Technical Details about the Expectation Maximization (EM) Algorithm

Dawen Liang
Columbia University
dliang@ee.columbia.edu

February 25, 2015

1 Introduction

Maximum Likelihood Estimation (MLE) is widely used as a method for estimating the parameters in a probabilistic model. The basic idea is to compute the parameter $\boldsymbol{\theta}^{\text{MLE}}$ where:

$$\boldsymbol{\theta}^{\text{MLE}} = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathrm{P}(\mathbf{X}|\boldsymbol{\theta})$$

$\mathrm{P}(\mathbf{X}|\boldsymbol{\theta})$ is the (observable) data likelihood. The parameter $\boldsymbol{\theta}$ is omitted sometimes for simple notation.

MLE is normally done by taking the derivative of the data likelihood $\mathrm{P}(\mathbf{X})$ with respect to the model parameter $\boldsymbol{\theta}$ and solving the equation. However, in some cases where we have hidden (unobserved) variables in the model, the derivative w.r.t. the model parameter does not have a close form solution. We will illustrate this problem with a simple example of mixture model with hidden variables.

1.1 An Example: Mixture of Bernoulli Distributions

Suppose we have N binary data points $x_1, x_2, \cdots, x_N$, each of which is i.i.d. drawn from one out of K Bernoulli Distribution with parameter q_k. Thus $p(x_n|q_k) = q_k^{x_n}(1-q_k)^{1-x_n}$. The probability of picking the kth Bernoulli component out of K is π_k, which is often referred as mixing proportion. We don't know beforehand which one out of K components each data point is drawn from, thus the variable representing these component assignments (will define later) is hidden in this mixture model. Write the data likelihood and log-likelihood:

$$\mathrm{P}(\mathbf{x}) = \prod_{n=1}^{N}\sum_{k=1}^{K}\pi_k p(x_n|q_k)$$

$$\log\mathrm{P}(\mathbf{x}) = \sum_{n=1}^{N}\log\sum_{k=1}^{K}\pi_k p(x_n|q_k)$$

with the parameter $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{q}\}$, and $\sum_k \pi_k = 1$.

Notice that the second summation is inside the $\log$, thus it is hard to decouple the parameters (which will lead to non-close form solution). Take the derivative w.r.t. q_k and set to 0, we obtain:

$$0 = \sum_{n=1}^{N}\frac{\pi_k}{\sum_{j=1}^{K}\pi_j p(x_n|q_j)}\cdot\frac{\partial p(x_n|q_k)}{\partial q_k}$$

where

$$\frac{\partial p(x_n|q_k)}{\partial q_k} = \left(\frac{x_n}{q_k} - \frac{1-x_n}{1-q_k}\right) p(x_n|q_k) \triangleq f(q_k; x_n) p(x_n|q_k)$$

Thus we have:

$$0 = \sum_{n=1}^{N} \frac{\pi_k p(x_n|q_k)}{\sum_{j=1}^{K} \pi_j p(x_n|q_j)} \cdot f(q_k; x_n)$$

The form of the $f(q_k; x_n)$ is not important in this case, we only need to know that it is a function of q_k. As for the first term in the summation, we define the hidden variables $\mathbf{z_n}$, representing the component assignment for data point x_n using a vector of size $K \times 1$. If x_n is drawn from the kth component, $z_{nk} = 1$ while the remaining are all 0. We could evaluate the posterior distribution of z_{nk}:

$$\begin{aligned} p(z_{nk} = 1|x_n, \boldsymbol{\theta}) =& \frac{p(z_{nk} = 1) p(x_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1) p(x_n|z_{nj} = 1)} \\ =& \frac{\pi_k p(x_n|q_k)}{\sum_{j=1}^{K} \pi_j p(x_n|q_j)} \end{aligned}$$

which is the first term in the summation. Since z_{nk} is a binary variable, the expectation $\mathbb{E}[z_{nk}] = p(z_{nk} = 1|x_n, \boldsymbol{\theta})$. We can think of this term as a "responsibility" of component k for data point x_n. Let's denote this term as $\gamma(z_{nk})$ for simplicity, following the notation from [1]. Now we know the MLE solution q_k^{MLE} must satisfy:

$$0 = \sum_{n=1}^{N} \gamma(z_{nk}) \cdot f(q_k; x_n)$$

It is impossible to have analytical solution, because in order to compute q_k^{MLE}, we need to compute $\gamma(z_{nk})$ which is dependent on q_k itself. This is not a coincident, we will see this again shortly when we derive the MLE solution for mixing proportion.

For the mixing proportion π_k, take the derivative with a Lagrange multiplier λ and set to 0, we obtain:

$$0 = \sum_{n=1}^{N} \frac{p(x_n|q_k)}{\sum_{j=1}^{K} \pi_j p(x_n|q_j)} + \lambda$$

We can multiply π_k on both sides and sum over k:

$$0 = \sum_{n=1}^{N} \frac{\sum_{k=1}^{K} \pi_k p(x_n|q_k)}{\sum_{j=1}^{K} \pi_j p(x_n|q_j)} + \sum_{k=1}^{K} \pi_k \lambda$$

Making use of use of the fact that $\sum_k \pi_k = 1$, we obtain: $\lambda = -N$. Substitute λ back to the derivative and multiply both side with π_k, we obtain:

$$\sum_{n=1}^{N} \frac{\pi_k p(x_n|q_k)}{\sum_{j=1}^{K} \pi_j p(x_n|q_j)} = \pi_k N$$

$$\sum_{n=1}^{N} \gamma(z_{nk}) = \pi_k N$$

We could define $\sum_{n=1}^{N} \gamma(z_{nk}) = N_k$, where N_k can be interpreted as the "expected" number of data points drawn from component k. Therefore:

$$\pi_k = \frac{N_k}{N}$$

which is again dependent on $\gamma(z_{nk})$, while $\gamma(z_{nk})$ depends on π_k.

This example of mixture of Bernoulli distributions illustrates the difficulty to directly maximize the likelihood for models with hidden variables. However, we could get some intuition about an iterative algorithm where the derivation above can be made use of: Start the algorithm by randomly initializing the parameter. Then in each iterative step, compute the $\gamma(z_{nk})$ based on the old parameter. Then the new parameter can be updated accordingly with the current value of $\gamma(z_{nk})$. This is the basic intuition behind Expectation Maximization (EM) algorithm.

2 EM in General

One of the problems with directly maximizing the observable data likelihood, as shown above, is that the summation is inside the logarithm. So what if we move on to the complete data likelihood $\mathrm{P}(\mathbf{X}, \mathbf{Z})$ and then marginalize the hidden variable $\mathbf{Z}$? Let's do the derivation[1].

Start from the log-likelihood:

$$\log \mathrm{P}(\mathbf{X}|\boldsymbol{\theta}) = \log \sum_{\mathbf{Z}} \mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

Here we make use of the variational point of view by adding a variational distribution $q(\mathbf{Z})$ and the fact that logarithm function is concave:

$$\begin{aligned}
\log \mathrm{P}(\mathbf{X}|\boldsymbol{\theta}) &= \log \Big(\sum_{\mathbf{Z}} \frac{\mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} q(\mathbf{Z}) \Big) \\
&\geq \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \Big(\frac{\mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \Big)
\end{aligned}$$

By making use of the Jensen's inequality, we move out the summation from the logarithm successfully. We could of course find out $q(\mathbf{Z})$ by exploring when the equality holds for Jensen's inequality. However, we will solve it from a different approach here. We first want to learn how much we have lost from the Jensen's inequality:

$$\begin{aligned}
\Delta &= \log \mathrm{P}(\mathbf{X}|\boldsymbol{\theta}) - \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \Big(\frac{\mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \Big) \\
&= \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \mathrm{P}(\mathbf{X}|\boldsymbol{\theta}) - \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \Big(\frac{\mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \Big) \\
&= \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \Big(\frac{\mathrm{P}(\mathbf{X}|\boldsymbol{\theta}) q(\mathbf{Z})}{\mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})} \Big) \\
&= \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \Big(\frac{q(\mathbf{Z})}{\mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})} \Big) \\
&= \mathrm{KL}(q \| p)
\end{aligned}$$

[1] There are actually a lot of different versions of derivations for EM algorithm. The one presented here gives the most intuition to the author.

Thus, the difference is actually the KL-divergence between the variational distribution $q(\mathbf{Z})$ and the posterior distribution $\mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$. Thus, we could rearrange the log-likelihood as:

$$\log \mathrm{P}(\mathbf{X}|\boldsymbol{\theta}) = \underbrace{\sum_{\mathbf{Z}} q(\mathbf{Z}) \log \Big(\frac{\mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \Big)}_{\mathcal{L}(q, \boldsymbol{\theta})} + \mathrm{KL}(q \| p)$$

Since the KL-divergence is non-negative for any q and p, $\mathcal{L}(q, \boldsymbol{\theta})$ acts as a lower-bound for the log-likelihood. Let's denote the log-likelihood as $\ell(\boldsymbol{\theta})$ for simplicity.

EM algorithm has 2 steps as its name suggests: Expectation(E) step and Maximization(M) step.

In the E step, from the variational point of view, our goal is to choose a proper distribution $q(\mathbf{Z})$ such that it best approximates the log-likelihood. At this moment, we have existing parameter $\boldsymbol{\theta}^{\text{old}}$. Thus we set the variational distribution $q(\mathbf{Z})$ equal the posterior distribution $\mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$ so that $\mathrm{KL}(q \| p) = 0$. In that case, we make the lower-bound $\mathcal{L}(q, \boldsymbol{\theta})$ equal $\ell(\boldsymbol{\theta})$.

In the M step, we have $q(\mathbf{Z})$ fixed and maximize the $\mathcal{L}(q, \boldsymbol{\theta})$, which is equivalent to maximize $\ell(\boldsymbol{\theta})$, w.r.t. the parameter $\boldsymbol{\theta}$. Unless we reach the convergence, the lower-bound $\mathcal{L}(q, \boldsymbol{\theta})$ will increase with the new parameter $\boldsymbol{\theta}^{\text{new}}$. Since the parameter $\boldsymbol{\theta}$ changes from the E step, KL-divergence no longer equals 0, which creates gap between $\mathcal{L}(q, \boldsymbol{\theta})$ and $\ell(\boldsymbol{\theta})$ again. And this gap will be filled out in the next E step.

To see analytically the objective function in M step, substitute $q(\mathbf{Z})$ with $\mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$ from E step:

$$\begin{aligned}
\mathcal{L}(q, \boldsymbol{\theta}) &= \sum_{\mathbf{Z}} \mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \log \Big(\frac{\mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{\mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})} \Big) \\
&= \underbrace{\sum_{\mathbf{Z}} \mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \log \mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}_{\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})} - \mathcal{H}\{\mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})\}
\end{aligned}$$

where $\mathcal{H}\{\mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})\}$ represents the negative entropy of $\mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$, which is irrelevant to the parameter $\boldsymbol{\theta}$. Thus we could consider it as a constant. What really matters is $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$ which we could view as the expectation of $\mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ under the posterior distribution $\mathrm{P}(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}})$. There are a few very nicely-drawn figures, visualizing the whole procedures of EM algorithm in Chapter 9 of [1].

The sweet spot in M step is that, instead of directly maximizing $\mathrm{P}(\mathbf{X})$ which does not have a close form solution, computing $\mathrm{P}(\mathbf{X}, \mathbf{Z})$ is generally much simpler because we can just view it as a model with no hidden variables.

EM algorithm is usually referred as a typical example of *coordinate ascent*, where in each E/M step, we have one variable fixed ($\boldsymbol{\theta}^{\text{old}}$ in E step and $q(\mathbf{Z})$ in M step), and maximize w.r.t. another one. Coordinate ascent is widely used in numerical optimization.

3 EM Applications in the Mixture Models

3.1 Mixture of Bernoulli Revised

Now let's go back to the problem we encountered earlier on mixture of Bernoulli distributions.

Assume we have some pre-set initial values $\boldsymbol{\theta}^{\text{old}} = \{\pi_k^{\text{old}}, q_k^{\text{old}}\}$ for parameter. We first write the complete likelihood and log-likelihood:

$$\mathrm{P}(\mathbf{x}, \mathbf{Z}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left(\pi_k \cdot q_k^{x_n} (1 - q_k)^{1 - x_n}\right)^{z_{nk}}$$

$$\log \mathrm{P}(\mathbf{x}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \big(\log \pi_k + x_n \log q_k + (1 - x_n) \log(1 - q_k) \big)$$

Note how this one differs from the observable data log-likelihood in Section 1.1. What we want to maximize in the M step is actually $\mathbb{E}_{\mathbf{z}}[\log \mathrm{P}(\mathbf{x}, \mathbf{Z})]$ where $\mathbf{Z}$ has been decomposed as z_{nk} in the complete log-likelihood. Also we have already shown that $\mathbb{E}(z_{nk}) = \gamma(z_{nk})$ in Section 1.1, thus:

$$\mathbb{E}_{\mathbf{Z}}[\log \mathrm{P}(\mathbf{x}, \mathbf{z})] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \big(\log \pi_k + x_n \log q_k + (1 - x_n) \log(1 - q_k) \big)$$

Take the derivative w.r.t. q_k and set to 0, we obtain:

$$q_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

As for π_k, similarly, we obtain:

$$\pi_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N}$$

To summarize the 2 steps of EM algorithm for the mixture of Bernoulli distributions:
E step: Compute $\gamma(z_{nk})$ with current parameter $\boldsymbol{\theta}^{\text{old}} = \{\pi_k^{\text{old}}, q_k^{\text{old}}\}$:

$$\gamma(z_{nk}) = p(z_{nk} = 1 | x_n, \boldsymbol{\theta}) = \frac{\pi_k^{\text{old}} p(x_n | q_k^{\text{old}})}{\sum_{j=1}^{K} \pi_j^{\text{old}} p(x_n | q_j^{\text{old}})}$$

M step: Update π_k and q_k:

$$\pi_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N}$$

$$q_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

$$\{\pi_k^{\text{new}}, q_k^{\text{new}}\} \to \{\pi_k^{\text{old}}, q_k^{\text{old}}\}$$

3.2 Mixture of Gaussian Distributions

A common scenario for applying EM algorithm is to estimate the parameter for mixture of Gaussian distributions, or Gaussian Mixture Models (GMM). The EM solution for GMM is actually very similar to the one for mixture of Bernoulli distributions derived above. Now assume we have N vectors of D dimensions $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ each of which is drawn i.i.d. from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ with mixing proportion π_k. Define the hidden variable $\mathbf{z}_n$ the same as in Section 1.1. Write the complete log-likelihood:

$$\log \mathrm{P}(\mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \big(\log \pi_k + \log \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \big)$$

where the expectation of complete log-likelihood under the posterior distribution follows:

$$\mathbb{E}_{\mathbf{Z}}[\log \mathrm{P}(\mathbf{X}, \mathbf{Z})] = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \big(\log \pi_k + \log \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \big)$$

Note in GMM, the kth component follows $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, thus:

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Similarly from the derivation in the mixture of Bernoulli distributions, we can obtain the $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and π_k by taking the derivative and setting to 0:

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^{\mathrm{T}} (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

$$\pi_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N}$$

We could see that these derivations are almost exactly the same with the mixture of Bernoulli distributions, except that in GMM there is one more parameter, covariance matrix $\boldsymbol{\Sigma}_k$, to estimate. Formally summarize the 2 steps of EM algorithm for GMM:

E step: Compute $\gamma(z_{nk})$ with current parameter $\boldsymbol{\theta}^{\text{old}} = \{\pi_k^{\text{old}}, \boldsymbol{\mu}_k^{\text{old}}, \boldsymbol{\Sigma}_k^{\text{old}}\}$:

$$\gamma(z_{nk}) = p(z_{nk} = 1 | \mathbf{x}_n, \boldsymbol{\theta}) = \frac{\pi_k^{\text{old}} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k^{\text{old}}, \boldsymbol{\Sigma}_k^{\text{old}})}{\sum_{j=1}^{K} \pi_j^{\text{old}} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j^{\text{old}}, \boldsymbol{\Sigma}_j^{\text{old}})}$$

M step: Update π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$:

$$\pi_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N}$$

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^{\mathrm{T}} (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

$$\{\pi_k^{\text{new}}, \boldsymbol{\mu}_k^{\text{new}}, \boldsymbol{\Sigma}_k^{\text{new}}\} \rightarrow \{\pi_k^{\text{old}}, \boldsymbol{\mu}_k^{\text{old}}, \boldsymbol{\Sigma}_k^{\text{old}}\}$$

3.3 Thoughts on the mixture models

A recent blog post[2] by Larry Wasserman pointed out the merit and defect of mixture models (mainly from the view of a statistician). Clearly Larry is not a fan of mixture models. I do agree with some of the points. However, as a not-too-deep-math machine learner, I still find mixture models (especially GMM) capable to do the job in some applications (which theoretical result doesn't care about).

[2] http://normaldeviate.wordpress.com/2012/08/04/mixture-models-the-twilight-zone-of-statistics/

4 Variations of the EM

EM algorithm can be thought of as an example of a generalized algorithm named GEM (generalized EM), where in the M step, the maximization can be only partially implemented. In this case, the likelihood can still be increased. Similarly, the E step can also be partially performed, which could actually lead to an incremental algorithm, as summarized in [2].

The EM algorithm described in Section 2 (I will call it normal EM) totally separate the 2 steps, which is computationally inefficient. In the E step, we calculate the component "responsibility" $\gamma(z_{nk})$ for each data point with each possible component assignment, and store all of the $N \cdot K$ values, as in the M step, all of them are required as reflected from the update rule. What incremental EM does is to make these 2 steps coherent. As proved in [2], both $q(\mathbf{Z})$ and $\log \mathrm{P}(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ can be factorized according to the data points, assuming the data points are independent. Furthermore, this could lead to the factorization of $\mathcal{L}(q, \boldsymbol{\theta})$, with local maxima unchanged:

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{n=1}^{N} \mathcal{L}_n(q_n, \boldsymbol{\theta})$$

where

$$\mathcal{L}_n(q_n, \boldsymbol{\theta}) = \mathbb{E}_{q_n}[\log \mathrm{P}(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta})] + \mathcal{H}\{q_n\}$$

Note that the factorization (product) is summation in the logarithm domain. We have shown that in the EM algorithm, both E and M steps can be considered as maximizing $\mathcal{L}(q, \boldsymbol{\theta})$ in a coordinate ascent manner, thus an incremental version of the EM is described in Algorithm 1:

Algorithm 1: EM algorithm with partially implemented E step

Initialization:

Randomly initialize $\boldsymbol{\theta}^{\text{old}}$ and q_n. q_n does not have to be consistent with $\boldsymbol{\theta}^{\text{old}}$;

Repeat until convergence:

- **E step**:
  Select some data point i to be updated:
  - Set q_j for the data points where $j \neq i$ unchanged.
  - Updated q_i to maximize $\mathcal{L}_i(q_i, \boldsymbol{\theta}^{\text{old}})$, given by the value $p(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\theta}^{\text{old}})$.
- **M step**:
  Update the parameter the same as the normal EM algorithm.

We can make some strategic decision for choosing data points, e.g. choose data points which we are more uncertain about their component assignment (q_i is not stabilized yet). However, essentially, this variation does not help with the inefficiency problem mentioned earlier. As we could see, the bottleneck is in the M step. To address this problem, assume the complete log-likelihood can be summarized in some form of the sufficient statistics $\mathrm{S}(\mathbf{X}, \mathbf{Z})$ (e.g. exponential family) and factorized as:

$$\mathrm{S}(\mathbf{X}, \mathbf{Z}) = \sum_{n=1}^{N} \mathrm{S}_n(\mathbf{x}_n, \mathbf{z}_n) \tag{1}$$

We can rewrite the normal EM algorithm in the form of sufficient statistics (omitted here). However, we could make use of the property of the sufficient statistics. In each iteration, we only compute how much $\mathrm{S}(\mathbf{X}, \mathbf{Z})$ will be increased given some of the chosen data points and then we can directly obtain the updated parameter given updated sufficient statistics. This incremental EM is described as in Algorithm 2.

Algorithm 2: Incremental EM algorithm

Initialization:

Randomly initialize $\boldsymbol{\theta}^{\text{old}}$ and $\mathrm{S}_n^{\text{old}}$. $\mathrm{S}_n^{\text{old}}$ does not have to be consistent with $\boldsymbol{\theta}^{\text{old}}$. S^{old} is computed given Eq. 1.

Repeat until convergence:

- **E step**:
  Select some data point i to be updated:
  - Set S_j where $j \neq i$ unchanged from the previous iteration.
  - Set $\mathrm{S}_i = \mathbb{E}_{q_i}[\mathrm{S}_i(\mathbf{x}_i, \mathbf{z}_i)]$, where $q_i = p(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\theta}^{\text{old}})$.
  - Set $\mathrm{S} = \mathrm{S}^{\text{old}} - \mathrm{S}_i^{\text{old}} + \mathrm{S}_i$
- **M step**:
  Update $\boldsymbol{\theta}$ based on the current value of S.

Note that in Algorithm 2, both E and M steps are independent of the number of data points. As mentioned above, the benefit comes from the incremental sufficient statistics, as it reflects the changes to the complete log-likelihood immediately. Thus speedy convergence can be achieved as we fully utilize the intermediate computation.

A sparse variant of the incremental EM can be further proposed [2]. The intuition behind this setting comes from the fact that in many cases, only a small portion of all the possible values of hidden variable $\mathbf{Z}$ has non-negligible probability. Substantial computation may be saved if we could "freeze" those negligible probability for many iterations, and only update those are chosen as plausible values. Such EM algorithm can still guarantee increasing the log-likelihood after each iteration, while mostly importantly, it is computationally efficient.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] R.M. Neal and G.E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *NATO ASI Series D, Behavioural and Social Sciences*, 89:355–370, 1998.