A GENTLE INTRODUCTION TO THE EM ALGORITHM. PART I: THEORY*

HEMANT D. TAGARE

1. Introduction. My aim is to introduce the Expectation-Maximization (EM) algorithm to you; especially some of its theory. I will skip proofs, but I will derive many formulae that have practical use. The EM algorithm is iterative and you should be familiar with its convergence properties. I will discuss them in detail.

I will present applications of the EM algorithm to signal and image processing in a companion tutorial called "A Gentle Introduction to the EM algorithm. Part II: Applications."

There are two excellent books about the EM algorithm [1, 2]. Martin Tanner's book is concise and deceptively simple looking. Every line in that book counts. Read it carefully. MacLachlan and Krishnan's book is more comprehensive and has many extensions of the EM algorithm. I learned a lot from it. I recommend both books to the EM enthusiast. Be warned that these books are written by statisticians for statisticians. If you are not a statistician, then some of the notation might be unfamiliar.

I cannot introduce the EM algorithm without discussing parameter estimation. So without any more fuss, let me review the principles of parameter estimation. I will assume that you understand the basic principles of probability – prior, joint, and conditional distributions and the Bayes rule.

2. Estimation Principles. In many scientific problems, the measurement vector $x = (x_1, \cdots, x_n)^T$ changes from experiment to experiment and its probability density $p(x)$ depends on some parameters $\theta = (\theta_1, \cdots, \theta_m)^T$. To show the dependence explicitly we write the probability density as $p(x \mid \theta)$.

We conduct one experiment, obtain a specific value of x and are asked to deduce a numerical value for θ using our knowledge of $p(x \mid \theta)$. This is a serious dilemma because *any* value of θ for which $p(x \mid \theta)$ is not zero could have caused the observation x. How should we choose a specific θ from all of these values?

Before we try to answer this question, let us quickly note one critical point – any specific choice we make, call it $\hat{\theta}$, will depend only on x since that is the solitary piece of information we have (other than $p(x \mid \theta)$, which is a fixed function). But many different θ can generate the same x, therefore our choice $\hat{\theta}$ will not always be the actual θ that generated the data. To keep this in mind, we will call the choice $\hat{\theta}$ an *estimate* of θ and the principle for choosing it, the *estimation principle*.

You should ask three questions of any estimation problem:

1. What estimation principle can be used here?
2. How is the estimate calculated in practice? Most estimation principles involve maximizing some function, so the second question really comes down to asking – how is the maximization performed in practice?
3. How good is the estimate? Very often, certain values of the estimate are more reliable than others and it is important to know whether the estimate you calculated is any good.

Let's begin with the first question. Leaving aside Bayesian approaches, there are two common estimation principles: the maximum-likelihood and the maximum-a-posterior.

2.1. The Maximum-likelihood Principle (ML). The maximum-likelihood principle states that we should choose as an estimate of θ, a value which maximizes the probability density of the observed data x. That is,

$$\hat{\theta} = \arg\max_{\theta} p(x \mid \theta).$$

$\hat{\theta}$ is called the *maximum-likelihood estimate* (MLE) of θ.

In plain english, the maximum-likelihood principle says "choose that parameter value for which the observed data is most likely to occur."

In practice, instead of maximizing the likelihood $p(x \mid \theta)$, we often maximize $\log p(x \mid \theta)$. This gives us the same estimate $\hat{\theta}$ since log is a monotonic function. The function $\log p(x \mid \theta)$ is called the *log-likelihood* function. Because the log-likelihood function is so commonly used, many authors use special notation for it. It is denoted by $l(\theta \mid x)$. That is,

$$l(\theta \mid x) = \log p(x \mid \theta). \tag{2.1}$$

Notice the sneaky change in the argument of $l()$ to what looks like "θ given x." This is simply a reminder that we are treating the log-likelihood as a function of θ for the given observation x. It is not a conditional distribution of θ given x. I found this change of notation confusing at first but got comfortable with it in time. If you find $l(\theta \mid x)$ confusing, feel free to substitute $\log p(x \mid \theta)$ for it.

In many ML calculations you can drop any multiplicative term in $p(x \mid \theta)$ which does not contain θ since that term will not influence the location of the maximum of $l(\theta \mid x)$.

2.2. The Maximum-a-posterior Principle (MAP). MAP differs from ML in that MAP assumes that the parameter θ is also a random variable which has a prior distribution $p(\theta)$. What I mean is that θ is known to belong to some region Ω (need not be compact) in $\mathcal{R}^m$ and that we have a valid probability distribution for $\theta \in \Omega$. Given θ, the conditional density of x is still $p(x \mid \theta)$, so that the probability x is

$$p(x) = \int_{\Omega} p(x \mid \theta) p(\theta) d\theta.$$

If we observe a specific value of x in an experiment, we can evaluate the *posterior* (posterior to the observation) probability density of θ using Bayes rule:

$$p(\theta \mid x) = \frac{p(x \mid \theta) p(\theta)}{p(x)}.$$

The maximum-a-posterior principle states that we should choose as an estimate of θ, the value which maximizes the posterior probability density of θ. That is,

$$\hat{\theta} = \arg\max_{\theta} p(\theta \mid x).$$

$\hat{\theta}$ is called the *maximum-a-posterior estimate* (MAPE) of θ.

Thus MAP suggests that the appropriate choice of the parameter value is the one that is most likely, given the observation. As before, we can maximize $\log p(\theta \mid x)$ instead. Observe that

$$\log p(\theta \mid x) = \log p(x \mid \theta) + \log p(\theta) - \log p(x),$$

and that the last term on the right hand side is independent of θ. So we can drop it from maximization and instead use only $\log p(x \mid \theta) + \log p(\theta)$. For MAP we define

$$l(\theta \mid x) = \log p(x \mid \theta) + \log p(\theta) \tag{2.2}$$

and maximize $l(\theta \mid x)$ w.r.t. θ. As before, you can freely drop any multiplicative terms in $p(x \mid \theta)$ and $p(\theta)$ which do not depend on θ.

It is common practice to develop algorithms which can be used for numerically calculating the MLE as well as the MAPE. As we shall see shortly, EM is one such algorithm. To treat both cases uniformly we assume that we are maximizing the log-likelihood $l(\theta \mid x)$ where the function may be defined according to equation (2.1) or equation (2.2). I suspect that this is the reason for introducing the notation of equation (2.1).

NOTE: *This is a good time to introduce the standard notation* $\frac{\partial l(\theta|x)}{\partial \theta}$. *and* $\frac{\partial^2 l(\theta|x)}{\partial \theta^2}$. *Recall that* $\theta = (\theta_1, \cdots, \theta_m)^T$ *is a vector. The term* $\frac{\partial l(\theta|x)}{\partial \theta}$ *stands for the gradient of* $l(\theta \mid x)$ *with respect to* θ, *that is:*

$$\frac{\partial l(\theta \mid x)}{\partial \theta} = (\frac{\partial l(\theta \mid x)}{\partial \theta_1}, \cdots, \frac{\partial l(\theta \mid x)}{\partial \theta_m})^T,$$

and the term $\frac{\partial^2 l(\theta|x)}{\partial \theta^2}$ *stands for the Hessian of* $l(\theta \mid x)$ *with respect to* θ. *That is,*

$$\frac{\partial^2 l(\theta \mid x)}{\partial \theta^2} = \begin{pmatrix} \frac{\partial^2 l(\theta|x)}{\partial \theta_1^2} & \frac{\partial^2 l(\theta|x)}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 l(\theta|x)}{\partial \theta_1 \partial \theta_m} \\ \frac{\partial^2 l(\theta|x)}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 l(\theta|x)}{\partial \theta_2^2} & \cdots & \frac{\partial^2 l(\theta|x)}{\partial \theta_2 \partial \theta_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 l(\theta|x)}{\partial \theta_m \partial \theta_1} & \frac{\partial^2 l(\theta|x)}{\partial \theta_m \partial \theta_2} & \cdots & \frac{\partial^2 l(\theta|x)}{\partial \theta_m^2} \end{pmatrix}.$$

END OF NOTE.

Let us now turn to the issue of practical calculation of the estimates.

2.3. Numerical Calculation. If you are lucky, $l(\theta \mid x)$ will be differentiable and you will be able to solve

$$\frac{\partial l(\theta \mid x)}{\partial \theta} = 0 \tag{2.3}$$

in closed form for a θ that maximizes $l()$. But don't hold your breath. In most cases, maximizing $l()$ is a complicated problem that does not have a closed form solution. Some sort of numerical optimization is required. Two schemes seem to be commonly used: co-ordinate ascent and Newton-Raphson.

2.3.1. Co-ordinate ascent. We split the parameters into groups and optimize the variables in each group, one at a time. Very often the groups can be selected such that the optimization within the group is simple. The technique is called co-ordinate ascent because it is similar to maximizing a multivariable function by optimizing along each variable (co-ordinate) at a time.

Let me illustrate this with a simple example. Suppose we split the components of θ into two groups so that we write θ as $\theta = \{\Theta_1, \Theta_2\}$. Let $\Theta_1^{(k)}$ and $\Theta_2^{(k)}$ be the current estimates of the components of the optimal θ. Then, the estimates are updated according to

$$\Theta_1^{(k+1)} = \arg \max_{\Theta_1} \; l(\Theta_1, \Theta_2^{(k)} \mid x)$$
$$\Theta_2^{(k+1)} = \arg \max_{\Theta_2} \; l(\Theta_1^{(k+1)}, \Theta_2 \mid x).$$

2.3.2. Newton-Raphson. Another common technique is to solve equation (2.3) by the Newton-Raphson procedure:

$$\theta^{(k+1)} = \theta^{(k)} + \left(- \frac{\partial^2 l(\theta \mid x)}{\partial \theta^2} \mid_{\theta^{(k)}} \right)^{-1} \left(\frac{\partial l(\theta \mid x)}{\partial \theta} \mid_{\theta^{(k)}} \right).$$

Here, $\theta^{(k+1)}$ and $\theta^{(k)}$ are the values of θ in the $k+1$st and the kth iteration. Note that the iteration requires the inversion of the $m \times m$ Hessian matrix and that the iteration is not guaranteed to converge unless $l()$ is a convex function of θ. Many other alternatives are available (modified Newton, quasi Newton, conjugate gradient, etc.), but they all suffer either from sensitivity to initial conditions or from high computational cost. This is a serious problem and perhaps the best motivation for looking at EM, whose convergence properties are not as bad and whose computation is often simple.

Finally, let's see how MLE and MAPE performance is evaluated.

2.4. MLE Performance. How good is a MLE? We can measure its performance this way: Assume that the real underlying parameter θ is fixed and we repeat the experiment to get all x which are generated from $p(x \mid \theta)$. For each x, we calculate the estimate $\hat{\theta}$. Therefore, $\hat{\theta}$ will vary from experiment to experiment and will in fact be a random variable. The distribution of $\hat{\theta} - \theta$ is a measure of how well MLE does. For many cases, it turns out that $\hat{\theta} - \theta$ is approximately normally distributed with zero mean:

$$\hat{\theta} - \theta \simeq N(0, C), \tag{2.4}$$

where, C, the variance-covariance matrix is given by $C = J^{-1}(\theta)$, and

$$J(\theta) = E\left(- \frac{\partial^2 l(\theta \mid x)}{\partial \theta^2} \right) \tag{2.5}$$

is called the *expected Fisher information matrix.*

Be sure that you understand what equation (2.5) means. The term $-\frac{\partial^2 l(\theta|x)}{\partial \theta^2}$ is a matrix. It is the negative of the Hessian of $l(\theta \mid x)$ with respect to θ. The $E()$ is the expectation with respect to x. That is, $J(\theta)$ is the average value of the negative Hessian, where is the average is over different values of x generated by the fixed θ.

You might ask whether we can really calculate $J(\theta)$ since we don't know θ to begin with. Good question! Usually we calculate $J(\hat{\theta})$ instead and ignore the difference. Actually, most statisticians suggest something even more drastic. More about that in a minute.

2.4.1. MAPE Performance. The argument we used to evaluate the performance of ML assumed that θ remained fixed. We can no longer do this since MAP is based on the assumption that θ is a random variable. However, a minor variation of the ML argument suffices.

Suppose that we repeat the experiment and keep only those instances that generate a specific observation x. Clearly, the MAPE in all of these trials is the same value $\hat{\theta}$ but the actual θ that generates the x varies from experiment to experiment. Thus the distribution of $\theta - \hat{\theta}$ can be used to evaluate the performance of MAPE. Once again, the distribution turns out to be approximately normal with zero mean:

$$\theta - \hat{\theta} \simeq N(0, C),$$

but the variance-covariance matrix is now given by $C = I^{-1}(\hat{\theta})$ where, $I(\hat{\theta})$ is the *observed Fisher information matrix*,

$$I(\hat{\theta}) = -\left(\frac{\partial^2 l(\theta \mid x)}{\partial \theta^2}\right)_{\theta=\hat{\theta}}.$$

A final comment on evaluating performance: Although the appropriate performance measure for ML is J and MAP is I, usually I is used in both cases and I wil continue to do so. This is the drastic suggestion I alluded to above. The justification for it is beyond the scope of this tutorial, and if you are interested, look up Tanner's book (chapter 1) [1].

2.5. Why EM?. We have now answered the three questions that we posed at the beginning of this section – we have the appropriate estimation principles, numerical algorithms for calculating the estimates, and means for evaluating the quality of an estimate.

Why do we need anything more than this? Mainly because standard numerical techniques become very complicated in real-world problems. This was appreciated by many statisticians, and to get around these complications, they invented simple optimization algorithms for specific likelihood functions. As more hand-crafted algorithms were invented, it became clear that the algorithms were quite similar. They all relied on the observation that estimation problems contain intermediate variables, called *latent data*. When expressed in terms of the latent data, the likelihood is easy to optimize. The trick in hand-crafting an optimization algorithm is to invent a simple iteration between the parameter θ and the distribution of latent data.

It takes a lot of work to do this without assuming a specific likelihood function, and it was 1977 before Dempster, Laird and Rubin (DLR) successfully handled the general case. DLR had to make clever use of Jensen's inequality to get the iteration to work. The iteration involves calculating an expectation followed by a maximization, and DLR dubbed it the "Expectation-maximization algorithm" or the "EM algorithm."

3. The EM algorithm. The derivation of the EM algorithm has two steps: first, we express the log-likelihood in terms of the distribution of latent data. Then,

we use this expression to invent an iteration which gives the sequence of values

$$\theta^{(k)},\ k = 1, \cdots$$

which are guaranteed to increase the log-likelihood :

$$l(\theta^{(k+1)} \mid x) - l(\theta^{(k)} \mid x) \geq 0. \tag{3.1}$$

The only way I know of deriving the EM algorithm is to work through these two steps in great algebraic detail. This is unfortunate, because a more abstract derivation could give us (or at least me) more intuition. I suspect there is such a derivation, but I have not seen it yet.

At any rate, let's go on ... We will assume for the moment that we are in the ML case so that θ is a fixed (unknown) parameter.

3.1. The log-likelihood and the distribution of latent data. Recall that the observed data is the vector $x = (x_1, \cdots, x_n)^T$. Let the latent data (the intermediate variable) be $z = (z_1, \cdots, z_p)^T$.

Now, remember how conditional probabilities are related to joint probabilities. If $p(u_1, u_2)$ is the joint probability of of u_1 and u_2, then the conditional probability $p(u_1 \mid u_2)$ is given by

$$p(u_1 \mid u_2) = \frac{p(u_1, u_2)}{p(u_2)},$$

where, $p(u_2) = \int p(u_1, u_2) du_1$. Therefore, $p(u_1, u_2) = p(u_1 \mid u_2)p(u_2)$.

We'll do this for $p(z, x \mid \theta)$.

$$p(z, x \mid \theta) = p(z \mid x, \theta)p(x \mid \theta). \tag{3.2}$$

Rearranging this a bit, we get

$$p(x \mid \theta) = \frac{p(z, x \mid \theta)}{p(z \mid x, \theta)}.$$

Taking logarithms, we have

$$\log p(x \mid \theta) = \log p(z, x \mid \theta) - \log p(z \mid x, \theta). \tag{3.3}$$

Now comes the tricky part. Notice that equation (3.3) is really an identity. It holds for any value of z we plug into right hand side. Suppose we generate z's according to the distribution $p(z \mid x, \theta)$ where we have chosen a value for θ, say $\theta^{(k)}$. If we evaluate the right hand side of equation (3.3) for any z generated this way, its value will always be equal to $\log p(x \mid \theta)$. Therefore, the expected value of the right hand side of (3.3) is also equal to $\log p(x \mid \theta)$.

Let's do this mathematically. We take the expected value of the equation (3.3) with respect to $z \mid x, \theta^{(k)}$ and note that the left hand side of equation (3.3) is independent of z, so that

$$\int \log p(x \mid \theta) p(z \mid x, \theta^{(k)}) dz = \log p(x \mid \theta) \int p(z \mid x, \theta^{(k)}) dz = \log p(x \mid \theta).$$

The right hand side is

$$\int \log p(z, x \mid \theta) p(z \mid x, \theta^{(k)}) dz - \int \log p(z \mid x, \theta) p(z \mid x, \theta^{(k)}) dz,$$

so that the entire equation is

$$\log p(x \mid \theta) = \int \log p(z, x \mid \theta) p(z \mid x, \theta^{(k)}) dz - \int \log p(z \mid x, \theta) p(z \mid x, \theta^{(k)}) dz. \tag{3.4}$$

Let me introduce some notation to make it easier to write equation (3.4). Let,

$$Q(\theta, \theta^{(k)}) = \int \log p(z, x \mid \theta) p(z \mid x, \theta^{(k)}) dz, \text{ and}$$
$$H(\theta, \theta^{(k)}) = \int \log p(z \mid \theta, x) p(z \mid x, \theta^{(k)}) dz,$$

so that

$$\log p(x \mid \theta) = Q(\theta, \theta^{(k)}) - H(\theta, \theta^{(k)}). \tag{3.5}$$

This is the end of the first step.

3.2. The EM iteration. What does it take to find a value $\theta^{(k+1)}$ so that

$$\log p(x \mid \theta^{(k+1)}) \geq \log p(x \mid \theta^{(k)})?$$

Using equation (3.5) we get

$$\begin{aligned} &Q(\theta^{(k+1)}, \theta^{(k)}) - Q(\theta^{(k)}, \theta^{(k)}) \\ &- H(\theta^{(k+1)}, \theta^{(k)}) + H(\theta^{(k)}, \theta^{(k)}) \geq 0 \end{aligned} \tag{3.6}$$

Let's look at each line on the left hand side of equation (3.6) starting from the H's and working upwards towards the Q's.

The $H()$ term has a remarkable property (which we will prove below). It turns out that when $H(\theta, \theta^{(k)})$ is viewed as function of θ (for a fixed $\theta^{(k)}$), then it has a maxima at $\theta = \theta^{(k)}$. That is, for any number $\theta^{(k+1)}$, we have

$$H(\theta^{(k+1)}, \theta^{(k)}) \leq H(\theta^{(k)}, \theta^{(k)})$$

and consequently the second line on the left hand side of equation (3.6) is non-negative for any $\theta^{(k+1)}$.

Thus, we can guarantee that equation (3.6) holds if we can guarantee that

$$Q(\theta^{(k+1)}, \theta^{(k)}) - Q(\theta^{(k)}, \theta^{(k)}) \geq 0.$$

But that is easy to do ! Just choose $\theta^{(k+1)}$ as the value of θ which maximizes $Q(\theta, \theta^{(k)})$ for fixed $\theta^{(k)}$. If $\theta^{(k+1)}$ is the maximizing value, then $Q(\theta^{(k+1)}, \theta^{(k)}) \geq Q(\theta^{(k)}, \theta^{(k)})$.

Therefore, we can guarantee the inequality $\log p(x \mid \theta^{(k+1)}) \geq \log p(x \mid \theta^{(k)})$ by choosing

$$\boxed{\theta^{(k+1)} = \arg\min_\theta Q(\theta, \theta^{(k)}).} \tag{3.7}$$

Equation (3.7) is the EM-algorithm. We simply begin with some initial point $\theta^{(0)}$ and apply equation (3.7) to obtain the sequence $\theta^{(k)}$.

Before we sit back and congratulate ourselves, let's remember that we have one bit of business to clear. We have to show that $H(\theta, \theta^{(k)})$ has a maxima at $\theta = \theta^{(k)}$. To do this, we need Jensen's inequality.

3.3. Jensen's inequality. Skip this section if you are familiar with Jensen's inequality. I will review it briefly without proofs. If you want to know more, read Hardy, Littlewood and Polya's brilliant book on inequalities [3].

A function Φ is *convex* if

$$\Phi\left(\frac{x_1+x_2}{2}\right) \leq \frac{1}{2}\Phi(x_1) + \frac{1}{2}\Phi(x_2)$$

for any two numbers x_1, x_2.

Jensen's inequality states that if $x_i, i = 1, \cdots, n$ are any n numbers and $p_i, i = 1, \cdots, n$ are non-negative numbers summing up to 1 ($\sum_i p_i = 1$), then, for any convex function Φ,

$$\Phi\left(\sum_i p_i x_i\right) \leq \sum_i p_i \Phi(x_i).$$

Jensen's inequality can also be written in an integral form. If $p(x) \geq 0$ is a non-negative function having $\int p(x)dx = 1$, then for any convex function Φ,

$$\Phi\left(\int f(x)p(x)dx\right) \leq \int \Phi\left(f(x)\right)p(x)dx.$$

We will use this form of the inequality and proceed by noting that $-\log$ is a convex function.

Returning to the H functions, we have

$$\begin{aligned}
H(\theta^{(k)}, \theta^{(k)}) - H(\theta, \theta^{(k)}) &= \int \log\left(\frac{p(z \mid \theta^{(k)}, x)}{p(z \mid \theta, x)}\right) p(z \mid \theta^{(k)}, x)dz \\
&= \int -\log\left(\frac{p(z \mid \theta, x)}{p(z \mid \theta^{(k)}, x)}\right) p(z \mid \theta^{(k)}, x)dz \\
&\geq -\log \int \left(\frac{p(z \mid \theta, x)}{p(z \mid \theta^{(k)}, x)}\right) p(z \mid \theta^{(k)}, x)dz \\
&= -\log \int p(z \mid \theta, x)dz \\
&= 0,
\end{aligned}$$

where, we used Jensen's inequality in the third step.

So, we get the result that $H(\theta, \theta^{(k)})$ has a maximum at $\theta = \theta^{(k)}$.

Our derivation of the EM algorithm is now complete.

3.4. Comments on the EM iteration.

1. With the discussion of Jensen's inequality in your hand, you should see the motivation behind the tricky step of taking the expectation of both sides of equation (3.3). Taking the expectation of both sides and applying Jensen's inequality allows us to ignore the second term. This really the insight behind the EM algorithm.

2. You might well ask why the EM-iteration of equation (3.7) is simpler. On the face of it, it just looks like we replaced the usual maximization ($\max_\theta l(\theta \mid x)$ with another maximization. Although this is true, for many problems the maximization in equation (3.7) is a lot easier to do. The procedure is usually broken into two phases. First the expectation

$$Q(\theta, \theta^{(k)}) = \int p(z, x \mid \theta) p(z \mid \theta^{(k)}, x)dz$$

is calculated in closed form. And, then the maximization

$$\theta^{(k+1)} = \arg\min_{\theta} Q(\theta, \theta^{(k)})$$

is performed. The first step is called the expectation step or the E step, and the second step is called the maximization step, or the M step.

3. What if the E step and/or the M step does not have a closed form solution? If the E step is complicated, Tanner [1] recommends Monte Carlo calculation. That is,

- Draw $z_1, \cdots, z_m$ i.i.d. from the distribution $p(z \mid x, \theta^{(k)})$,
- Approximate $Q(\theta, \theta^{(k)})$ as

$$Q(\theta, \theta^{(k)}) \simeq \frac{1}{m} \sum_i \log p(\theta \mid z_i, x).$$

If the M step does not have a closed form solution, we must turn once again to standard numerical techniques. Actually, in many cases it is possible to hand-craft an algorithm to find a $\theta^{(k+1)}$ which increases (but not necessarily maximizes) the value of $Q(\theta, \theta^{(k)})$. That, is we can find $\theta^{(k+1)}$ such that

$$Q(\theta^{(k+1)}, \theta^{(k)}) > Q(\theta^{(k)}, \theta^{(k)}). \tag{3.8}$$

Since this is all that is required to generate a sequence of $\theta^{(k)}$ that increases the log-likelihood, this procedure is useful as well. Dempster, Laird and Rubin recognized this and they called it the *generalized EM algorithm* or the GEM algorithm.

4. Instead of thinking of the EM-iteration as separate E and M steps, think of it as a single step which takes $\theta^{(k)}$ as input and produces $\theta^{(k+1)}$ as an output. From this point of view, the EM iteration is a mapping M that takes any element $\theta^{(k)}$ of Ω to some other element $\theta^{(k+1)}$ of Ω. That is:

$$\begin{aligned} M &: \Omega \to \Omega \\ \theta^{(k+1)} &= M(\theta^{(k)}). \end{aligned}$$

The ML or MAP estimate is a fixed point of M. This way of looking at the EM iteration is very useful when discussing convergence of the EM algorithm. I will use this point of view below.

5. Finally, let me introduce some standard EM terminology which I have avoided so far. For the form of the EM we have developed above, the concatenation (z, x) of the observed variable x and the latent data z is called the *complete data* for the problem.
Since (z, x) is called the complete data, the observed data x is named the *incomplete data*.
I find this terminology rather strange, but it is standard and you should get used to it. I don't know how it came about. My guess is that it came from truncated data problems (where the observed data are truncated versions of the real data) that a lot of the early EM-like algorithms were designed for.

3.5. EM for MAP. The EM algorithm easily adapts for MAP estimation. The E and M steps are:
E-Step: Calculate

$$Q(\theta, \theta^{(k)}) = \int \log p(\theta \mid z, x) p(z \mid x, \theta^{(k)}) dz,$$

M-Step: The M-step uses the prior $p(\theta)$:

$$\theta^{(k+1)} = \arg\ \max_{\theta} Q(\theta, \theta^{(k)}) + \log p(\theta).$$

Notice that in the E step we use $\log p(\theta \mid z, x)$. This is possible since θ is a random variable for MAP.

4. Calculating the quality of the EM estimate. If the EM iterates converge to the ML or the MAP estimate, you can evaluate quality of the estimate by calculating the Fisher information matrix.

But what can you do if the expression for $\frac{\partial^2 l(\theta|x)}{\partial \theta^2}$ is messy? After all, we are considering cases where the Newton-Raphson is cumbersome (and the cumbersome part of Newton-Raphson is the evaluation of $\frac{\partial^2 l(\theta|x)}{\partial \theta^2}$ and its inverse). Can we calculate the information matrices in a simpler fashion, possibly by further exploiting the EM framework?

Yes. The EM framework gives alternate calculations. Whether these are simpler depends on the form of the densities involved.

Below, I give some of the popular alternatives.

4.1. The missing information principle. The most intriguing method is called "the missing information principle." The key idea here is that since it is easier to calculate Q and H functions than the log-likelihood, an expression for the information matrix in terms of these matrices might lead to a simpler expression.

Let's begin with the familiar identity of equation (3.3) which I will reproduce here:

$$\log p(x \mid \theta) = \log p(z, x \mid \theta) - \log(z \mid x, \theta).$$

From this we get

$$-\frac{\partial^2 \log p(x \mid \theta)}{\partial \theta^2} = -\frac{\partial^2 \log p(z, x \mid \theta)}{\partial \theta^2} + \frac{\partial^2 \log p(z \mid x, \theta)}{\partial \theta^2}.$$

We will proceed as we did for the derivation of the EM algorithm and take the expected value of both sides with respect to $z \mid x, \theta$. Noting that the left hand side is independent of z, we get

$$-\frac{\partial^2 \log p(x \mid \theta)}{\partial \theta^2} = \int -\frac{\partial^2 \log p(z, x \mid \theta)}{\partial \theta^2} p(z \mid x, \theta) dz + \int \frac{\partial^2 \log p(z \mid x, \theta)}{\partial \theta^2} p(z \mid x, \theta) dz$$

which can be rewritten as

$$-\frac{\partial^2 \log p(\theta \mid x)}{\partial \theta^2} = \left\{ -\frac{\partial^2 Q(\theta, \phi)}{\partial \theta^2} \right\}_{\phi=\theta} - \left\{ -\frac{\partial^2 H(\theta, \phi)}{\partial \theta^2} \right\}_{\phi=\theta} \tag{4.1}$$

This is the required expression for the information matrix in terms of Q and H. The first term on the right hand side is called "the complete information matrix" of the problem. The second term is called the "missing information matrix." The result of equation (4.1) is often paraphrased by saying the the information matrix is the complete information matrix minus the missing information matrix.

4.2. Meng and Rubin's method. Meng and Rubin's method is semi-numerical. It is suitable when closed form expressions are not easily available to apply the missing information principle or Louis' method. Actually, I should be a little more careful about what this means. Notice that when you use the missing information principle, the term

$$\left\{ \frac{-\partial^2 Q(\theta,\phi)}{\partial\theta^2} \right\}_{\phi=\theta}$$

is not a problem because you have an expression for $Q(\theta,\phi)$ (otherwise you would not have EM in the first place !). So, the H term is the potential source of the problem. Meng and Rubin's trick is to find a simple way to avoid using the H term.

Recall that we can view the EM algorithm as the mapping $\theta^{(k+1)} = M(\theta^{(k)})$. If θ^* is a fixed point of this iteration, then Meng and Rubin showed that the various information matrices were related at θ^* by

$$\left\{ \frac{\partial M(\theta)}{\partial\theta} \right\}_{\theta=\theta^*} \left\{ \frac{\partial^2 Q(\theta,\theta^*)}{\partial\theta^2} \right\}_{\theta=\theta^*} = \frac{\partial^2 H(\theta,\theta^*)}{\partial\theta^2} \mid_{\theta=\theta^*} .$$

We can substitute this in the missing information principle to get

$$\begin{aligned} -\frac{\partial^2 \log(\theta \mid x)}{\partial\theta^2} &= \left\{ -\frac{\partial^2 Q(\theta,\theta^*)}{\partial\theta^2} \right\}_{\phi=\theta} - \left\{ -\frac{\partial^2 H(\theta,\theta^*)}{\partial\theta^2} \right\}_{\phi=\theta} \\ &= \left(I - \frac{\partial M(\theta)}{\partial\theta} \mid_{\theta=\theta^*} \right) \left(\frac{-\partial^2 Q(\theta,\theta^*)}{\partial\theta^2} \right)_{\theta=\theta^*} . \end{aligned} \tag{4.2}$$

All that remains now is to approximate $\frac{\partial M(\theta)}{\partial\theta}$ numerically.

Note that M is really a vector valued function (since it produces $\theta^{(k+1)}$ which is a vector). Let $M_j(\theta)$ be the j^{th} component of M. Evaluating $\frac{\partial M(\theta)}{\partial\theta}$ is the same as calculating the numbers $r_{i,j}$,

$$r_{i,j} = \frac{\partial M_j(\theta)}{\partial\theta_i},$$

where, θ_i is the i^{th} component of θ. If we can numerically approximate $r_{i,j}$, then we can stack all $r_{i,j}$'s in a matrix to get a numerical approximation to $\frac{\partial M}{\partial\theta}$.

Now suppose that $\theta^* = (\theta_1^*, \cdots \theta_m^*)$ is a fixed point of M, and that $\tilde{\theta}^i$ is the vector obtained by perturbing the i^{th} component of θ^*, i.e.

$$\tilde{\theta}^i = \left(\theta_1^*, \cdots, \theta_{i-1}^*, \tilde{\theta}_i, \theta_{i+1}^*, \cdots, \theta_m^* \right),$$

where, $\tilde{\theta}_i \neq \theta_i^*$. Then,

$$\begin{aligned} r_{i,j} &= \frac{\partial M_j(\theta)}{\partial\theta_i} \\ &\simeq \frac{M_j(\tilde{\theta}^i) - M_j(\theta^*)}{\tilde{\theta}_i - \theta_i^*} \\ &= \frac{M_j(\tilde{\theta}^i) - \theta_j^*}{\tilde{\theta}_i^i - \theta_i^*}, \end{aligned} \tag{4.3}$$

where, I have used fact that θ^* is a fixed point of M.

Meng and Rubin's algorithm works like this:

1. Continue EM till it converges. (This is EM).
2. For each i, j calculate $r_{i,j}$ numerically by using the perturbation described above and equation (4.3)). Don't loose sight of the fact that the calculation of $r_{i,j}$ according to equation (4.3) requires one EM iteration for the numerator.
3. Arrange $r_{i,j}$ in a matrix, called $\frac{\partial M}{\partial \theta}$ such that $r_{i,j}$ is the i, j^{th} entry in the matrix. Substitute this matrix in equation (4.2) to get a numerical expression for the information matrix.

Meng and Rubin warn that the variance-covariance matrix you get from this procedure will not necessarily be symmetric (Can you see why? Hint: the numerical derivative calculated in equation (4.3) is not a symmetric difference). They recommend that if this technique gives V as the variance-covariance matrix, then you should use $1/2(V + V^T)$ as the symmetric variance-covariance matrix.

5. Convergence of the EM algorithm. I will now discuss convergence properties of the EM algorithm. Proofs of these properties are complicated and I will skip them.

Convergence of the EM algorithm requires the log-likelihood function $l(\theta \mid x)$ and the $Q()$ function to be "well behaved." The requirements for good behavior are pretty weak and are satisfied in most situations. Beware though, there are cases of practical importance when these conditions are violated and the EM algorithm does strange things. More about this later.

The idea of "convergence" of the EM algorithm needs some discussion. Since the EM produces a sequence of values $\theta^{(k)}$ such that $l(\theta^{(k+1)} \mid x) \geq l(\theta^{(k)} \mid x)$ we can talk either about the convergence of l or about the convergence of θ. The first does not imply the second since $l()$ may be multi-modal causing $\theta^{(k)}$ to bounce around while $l(\theta^{(k)} \mid x)$ converges to some value. So we would like conditions under which both converge.

Finally, note that the EM does not guarantee that $l(\theta^{(k+1)} \mid x)$ will be strictly greater than $l(\theta^{(k)} \mid x)$. So that it is quite likely that an EM sequence may get trapped at a stationary point rather than a maximal point. In general, that is the best that can be done – the theorems only guarantee that $l()$ will converge to a stationary point.

5.0.1. Conditions on the log-likelihood. As I said above, we need conditions on $l()$ and $Q()$ to guarantee convergence. Here are the conditions on $l()$:

1. The first condition is just the restatement of the fact that the set of feasible parameters Ω is contained in a finite dimensional Euclidean space:

$$\Omega \in \mathcal{R}^m.$$

2. Loosely speaking, the second condition requires that the set of parameters for which $l()$ is high to be "small." More precisely, let Ω_θ be the set

$$\Omega_{\theta_0} = \{\alpha \in \Omega : l(\alpha \mid x) \geq l(\theta_0 \mid x)\}$$

for $l(\theta_0 \mid x) > -\infty$. The second condition requires the set Ω_{θ_0} to be compact for any θ_0.

This simply means that the only way to increase the value of the log-likelihood function is to stay within the set Ω_{θ_0}. You should easily guess now that the compactness of Ω_{θ_0} will give us convergence.

NOTE *If you are not familiar with the notion of a compact set, you can use the following result - all compact sets in $\mathcal{R}^m$ are closed and bounded. Thus, this condition requires Ω_{θ_0} to*

be closed and bounded. Therefore, any Cauchy sequence in Ω_{θ_0} will converge to some point in Ω_{θ_0}. END OF NOTE

3. Finally, we require ordinary continuity and differentiability conditions to talk meaningfully about a singular point. The third condition requires the log-likelihood $l(\theta \mid x)$ to be continuous in Ω and differentiable in the interior of Ω.

NOTE *Since $l(\theta \mid x)$ is differentiable, $\frac{\partial l}{\partial \theta}$ exists. Any point in the interior of Ω at which $\frac{\partial l}{\partial \theta} = 0$ is called a singular point of $l()$. A singular point is either a local maxima, a local minima, or a saddle point. The value of $l()$ at the singular point is called a singular value of $l()$. Singular points and values are sometimes called stationary points and values.* END OF NOTE

These condition should not come as a surprise to you. The proof of convergence for almost any problem requires conditions analogous to these.

Of the three conditions, the compactness of Ω_{θ_0} can be problematic at times. MacLachlan and Krishnan [2] show that this condition is violated in a common application of EM - estimation of component means, variances, and mixture coefficients in a mixture-of-normals model. Consequently, the EM algorithm does not have good convergence (in theory and in practice) and artificial constraints are required to make it behave well.

5.0.2. Condition on Q. We need one condition on $Q(.,.)$ which is rather mild: $Q(.,.)$ must be a continuous function of both its arguments.

With these conditions in place, we are ready for the main convergence theorem:

THEOREM: If the log-likelihood and $Q(.,.)$ satisfy conditions stated above, then the limit points of any sequence $\theta^{(k)}$ generated by the EM algorithm are singular points of $l(\theta \mid x)$. Further, the convergence of $l(\theta^{(k)} \mid x)$ is monotonic to its singular value.

This theorem was first proven by Wu, and the conditions on $l()$ and $Q()$ used in the theorem are sometimes called Wu's regularity conditions.

Don't miss the part of the theorem which says that *any* sequence generated by the EM algorithm converges to some singular point. So you can pick any reasonable initial starting point to enjoy the benefits of EM.

The monotonic convergence of $l()$ should not come as a surprise. This is just the restatement of condition $l(\theta^{(k+1)} \mid x) \geq l(\theta^{(k)} \mid x)$.

Finally, a warning – the theorem really applies to the EM algorithm. I am not terribly sure what happens to the sequence produced by GEM. I will try to find out more.

Do not be tempted to ignore these convergence conditions. There are well known cases where the conditions are violated the EM converges to a minimum ! The GEM is also known to behave badly, for example the $l(\theta^{(k)} \mid x)$ sequence can converge without the $\theta^{(k)}$ sequence converging.

If you find that your EM algorithm is converging to something strange, check to see if any of the above conditions are violated (well, check you code for bugs first).

5.1. The Rate of Convergence. The rate of convergence of the EM algorithm is linear. In the neighborhood of the fixed point θ^* of M, the rate is given by

$$\left\{\frac{\partial^2 H(\theta,\theta^*)}{\partial\theta^2}\right\}_{\theta=\theta^*}\left\{\frac{\partial^2 Q(\theta,\theta^*)}{\partial\theta^2}\right\}^{-1}_{\theta=\theta^*}.$$

This is perhaps the biggest problem with the EM algorithm. Once the EM iterates get close to the fixed point, the convergence is rather slow. Many alternates have been suggested to speed up the convergence, but none of these have the simplicity of the EM iteration. You should refer to Tanner [1] and MacLachlan and Krishnan [2] for these options.

6. Where next?. I have finished what I started to do. If you have read through all of this, you should have a good idea of the theory of the EM algorithm. Read Part II of this tutorial for applications of EM.

I also encourage you to read the proofs of the many results that I quoted.

Oh yes, if you find typos, mistakes, bugs etc. in this, do let me know.

REFERENCES

[1] Martin A. Tanner, *Tools for Statistical Inference*, 2nd ed., Springer Series in Statistics, Springer-Verlag, 1993.

[2] Geoffrey J. MacLachlan, and Thriyambakam Krishnan, *The EM Algorithm and Extensions*, Wiley Series in Probability and Statistics, John Wiley and Sons, 1997.

[3] G. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*, 2nd ed., Cambridge University Press, 1952.