Expectation Maximization (EM) Algorithm

Gene Katsevich, Kenneth Tay, Stephen Bates

July 18, 2019

1 Introduction

For many interesting statistical models, the likelihood is non-convex and hence difficult to work with. As a result, in order to estimate the parameters by maximum likelihood we must turn to algorithms for nonconvex optimization, and the EM algorithm is one useful example. EM is particularly appealing for statistical models involving latent variables, because in these models the EM steps can often be formulated analytically and executed quickly.

2 Motivating example

As a first example, we will consider a mixture model based on a question from problem 2 of the 2005 Applied qual. We observe lifetimes T_i of n bacteria. It is believed that some small unknown fraction ε of the n bacteria have exponential distribution with mean μ, while the remaining bacteria have exponential distribution with mean 1. How can we estimate ε and μ?

We can try the maximum likelihood approach. Since

$$T_1, \ldots, T_n \overset{i.i.d.}{\sim} (1-\varepsilon) \cdot \text{Exp}(1) + \varepsilon \cdot \text{Exp}(\mu),$$

the log-likelihood is

$$\ell(\varepsilon, \mu; T) = \sum_{i=1}^{n} \log\left(\varepsilon \cdot \mu^{-1} e^{-T_i/\mu} + (1-\varepsilon) e^{-T_i}\right).$$

The mixture aspect of this problem has led to the log of a sum in $\ell(\varepsilon, \mu; T)$, which makes find the MLE difficult:

- There is no closed form for $(\hat{\varepsilon}, \hat{\mu})$, so an iterative approach will be necessary, and
- $\ell(\varepsilon, \mu; T)$ is non-convex in (ε, μ).

The EM algorithm is an iterative approach to maximizing a likelihood, designed for the case when there are latent variables (or missing data) in the problem. In our example, we take the latent variables to be the identities of the mixture component each bacteria belongs to.

2.1 Complete data log-likelihood

For each $i = 1, \ldots, n$, let Z_i be the random variable such that $Z_i = 0$ means bacteria i belongs to the Exp(1) component, while $Z_i = 1$ means it belongs to the Exp(μ) component. This means that

$$T_i \mid Z_i = 0 \sim \text{Exp}(1), \qquad T_i \mid Z_i = 1 \sim \text{Exp}(\mu).$$

Wouldn't it be great if we got to observe the Z_i's? Let's indulge ourselves and pretend for a moment that we have observed these latent variables. This leads to the **complete data likelihood**:

$$L(\varepsilon, \mu; T, Z) = \prod_{i=1}^{n} \varepsilon^{Z_i}(1-\varepsilon)^{1-Z_i} \left(\frac{1}{\mu} e^{-T_i/\mu}\right)^{Z_i} \left(e^{-T_i}\right)^{1-Z_i}. \tag{1}$$

Taking a log, we get the complete data log-likelihood

$$\begin{aligned} \ell(\varepsilon, \mu; T, Z) &= \sum_{i=1}^{n} \left[Z_i \log \varepsilon + (1 - Z_i) \log(1-\varepsilon) - Z_i \log \mu - \frac{T_i Z_i}{\mu} - T_i(1 - Z_i) \right] \\ &= \log \varepsilon \sum_{i=1}^{n} Z_i + \log(1-\varepsilon) \sum_{i=1}^{n} (1 - Z_i) - \log \mu \sum_{i=1}^{n} Z_i - \frac{1}{\mu} \sum_{i=1}^{n} T_i Z_i + C. \end{aligned} \tag{2}$$

We no longer have logs of sums and we can easily derive the MLE:

$$\hat{\varepsilon} = \frac{\sum_i Z_i}{n}, \qquad \hat{\mu} = \frac{\sum_i T_i Z_i}{\sum_i Z_i}.$$

That's very nice, and exactly what we would have expected. Of course, we can't do this because we don't know the Z_i!

2.2 Expected complete data log-likelihood

One way around this is to guess what the values of the Z_i are. Can we do better instead of random guessing? In the spirit of an iterative algorithm, let's assume that we have some

guess for the parameters $(\hat{\varepsilon}^k, \hat{\mu}^k)$. We can then plug in the expectations of the Z_i's under this guess into the complete data log-likelihood (2).

Let

$$\begin{aligned}\hat{\pi}_i^k &= \mathbb{E}_{\hat{\varepsilon}^k,\hat{\mu}^k}\left[Z_i \mid T_i\right] = \mathbb{P}_{\hat{\varepsilon}^k,\hat{\mu}^k}\left[Z_i = 1 \mid T_i\right] \\ &= \frac{\hat{\varepsilon}^k \frac{1}{\hat{\mu}^k} e^{-T_i/\hat{\mu}^k}}{(1-\hat{\varepsilon}^k)e^{-T_i} + \hat{\varepsilon}^k \frac{1}{\hat{\mu}^k} e^{-T_i/\hat{\mu}^k}}.\end{aligned}$$

Here, $\hat{\pi}_i^k$ reflects how likely it is that T_i was drawn from $\text{Exp}(\mu)$. Then, we can write down the **expected complete data log-likelihood**:

$$\begin{aligned}\tilde{\ell}^k(\varepsilon, \mu; T) &= \mathbb{E}_{\hat{\varepsilon}^k,\hat{\mu}^k}\left[\ell(\varepsilon, \mu; T, Z) \mid T\right] \\ &= \log\varepsilon \sum_{i=1}^n \hat{\pi}_i^k + \log(1-\varepsilon)\sum_{i=1}^n (1-\hat{\pi}_i^k) - \log\mu \sum_{i=1}^n \hat{\pi}_i^k - \frac{1}{\mu}\sum_{i=1}^n T_i \hat{\pi}_i^k.\end{aligned} \tag{3}$$

That is a nice expression! We can easily optimize this to get our next guess for ε and μ:

$$\hat{\varepsilon}^{k+1} = \frac{\sum_i \hat{\pi}_i^k}{n}, \qquad \hat{\mu}^{k+1} = \frac{\sum_i \hat{\pi}_i^k T_i}{\sum_i \hat{\pi}_i^k}. \tag{4}$$

This is in essence what the EM algorithm is: (3) is the E (Expectation) step, while (4) is the M (Maximization) step.

3 EM in general

Assume that we have data X and latent variables Z, jointly distributed according to the law $p_\theta(X, Z)$. This joint law is easy to work with, but because we do not observe Z, we must deal with

$$\log p_\theta(X) = \log\left[\sum_z p_\theta(X, Z = z)\right].$$

There's the log of a sum again! Let's try to get around it in the following way: Let θ^k be the current estimate of θ. Then

$$\begin{aligned}\ell(\theta) = \log(p_\theta(X)) &= E_{Z\sim p_{\theta^k}(Z|X)}\left[\log(p_\theta(X))\right] \\ &= E_{Z\sim p_{\theta^k}(Z|X)}\left[\log\left(\frac{p_\theta(X,Z)}{p_\theta(Z|X)}\right)\right] \\ &= \underbrace{E_{Z\sim p_{\theta^k}(Z|X)}\left[\log\left(p_\theta(X,Z)\right)\right]}_{\tilde{l}(\theta;\theta^k)} - \underbrace{E_{Z\sim p_{\theta^k}(Z|X)}\left[\log\left(p_\theta(Z|X)\right)\right]}_{R(\theta;\theta^k)}\end{aligned}$$

We thus have an neat decomposition of $\log(p_\theta(X))$ involving the expected complete data log-likelihood, and we can turn this into a lower bound by noting that $R(\theta;\theta^k) \leq R(\theta^k;\theta^k)$, because

$$\begin{aligned} R(\theta;\theta^k) - R(\theta^k;\theta^k) &= E_{Z\sim p_{\theta^k}(Z|X)}\left[\log\left(\frac{p_\theta(Z|X)}{p_{\theta^k}(Z|X)}\right)\right] \\ &\leq \log E_{Z\sim p_{\theta^k}(Z|X)}\left[\frac{p_\theta(Z|X)}{p_{\theta^k}(Z|X)}\right] \\ &= \log E_{Z\sim p_\theta(Z|X)}[1] = 0 \end{aligned}$$

Consequently, we have the following lower bound for the log-likelihood:

$$\ell(\theta) = \log(p_\theta(X)) \geq \tilde{\ell}(\theta;\theta^k) + R(\theta^k;\theta^k).$$

Note that if we cannot closed-form updates for the M step, we could take a Newton or gradient step instead, and by the argument above, as long as we increase $\tilde{\ell}(\theta;\theta^k)$, the expected log-likelihood $\ell(\theta)$ will increase. Notice that this lower bound is tight at $\theta = \theta^k$. Furthermore, only the first term depends on θ, so maximizing $\tilde{\ell}(\theta;\theta^k)$ over θ will yield a new point with higher log-likelihood, as shown in the figure below. Motivated by the above observation, the EM algorithm proceeds in two steps:

1. **E-step:** compute $\tilde{\ell}(\theta;\theta^k)$
2. **M-step:** find θ to maximize $\tilde{\ell}(\theta;\theta^k)$

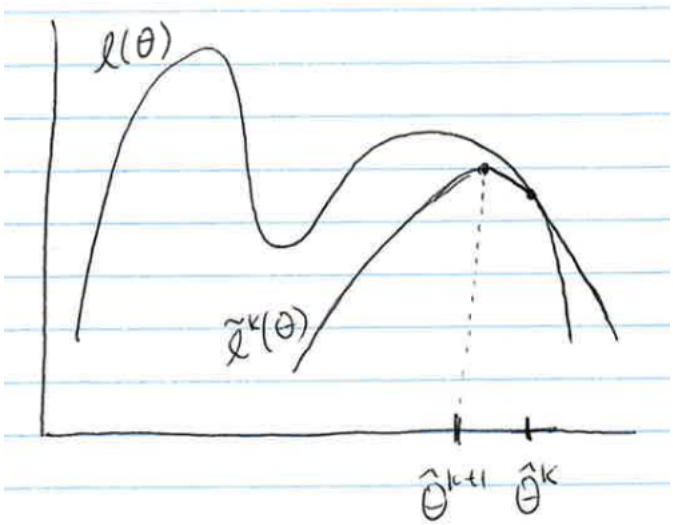

Figure 1: A visualization of an EM step.

By the above remarks, we see that the EM algorithm is an ascent method; **the likelihood increases at each step**. While this is reassuring, this does not imply that the algorithm

finds a global optimum. As a result, one typically does several runs of EM with different starting values and chooses the resulting estimate with the highest likelihood.

Remarks

The EM algorithm is one of many optimization algorithms. Its convergence can be slow depending on the problem, and other methods might outperform it. Nevertheless, EM is popular among statisticians for problems involving latent variables, because it has a n intuitive statistical structure and often has closed-form updates at each iteration.

Note that if we cannot closed-form updates for the M step, we could take a Newton or gradient step instead, and by the argument above, as long as we increase $\tilde{\ell}(\theta; \theta^k)$, the log-likelihood $\ell(\theta)$ will increase.