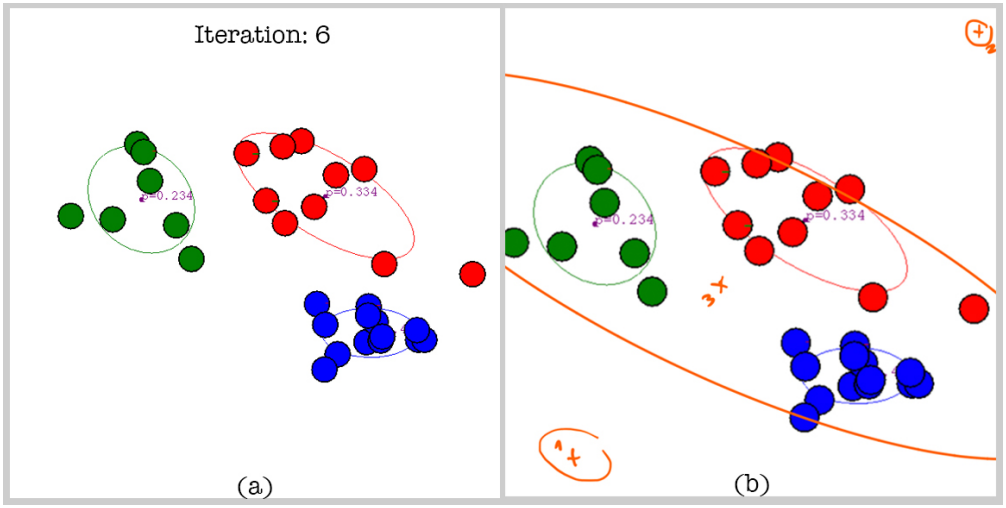

Figure 2: (a) Last iteration of the EM algorithm (b) EM reaching to local optima with two empty clusters

2 Generalized Expectation Maximization

A more formal derivation of expectation maximization algorithms can be obtained by defining a lower bound on the log likelihood. To do so, we need to make use of Jensen's inequality.

2.1 Jensen's Inequality

Jensen's inequality relates the expectation of a convex function to the convex function of an expectation. A convex function is a function where the area above the curve is a convex set. For a convex function, the line segment that connects any two points on the functions falls above the function. A concave function is the opposite of a convex function so, the opposite of the mentioned property holds, i.e., the line segment that connects any two points on the functions falls below the function.

For a convex and a concave function, you can write down the Jensen's inequality. Jensen's inequality stats: given that the function f is concave, the line segment that connects any two points on the functions falls below the function. For example, for any two points on the x axis, x_1 and x_2, if I choose any random point $x' = tx_1 + (1-t)x_2$ between them, where $t \in [0,1]$, following expression holds:

$$f(tx_1 + (1-t)x_2) \geq tf(x_1) + (1-t)f(x_2)) \tag{19}$$

and the opposite holds for a convex function:

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)) \tag{20}$$

An Example of convex and concave function is shown in Equation 3

Figure 3: A convex function (left) and a concave function (right)

Corollary: Jensen's inequality generalizes and particularly generalizes to random variables. For random variables we can look at the difference between a function of the expectation vs the expectation of a function. Taking an Expectation is like taking a sum over different terms so, the relationship between a function of expectation and expectation of a function is as follows:

$$\phi(\mathbb{E}[x]) \begin{cases} \leq \mathbb{E}[\phi(x)], & \text{if } \phi(x) \text{ convex.} \\ \geq \mathbb{E}[\phi(x)], & \text{if } \phi(x) \text{ concave.} \end{cases} \tag{21}$$

We now note that the logarithm is a concave function, which implies

$$\log(\mathbb{E}[x]) \geq \mathbb{E}[\log(x)]. \tag{22}$$

2.2 Lower Bounds on the Log Normalizing Constant

When we are doing probabilistic modeling, we can define this notion of normalizing constant. Suppose we have a random variable $x \sim \pi(x)$ where $\pi(x) = \gamma(x)/Z$ is a density for which we are only able to

evaluate the corresponding unnormalized density $\gamma(x)$. We can use the importance sampling trick to rewrite the integral for the normalizing constant as

$$Z := \int dx\, \gamma(x) = \int dx\; q(x)\, \frac{\gamma(x)}{q(x)} = \mathbb{E}_{X\sim q(x)}\left[\frac{\gamma(x)}{q(x)}\right]. \tag{23}$$

If we add a log inside our expectation, then we can use Jensen's inequality. So, we are getting the expected value of $\log \frac{\gamma(x)}{q(x)}$ w.r.t. an arbitrary distribution $q(x)$. According to Equation 24, by applying Jensen's inequality, this would be less than or equal to $logZ$.

$$\mathcal{L} := \mathbb{E}_{X\sim q(x)}\left[\log \frac{\gamma(x)}{q(x)}\right] \leq \log \mathbb{E}_{X\sim q(x)}\left[\frac{\gamma(x)}{q(x)}\right] = \log Z. \tag{24}$$

In the context of expectation maximization, we have an unnormalized density $\gamma(z;\theta) = p(y,z;\theta)$ with a normalizing constant $Z(\theta) = p(y;\theta)$. We can now introduce a distribution $q(z;\gamma)$ to define the lower bound

$$\mathcal{L}(\theta,\gamma) := \mathbb{E}_{z\sim q(z;\gamma)}\left[\log \frac{p(y,z;\theta)}{q(z;\gamma)}\right] \leq \log p(y;\theta). \tag{25}$$

If we now iterate between optimizing this bound with respect to γ and optimizing it with respect to θ, then we obtain the algorithm

Algorithm: Generalized EM

- Initialize: θ
- Repeat until $\mathcal{L}(\theta,\gamma)$ change below threshold
  1. **Expectation Step**

$$\gamma = \underset{\gamma}{\operatorname{argmax}}\, \mathcal{L}(\theta,\gamma) \tag{26}$$

  2. **Maximization Step**

$$\theta = \underset{\theta}{\operatorname{argmax}}\, \mathcal{L}(\theta,\gamma) \tag{27}$$

In this algorithm, the name "expectation" step is in some sense a misnomer; we are simply finding the parameters that maximize our objective function. That said, we will see that in practice, the expectation step generally computes some set of expected sufficient statistics that can then be used to perform the maximization step.

2.3 Intermezzo: Kullback-Leibler Divergence

The KL divergence between two densities $q(x)$ and $\pi(x)$ is defined as

$$\mathrm{KL}(q(x) \,\|\, \pi(x)) := \int dx\; q(x) \log \frac{q(x)}{\pi(x)}. \tag{28}$$

2.3.1 Properties

There are two properties associated with KL divergence. To prove the first property, if we start with the negative of KL divergence, we move the negative sign into the integral and inverse what is in the log and then using Jensen's inequality,

1. $\mathrm{KL}(q(x) \parallel \pi(x)) > 0$ (Positive Semi-definite)

$$\begin{aligned} Proof: -\mathrm{KL}(q(x) \parallel \pi(x)) &:= \int dx\ q(x) log \frac{\pi(x)}{q(x)} \\ &= \mathbb{E}_{X\sim q(x)}\left[log \frac{\pi(x)}{q(x)} \right] \\ &\leq log \mathbb{E}_{X\sim q(x)}\left[\frac{\pi(x)}{q(x)} \right] = log(1) = 0 \end{aligned} \tag{29}$$

$$\text{Because} \int dx q(x) \frac{\pi(x)}{q(x)} = \int dx \pi(x) = 1$$

2. $\mathrm{KL}(q(x) \parallel \pi(x)) = 0 \leftrightarrow q(x) = \pi(x)$

$$Proof: \int dx\ q(x) \log \frac{\pi(x)}{q(x)} = \int dx\ \pi(x) \log \frac{\pi(x)}{\pi(x)} = \int dx\ \pi(x) \log 1 = 0 \tag{30}$$

2.3.2 Relationship to Lower bound

We can relate the KL divergence to our lower bound. As it is shown in Equation 31 we write the definition of our lower bound and expand it by applying the Bayes rule. The $\log p(y;\theta)$ does not depend on z and comes out of the expectation. The other part is simply the KL divergence and because it is greater than or equal to zero, our lower bound is less than or equal to $\log p(y;\theta)$. An alternative (but equivalent) view is to interpret the expectation step as minimization of the Kullback-Leibler divergence between $q(z;\gamma)$ and the posterior on $p(z \mid y;\theta)$.

$$\begin{aligned} \mathcal{L}(\theta,\gamma) :=& \mathbb{E}_{z\sim q(z;\gamma)}\left[\log \frac{p(y,z;\theta)}{q(z;\gamma)} \right] \\ =& \mathbb{E}_{z\sim q(z;\gamma)}\left[\log p(y;\theta) + \log \frac{p(z \mid y;\theta)}{q(z;\gamma)} \right] \\ =& \log p(y;\theta) - \mathbb{E}_{z\sim q(z;\gamma)}\left[\log \frac{q(z;\gamma)}{p(z \mid y;\theta)} \right] \\ =& \log p(y;\theta) - \mathrm{KL}(q(z;\gamma) \parallel p(z \mid y;\theta)) \\ \leq& \log p(y;\theta) \end{aligned} \tag{31}$$

2.4 Role of KL minimization in Generalized Expectation Maximization

The key implication of Equation 31 is that *maximizing* $\mathcal{L}(\theta;\gamma)$ with respect to γ is equivalent to *minimizing* $\mathrm{KL}(q(z;\gamma) \parallel p(z \mid y;\theta))$ with respect to γ. Since the KL is 0 when $q(z;\gamma) = p(z \mid y;\theta)$, this choice of distribution also maximizes the lower bound.

Under this interpretation we see that the expectation step can alternately be interpreted as defining

$$q(z) := p(z \mid y;\theta). \tag{32}$$

An alternate interpretation is that we expectation step computes expected values relative to $p(z \mid y; \theta)$, which are then used to update the parameters. To see which expectations we need to compute in a mixture model, we will write out the derivative of the lower bound

$$\begin{aligned}\frac{\partial}{\partial \mu_l}\mathcal{L}(\theta,\gamma) &= \frac{\partial}{\partial \mu_l}\sum_{n=1}^{N}\mathbb{E}_{q(z_n)}\left[\log\frac{p(y_n,z_n;\theta)}{q(z_n)}\right] \\ &= \sum_{n=1}^{N}\mathbb{E}_{q(z_n)}\left[\frac{\partial}{\partial \mu_l}\log p(y_n,z_n;\theta)\right] \\ &= \sum_{n=1}^{N}\mathbb{E}_{q(z_n)}[\mathrm{I}[z_n=l]](y_n-\mu_l)\Sigma_l^{-1}\end{aligned} \tag{33}$$

In other words, we see that the update equations are analogous to the ones we derived for hard K-means, with the exception of the fact that we now replace $I[z_n = k]$ with the expected value

$$\gamma_{nk} = \mathbb{E}_{q(z_n)}[I[z_n=k]] \tag{34}$$
$$= \mathbb{E}_{p(z_n|y_n;\theta)}[I[z_n=k]] \tag{35}$$
$$= p(z_n = k \mid y_n;\theta) \tag{36}$$
$$= \frac{p(y_n,z_n=k\mid;\theta)}{\sum_{l=1}^{L}p(y_n,z_n=l\mid;\theta)} \tag{37}$$

When we put all of this together, we derive the algorithm

Algorithm: Generalized EM for Gaussian Mixtures

- Initialize: θ
- Repeat until $\mathcal{L}(\theta,\gamma)$ change below threshold

1. **Expectation Step:**

$$\gamma_{nk} \;=\; E_{p(z_n|y_n;\theta)}[I[z_n=k]] \;=\; p(z_n=k\mid y_n;\theta) \;=\; \frac{p(y_n,z_n=k;\theta)}{\sum_l p(y_n,z_n=l;\theta)}, \tag{38}$$

$$N_k = \sum_{n=1}^{N}\gamma_{nk}. \tag{39}$$

2. **Maximization Step:**

$$\begin{aligned}\mu_k &= \frac{1}{N_k}\sum_{n=1}^{N}\gamma_{nk}y_n, \\ \Sigma_k &= \left(\frac{1}{N_k}\sum_{n=1}^{N}\gamma_{nk}y_ny_n^\top\right)-\mu_k\mu_k^\top, \\ \pi_k &= \frac{N_k}{N}.\end{aligned} \tag{40}$$