Everything you always wanted to know about the EM algorithm but . . . . .

Anand Rangarajan

September 17, 1998

1 Introduction

The EM algorithm has achieved considerable notoriety in the last decade especially in the computer vision, neural networks and medical imaging research communities. We would like to dispel some widely held myths about the EM algorithm while retaining some of its better qualities. In particular, we would like to show that there is no guarantee that

1. the EM algorithm is fast, or that
2. the EM algorithm can automagically escape local minima.

Some of the positive qualities of the EM algorithm which go a long way toward explaining its wide acceptance are

1. very few free parameters to tune unlike gradient descent and conjugate gradient algorithms which require line search parameters to be set, and
2. automagical constraint satisfaction of certain constraints like positivity and specific linear constraints.

We have chosen mixture models as applied to point feature registration to showcase the EM algorithm. In general, mixture models are a good place to use the EM algorithm.

2 Mixture models for point feature registration

In Figure 1, we show an example of the point feature registration problem. Two point sets are depicted with the point set on the left having many more points than the point set on the right. While this example is somewhat artificial, it has been chosen to illustrate the kind of problem that can be modeled using mixture densities. The motivation behind choosing this particular example

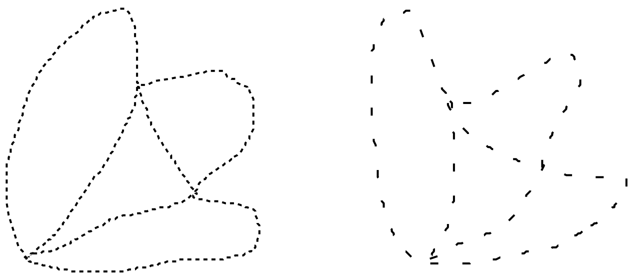

Figure 1: Point feature registration problem. Left: Finely sampled point set. Right: Coarsely sampled point set which can act as a set of cluster centers for the point set on the left.

is as follows: the point set on the right of Figure 1 can serve as a set of point cluster centers for the more finely sampled point set on the left. Imagine that a 2D affine spatial mapping can bring the point set on the left into a reasonably good registration with the point set on the right. (While this can be disputed, it is not really germane to our primary focus here, namely, the EM algorithm.) To obtain a good registration, we would have to solve for the affine mapping as well as the correspondences between the point features on the left and those on the right. Since the number of points in the left point set far outnumber the number of points in the right point set, we can think of correspondence as really a clustering problem where we have to determine the cluster memberships of every point in the left point set to a "cluster center" point in the right point set. In this way, a correspondence problem can be converted into a clustering problem.

Mixture models are convenient frameworks for couching clustering problems. In particular, Gaussian mixtures are very popular since they allow one to have a cake and eat it too. Familiar Gaussian distribution techniques can still be used in Gaussian mixtures while the fallacy of using Gaussian distributions to model multi-modal distributions is neatly sidestepped. Essentially, the basic idea behind Gaussian mixtures is the assumption that the data are generated from a fixed number of classes K with $\{\pi_a, a \in \{1, \ldots, K\}\}$being the *a priori* class probabilities. In the case of point feature matching, let $\{X_i, i \in \{1, \ldots, N\}\}$ denote the oversampled point set (the left point set in Figure 1), $\{Y_a, a \in \{1, \ldots, K\}\}$ denote the cluster center point set (the point set on the right in Figure 1) and (A, t) denote the affine spatial mapping. The mixture density of X is written as:

$$p(X|Y, A, t, C, \pi) = \prod_{i=1}^{N} \sum_{a=1}^{K} \pi_a p(X_i|Y_a, A, t, C_a) \tag{1}$$

where

$$p(X_i|Y_a, A, t, C_a) = \frac{1}{2\pi|C_a|^{\frac{1}{2}}} \exp\{-(X_i - AY_a - t)^T C_a^{-1}(X_i - AY_a - t)\}$$

and $\pi_a > 0, \forall a \in \{1, \ldots, K\}$ and $\sum_{a=1}^{K} \pi_a = 1$. Each point i in point set X is denoted as X_i and is parameterized as $(x_i^{(1)}, x_i^{(2)})^T$ with a similar parameterization for cluster center points in Y. C_a is the 2×2 covariance matrix corresponding to the a^{th} cluster center point.

Point feature matching (within the mixture density context) is now recast as a maximum likelihood (ML) problem. The registration parameters (A, t) as well as the unknown mixture parameters C and π are estimated by minimizing the following negative log-likelihood energy function:

$$(A^*, t^*, C^*, \pi^*) = \arg \min_{A,t,C,\pi} - \log p(X|Y, A, t, C, \pi) \tag{2}$$

subject to

$$\sum_{a=1}^{K} \pi_a = 1, \; \pi_a > 0, \forall a \in \{1, \ldots, K\}. \tag{3}$$

The motivation behind explicitly introducing a covariance matrix C in the point feature matching mixture model above is in being able to model the degree to which the over sampled points in X will not lie on top of their corresponding cluster centers in Y even after a "perfect" registration. The covariance matrix C is able to introduce directions in $2D$ in which some of the points in X will depart from their chosen cluster centers in Y.

3 The EM Algorithm for mixture densities

Minimizing the negative log likelihood energy function in (2) is not straightforward. For instance, we could run gradient descent on all the variables A, t, C, π while enforcing the constraints in (3) via gradient projection methods. The strength of the EM algorithm lies in its ability to provide an extremely simple sequence of iterates that are guaranteed to find a local minimum of (2). Gradient projection methods, line searches etc. are completely unnecessary.

At the heart of all EM algorithms is an *incomplete/complete* data parameterization. In the case of the mixture model, note that the log likelihood in (2) does not specify which cluster center point a in Y is most closely associated with a given point i in X. If such a set of cluster memberships were known, the task of estimating the spatial mapping parameters would be trivial: use the cluster memberships to associate each point i in X with a point a in Y and solve a simple least-squares energy function to get (A, t). Denote the set of *hidden* cluster memberships by $\{M_{ai}, \; a \in \{1, , , , K\}, i \in \{1, , , , N\}\}$. Now, after summoning all your esthetic and artistic sensibilities, write down the complete data mixture likelihood:

$$p(M, X|Y, A, t, C, \pi) = \prod_{i=1}^{N} \prod_{a=1}^{K} [\pi_a p(X_i|Y_a, A, t, C_a)]^{M_{ai}} \tag{4}$$

where

$$\sum_{a=1}^{K} M_{ai} = 1, \; \forall i \in \{1, \ldots, N\}, \text{ and } M_{ai} \in 0, 1.$$

From the constraints on M, we can show that

$$\sum_{\{M\}} p(M, X|Y, A, t, C, \pi) = p(X|Y, A, t, C, \pi). \tag{5}$$

To see how (5) obtains, consider the following simple example. Let $K = 2$ and $N = 1$ in (5). Then, we need to show that

$$\sum_{\{M\}} [\pi_1 p(X_1|Y_1, A, t, C_1)]^{M_{11}} [\pi_2 p(X_1|Y_2, A, t, C_2)]^{M_{21}} = \pi_1 p(X_1|Y_1, A, t, C_1) + \pi_2 p(X_1|Y_2, A, t, C_2). \tag{6}$$

The constraints on M for this simple example are $M_{11} + M_{21} = 1$, and $\{M_{a1} \in \{0, 1\}, \forall a \in \{1, 2\}\}$. There are only two possible configurations of M that support these constraints, $\{M_{11}, M_{21}\} = \{1, 0\}$ and $\{M_{11}, M_{21}\} = \{0, 1\}$. When we sum over the left of (6), we are really summing over these two configurations. It should be obvious that the right side of (6) obtains. The general case is a straightforward extension of this example. For a general unspecified K clusters, the number of configurations supporting the constraints is K, just in the above case of $K = 2$. The configuration space summation is also independent of the i index. Equation 5 shows that we get the original incomplete date likelihood in (1) by summing over all the configurations of M in the complete data likelihood in (4). This completes the first arc of the EM algorithm, namely, the specification of a complete but hidden data variable which when summed over (or integrated out) in a complete data likelihood yields the original incomplete data likelihood.

We begin the second arc of the EM algorithm with Bayes' theorem:

$$p(X|Y, A, t, C, \pi) = \frac{p(M, X|Y, A, t, C, \pi)}{p(M|X, Y, A, t, C, \pi)}. \tag{7}$$

From (4) and (7), we get

$$p(M|X, Y, A, t, C, \pi) = \prod_{i=1}^{N} \prod_{a=1}^{K} [\frac{\pi_a p(X_i|Y_a, A, t, C_a)}{\sum_{b=1}^{K} \pi_b p(X_i|Y_b, A, t, C_b)}]^{M_{ai}}. \tag{8}$$

To obtain (8), we have used the constraints on M. Recall from (2) that our main goal is to maximize the left side of (7). Rewriting (7) as a log-likelihood, we get

$$-\log p(X|Y, A, t, C, \pi) = -\log p(M, X|Y, A, t, C, \pi) + \log p(M|X, Y, A, t, C, \pi). \tag{9}$$

The basic idea in EM is not to minimize the left side of (9) but instead to minimize a quantity related to the expression on the right. The reason why we cannot just minimize the expression on the right is because we do not know M. (M is a hidden data variable.)

The EM trick:

Assume a series of updates for the variables of interest (A, t, C, π). Imagine we are at step n and that we currently have $(A^{(n)}, t^{(n)}, C^{(n)}, \pi^{(n)})$. The EM trick consists of taking conditional expectations of both sides of (9) w.r.t. $p(M|X, Y, A^{(n)}, t^{(n)}, C^{(n)}, \pi^{(n)})$. This is called the *E-step*. The left hand side of (9) after we take conditional expectations is

$$-<\log p(X|Y, A, t, C, \pi)>_{X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}} = \\ -\sum_{\{M\}} p(M|X, Y, A^{(n)}, t^{(n)}, C^{(n)}, \pi^{(n)}) \log p(X|Y, A, t, C, \pi).$$

Since M does not appear in the left side of (9), we get

$$-<\log p(X|Y, A, t, C, \pi)>_{X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}} = -\log p(X|Y, A, t, C, \pi)$$

since $\sum_{\{M\}} p(M|X, Y, A^{(n)}, t^{(n)}, C^{(n)}, \pi^{(n)}) = 1$ by definition. The conditional expectation has left the negative log likelihood unchanged. The right hand side of (9) has two terms: The first term after we take conditional expectations is

$$<-\log p(M, X|Y, A, t, C, \pi)>_{X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}} = \\ -\sum_{\{M\}} p(M|X, Y, A^{(n)}, t^{(n)}, C^{(n)}, \pi^{(n)}) \log p(M, X|Y, A, t, C, \pi). \tag{10}$$

The second term after we take conditional expectations is

$$<\log p(M|X, Y, A, t, C, \pi)>_{X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}} = \\ \sum_{\{M\}} p(M|X, Y, A^{(n)}, t^{(n)}, C^{(n)}, \pi^{(n)}) \log p(M|X, Y, A, t, C, \pi). \tag{11}$$

Let's dispense with the second term once and for all. The EM algorithm is based on minimizing a quantity related to the right hand side of (9) instead of the negative log-likelihood on the left hand side. There are two terms on the right hand side of (9). Set aside the first term for the moment. The second term is (11) above. Do not get confused by the presence of both $(A^{(n)}, t^{(n)}, C^{(n)}, \pi^{(n)})$ and the unknown (A, t, C, π). We are at step n in a sequence of updates and the variables in the second term are (A, t, C, π). The second term can be safely ignored because of the Kullback-Leibler (KL) distance measure. The KL distance is a non-negative distance between two probability distributions. Without deriving the distance, we baldly state that

$$\sum_{\{M\}} p(M|X, Y, A^{(n)}, t^{(n)}, C^{(n)}, \pi^{(n)}) \log \frac{p(M|X, Y, A^{(n)}, t^{(n)}, C^{(n)}, \pi^{(n)})}{p(M|X, Y, A, t, C, \pi)} \geq 0 \tag{12}$$

with the equality sign holding if and only if $p(M|X,Y,A,t,C,\pi) = p(M|X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)})$ (Cover and Thomas, 1991). Continue to focus on the second term above. Since we are at step n, the initial condition for the second term is

$$E^{(n)}_{\mathrm{secondterm}} = \sum_{\{M\}} p(M|X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}) \log\, p(M|X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}). \tag{13}$$

At the $(n+1)\mathrm{th}$ step, we would have $(A^{(n+1)},t^{(n+1)},C^{(n+1)},\pi^{(n+1)})$. The corresponding energy of the second term at the $(n+1)\mathrm{th}$ step is

$$E^{(n+1)}_{\mathrm{secondterm}} = \sum_{\{M\}} p(M|X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}) \log\, p(M|X,Y,A^{(n+1)},t^{(n+1)},C^{(n+1)},\pi^{(n+1)}). \tag{14}$$

The change in energy for the second term is

$$E^{(n)}_{\mathrm{secondterm}} - E^{(n+1)}_{\mathrm{secondterm}} = \sum_{\{M\}} p(M|X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}) \log \frac{p(M|X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)})}{p(M|X,Y,A^{(n+1)},t^{(n+1)},C^{(n+1)},\pi^{(n+1)})}.$$

But the change in energy is just a special case of the KL distance in (12) with the $(n+1)\mathrm{th}$ step $(A^{(n+1)},t^{(n+1)},C^{(n+1)},\pi^{(n+1)})$ inserted in place of the unspecified (A,t,C,π). *Therefore the change in energy of the second term from step n to step $n+1$ is guaranteed to be greater than zero for any update $(A^{(n+1)},t^{(n+1)},C^{(n+1)},\pi^{(n+1)})$ as long as $(A^{(n+1)},t^{(n+1)},C^{(n+1)},\pi^{(n+1)}) \neq (A^{(n)},t^{(n)},C^{(n)},\pi^{(n)})$ by virtue of the energy difference being a Kullback-Leibler distance measure.* This kind of argument is present in all EM algorithms. We can safely dispense with the energy term in (11). Therefore, minimizing the negative log-likelihood $-\log\, p(X|Y,A,t,C,\pi)$ is the same as minimizing the first term (10). At the end of the *E-step*, we have

$$\min_{A,t,C,\pi} -\log\, p(X|Y,A,t,C,\pi) = \min_{A,t,C,\pi} -\sum_{\{M\}} p(M|X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}) \log\, p(M,X|Y,A,t,C,\pi). \tag{15}$$

This is not as horrendous as it seems (especially for the mixture model). From (4), we know that $p(M,X|Y,A,t,C,\pi) = \prod_{i=1}^{N}\prod_{a=1}^{K}[\pi_a p(X_i|Y_a,A,t,C_a)]^{M_{ai}}$. Therefore the corresponding negative log- likelihood $-\log\, p(M,X|Y,A,t,C,\pi) = -\sum_{a=1}^{K}\sum_{i=1}^{N} M_{ai}\log\,[\pi_a p(X_i|Y_a,A,t,C_a)]$. The conditional expectation in (15) reduces to

$$\begin{aligned} -\sum_{\{M\}} p(M|X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}) \log\, p(M,X|Y,A,t,C,\pi) = \\ -\sum_{ai} < M_{ai} >_{X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}} \log\,[\pi_a p(X_i|Y_a,A,t,C_a)] \end{aligned} \tag{16}$$

The conditional expectation $< M_{ai} >_{X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}}$ can be easily evaluated from the density function in (8). Since the density function is basically a simple extension to a Bernoulli probability, we state without proof that

$$< M_{ai} >_{X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}} = \frac{\pi_a^{(n)} p(X_i | Y_a, A^{(n)}, t^{(n)}, C_a^{(n)})}{\sum_{b=1}^{K} \pi_b^{(n)} p(X_i | Y_b, A^{(n)}, t^{(n)}, C_b^{(n)})}. \tag{17}$$

With the update for $< M_{ai} >$ in place we can turn to the updates of the rest of the variables. This is pretty straightforward. All we have to do is differentiate the right side of (16) w.r.t. each variable and solve. We get

$$\begin{aligned} \pi_a^{(n+1)} &= \frac{\sum_{i=1}^{N} < M_{ai} >_{X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}}}{N}, \\ C_a^{(n+1)} &= \sum_{i=1}^{N} < M_{ai} >_{X,Y,A^{(n)},t^{(n)},C^{(n)},\pi^{(n)}} (X_i - A^{(n)} Y_a - t^{(n)})(X_i - A^{(n)} y_a - t^{(n)})^T \end{aligned} \tag{18}$$

followed by a straightforward least-squares update for the spatial mapping parameters (A, t). The least-squares update is obtained by differentiating (16) w.r.t. (A, t) and solving for them. The EM algorithm then consists of a pair of updates. First the complete date is updated in (17) followed by the update of the remaining parameters in (18) [including the least-squares update for the spatial mapping parameters (A, t).]

4 Discussion

The EM algorithm is clearly an elegant algorithm—especially for mixture densities. As we have seen, there are no step-size parameters to be set or gradient projections to be evaluated. The algorithm is guaranteed to find a local minimum by virtue of following a descent direction of the original mixture negative log-likelihood. We began with a complex looking mixture log-likelihood (2) and we ended with a pair of updates of the complete data (actually conditional expectation of the complete data) followed by updates of the remaining parameters [(17) and (18)]. The algorithm has the flavor of grouped coordinate-descent and indeed it is just that. [This was first shown by Hathaway (Hathaway, 1986).] The caveat is that there is no guarantee that grouped coordinate-descent is a fast algorithm. It turns out that the EM algorithm has some of the properties of a good Quasi-Newton method when applied to mixtures (Redner and Walker, 1984). However, in other applications—for instance tomographic reconstruction—the EM algorithm is very slow and inefficient (Mumcuoglu et al., 1994). Finally, EM algorithms have to be derived from first principles for new log-likelihoods and there is no guarantee that we will always obtain a simple algorithm (as simple as the case in mixtures).

References

Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. John Wiley and Sons, New York, NY.

Hathaway, R. (1986). Another interpretation of the EM algorithm for mixture distributions. *Statistics and Probability Letters*, 4:53–56.

Mumcuoglu, E., Leahy, R., Cherry, S., and Zhou, Z. (1994). Fast gradient-based methods for Bayesian reconstruction of transmission and emission PET images. *IEEE Trans. Med. Imag.*, 13(4):687–701.

Redner, R. A. and Walker, H. F. (1984). Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review*, 26(2):195–239.