Expectation-Maximization as lower bound maximization

Thomas P. Minka

November 4, 1998

Abstract

The Expectation-Maximization algorithm given by Dempster et al (1977) has enjoyed considerable popularity for solving MAP estimation problems. This note derives EM from the lower bounding viewpoint (Luttrell, 1994), which better illustrates the convergence properties of the algorithm and its variants. The algorithm is illustrated with two examples: pooling data from multiple noisy sources and fitting a mixture density.

1 Introduction

The Expectation-Maximization (EM) algorithm is an iterative optimization technique specifically designed for probabilistic models. It uses a different strategy than gradient descent or Newton's method and sometimes provides faster convergence. However, it is still a local technique, and so is just as susceptible to local minima.

The difference between EM and gradient descent is illustrated in figure 1. Starting from the current guess, gradient descent makes a linear approximation to the objective function, then takes some step uphill. Unfortunately, we don't know in advance how good the linear approximation is and consequently how big of a step we can take. Newton's method replaces the line with a quadratic but suffers from the same difficulty.

EM instead makes a local approximation that is a lower bound to the objective function. This is called the *primal-dual* method (Bazaraa and Shetty, 1979). The lower bound can have any functional form, in principle. Choosing the new guess to maximize the lower bound will always be an improvement over the previous guess, unless the gradient was zero there. So the idea is to alternate between computing a lower bound (the "E-step") and maximizing the bound (the "M-step"), until a point of zero gradient is reached.

The bound used by EM is the following form of Jensen's inequality:

$$\begin{aligned} \sum_j g(j) a_j &\geq \prod_j g(j)^{a_j} \qquad (1) \\ \text{provided } \sum_j a_j &= 1 \\ a_j &\geq 0 \\ g(j) &\geq 0 \end{aligned}$$

That is, an arithmetic mean is never smaller than a geometric mean.

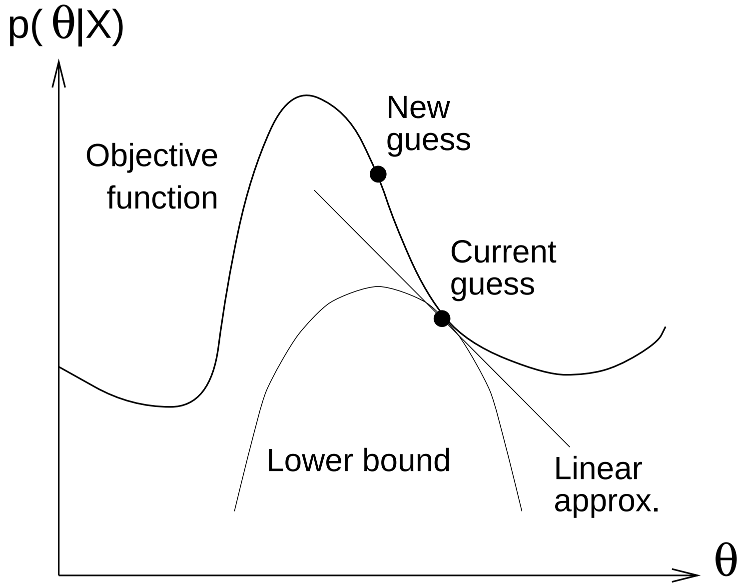

Figure 1: Maximizing a function with lower-bound approximation vs. linear approximation.

2 The General EM Algorithm

Maximum A-Posteriori (MAP) estimation concerns the maximization of the function

$$f(\theta) = p(\mathbf{X}, \theta) \tag{2}$$

where $\mathbf{X}$ is the matrix of observed data. If $f(\theta)$ is a simple function, then its maximum can often be found analytically, e.g. by equating its gradient to zero. However, it often happens that $f(\theta)$ has the form of a *mixture* of simple functions:

$$f(\theta) = p(\mathbf{X}, \theta) = \int_{\mathbf{h}} p(\mathbf{X}, \mathbf{h}, \theta) \tag{3}$$

This is the situation which EM addresses. One case where this situation arises is when part of the data set is missing and we must integrate over the possible values for the missing data. This happens when using a Gaussian mixture model (section 4). Another case is when the data model has nuisance parameters that we don't know the value of and are not interested in estimating. This happens when pooling data from sources with unknown noise variance (section 3).

Given a guess for θ, the idea is to lower-bound $f(\theta)$ with a function $g(\theta, q(\mathbf{h}))$, parameterized by the free variables $q(\mathbf{h})$ (there is one free parameter for every value of $\mathbf{h}$):

$$\begin{aligned} f(\theta) = \int_{\mathbf{h}} \frac{p(\mathbf{X}, \mathbf{h}, \theta)}{q(\mathbf{h})} q(\mathbf{h}) \quad &\geq \quad g(\theta, q(\mathbf{h})) = \prod_{\mathbf{h}} \left(\frac{p(\mathbf{X}, \mathbf{h}, \theta)}{q(\mathbf{h})} \right)^{q(\mathbf{h})} \\ \text{provided} \quad \int_{\mathbf{h}} q(\mathbf{h}) \quad &= \quad 1 \end{aligned} \tag{4}$$

The provision implies that $q(\mathbf{h})$ is a valid probability distribution over $\mathbf{h}$. Define G to be the logarithm of the bound (Neal and Hinton, 1993):

$$G(\theta, q) = \log g(\theta, q) = \int_{\mathbf{h}} q(\mathbf{h}) \log p(\mathbf{X}, \mathbf{h}, \theta) - q(\mathbf{h}) \log q(\mathbf{h}) \tag{5}$$

The inequality (4) is true for any q, but we also want the lower bound to touch f at the current guess for θ. So we choose q to maximize $G(\theta, q)$. This raises the lower bound in figure 1 to touch the objective. Adding a Lagrange multiplier for the constraint on q gives:

$$\begin{aligned} G(\theta, q) &= \lambda(1 - \int_{\mathbf{h}} q(\mathbf{h})) + \int_{\mathbf{h}} q(\mathbf{h}) \log p(\mathbf{X}, \mathbf{h}, \theta) - q(\mathbf{h}) \log q(\mathbf{h}) && (6) \\ \frac{\mathrm{d}G}{\mathrm{d}q(\mathbf{h})} &= -\lambda - 1 + \log p(\mathbf{X}, \mathbf{h}, \theta) - \log q(\mathbf{h}) = 0 && (7) \\ q(\mathbf{h}) &= \frac{p(\mathbf{X}, \mathbf{h}, \theta)}{\int_{\mathbf{h}} p(\mathbf{X}, \mathbf{h}, \theta)} = p(\mathbf{h}|\mathbf{X}, \theta) && (8) \end{aligned}$$

For this choice of q, the bound becomes

$$g(\theta, q) = \prod_{\mathbf{h}} \left(\frac{p(\mathbf{X}, \mathbf{h}, \theta)}{p(\mathbf{h}|\mathbf{X}, \theta)} \right)^{q(\mathbf{h})} = \prod_{\mathbf{h}} (p(\mathbf{X}, \theta))^{q(\mathbf{h})} = p(\mathbf{X}, \theta)^{(\int_{\mathbf{h}} q(\mathbf{h}))} = p(\mathbf{X}, \theta) \tag{9}$$

so indeed it touches the objective $f(\theta)$ at the current guess for θ. Another way to see this result is to rewrite $G(\theta, q)$ as

$$\begin{aligned} G(\theta, q) &= E_{q(\mathbf{h})} \left[\log \frac{p(\mathbf{X}, \mathbf{h}, \theta)}{q(\mathbf{h})} \right] && (10) \\ &= -E_{q(\mathbf{h})} \left[\log \frac{q(\mathbf{h})}{p(\mathbf{h}|\mathbf{X}, \theta)} \right] + \log p(\mathbf{X}, \theta) && (11) \\ &= -D(q(\mathbf{h}) \,||\, p(\mathbf{h}|\mathbf{X}, \theta)) + \log p(\mathbf{X}, \theta) && (12) \end{aligned}$$

assuming $q(\mathbf{h})$ is a valid probability distribution. The relative entropy $D(q||p)$ is a measure of distance between distributions q and p. Therefore $F(\theta, q)$ is maximized over q when this distance is zero, i.e. $q(\mathbf{h}) = p(\mathbf{h}|\mathbf{X}, \theta)$ (8), at which point $G(\theta, q) = \log p(\mathbf{X}, \theta)$. This interpretation is from Buntine (1996).

Finding q to get a good bound is the "E-step" of the algorithm. To get the next guess for θ, we maximize the bound over θ (this is the "M-step"). This step is problem-dependent. The relevant term of G is

$$\int_{\mathbf{h}} q(\mathbf{h}) \log p(\mathbf{X}, \mathbf{h}, \theta) = E_{q(\mathbf{h})} [\log p(\mathbf{X}, \mathbf{h}, \theta)] \tag{13}$$

This may be difficult to do exactly; fortunately, it isn't strictly necessary to maximize the bound over θ. As we can see from figure 1, any improvement of $G(\theta, q)$ along θ will do. This is sometimes called "generalized EM." Not all such algorithms are meaningful, however. From the figure, it is clear that the derivative of g at the current guess is identical to the derivative of f. This can also be shown formally:

$$\frac{d\log f}{d\theta} = \frac{\int_{\mathbf{h}} \frac{dp(\mathbf{X},\mathbf{h},\theta)}{d\theta}}{\int_{\mathbf{h}} p(\mathbf{X},\mathbf{h},\theta)} = \frac{\int_{\mathbf{h}} p(\mathbf{X},\mathbf{h},\theta)\frac{d\log p(\mathbf{X},\mathbf{h},\theta)}{d\theta}}{\int_{\mathbf{h}} p(\mathbf{X},\mathbf{h},\theta)} = \int_{\mathbf{h}} q(\mathbf{h})\frac{d\log p(\mathbf{X},\mathbf{h},\theta)}{d\theta} \tag{14}$$

So if we take a gradient step on g, then this is equivalent to taking a gradient step on f, and we have accomplished nothing by using EM. We have to either take several gradient steps, or use a second-order technique like Newton-Raphson. The second derivative of $\log f$ is

$$\frac{d^2\log f}{d\theta^2} = \int_{\mathbf{h}} q(\mathbf{h})\frac{d^2\log p(\mathbf{X},\mathbf{h},\theta)}{d\theta^2} + \int_{\mathbf{h}} q(\mathbf{h})\left(\frac{d\log p(\mathbf{X},\mathbf{h},\theta)}{d\theta}\right)^2 - \left(\int_{\mathbf{h}} q(\mathbf{h})\frac{d\log p(\mathbf{X},\mathbf{h},\theta)}{d\theta}\right)^2 \tag{15}$$

which is the second derivative of G plus extra terms. So EM does help Newton-Raphson by simplifying the second derivative.

It is also not necessary to fully maximize G over q during the E-step (Neal and Hinton, 1993), i.e. the bound need not touch the objective function. Any local maximum of $G(\theta, q)$ (in both variables) is a local maximum of $f(\theta)$, so any way of maximizing G will do, regardless of whether it increases $f(\theta)$ at each step. We can take a small step along q, a small step along θ, change both at once, etc.

Furthermore, the representation of q need not be complete. The only properties of q that we need are those that affect the M-step. This usually amounts to the mean and covariance matrix of $\mathbf{h}$, which are therefore the only things that need to be computed in the E-step (hence its name). Even if these moments cannot be computed analytically, they can be estimated, e.g. by sampling methods.

3 EM for data pooling

As an example, consider the problem of pooling data from multiple noisy sources. We want to estimate θ which can be considered a random variable whose prior $p(\theta)$ is essentially uniform over the real line. We have two different measuring devices which produce independent measurements a and b respectively. Unfortunately, the devices can only make noisy measurements, and the noise variances are different:

$$p(a|\theta, v_a) \sim \mathcal{N}(\theta, v_a) \tag{16}$$

$$p(b|\theta, v_b) \sim \mathcal{N}(\theta, v_b) \tag{17}$$

We don't know the noise variances, so we give them noninformative priors independent of θ:

$$p(v_a|\theta) = 1/v_a \tag{18}$$

$$p(v_b|\theta) = 1/v_b \tag{19}$$

Let the data be

$$\begin{aligned}\mathbf{A} &= \{a_i\} \qquad i = 1..N_a &(20)\\ \mathbf{B} &= \{b_i\} \qquad i = 1..N_b &(21)\end{aligned}$$

The function we want to maximize is

$$\begin{aligned}f(\theta) &= p(\mathbf{A}, \mathbf{B}, \theta) &(22)\\ &= \int_{v_a}\int_{v_b} p(\mathbf{A}, \mathbf{B}, v_a, v_b, \theta) &(23)\\ &= \int_{v_a}\int_{v_b} p(\mathbf{A}|\theta, v_a)p(v_a|\theta)p(\mathbf{B}|\theta, v_b)p(v_b|\theta)p(\theta) &(24)\end{aligned}$$

which has the form in (3) suited to EM. The nuisance parameters here are the variances.

For the E-step, we compute the following using our current guess for θ (call it θ^{old}):

$$\begin{aligned}q(v_a, v_b) &= p(v_a, v_b|\mathbf{A}, \mathbf{B}, \theta^{\text{old}}) &(25)\\ &= p(v_a|\mathbf{A}, \theta^{\text{old}})p(v_b|\mathbf{B}, \theta^{\text{old}}) &(26)\\ &= q_a(v_a)q_b(v_b) &(27)\\ q_a(v_a) &\sim \chi^{-2}(S_a^{\text{old}}, N_a) &(28)\\ &= \frac{1}{\Gamma(N_a/2)v_a}\left(\frac{S_a^{\text{old}}}{2v_a}\right)^{N_a/2}\exp(-\frac{S_a^{\text{old}}}{2v_a}) &(29)\\ q_b(v_b) &\sim \chi^{-2}(S_b^{\text{old}}, N_b) &(30)\\ S_a^{\text{old}} &= \sum_i (a_i - \theta^{\text{old}})^2 &(31)\\ S_b^{\text{old}} &= \sum_i (b_i - \theta^{\text{old}})^2 &(32)\end{aligned}$$

So the logarithm of the lower bound is

$$\begin{aligned}G(\theta, q) &= \int_{v_a}\int_{v_b} q(v_a, v_b)\log p(\mathbf{A}, \mathbf{B}, v_a, v_b, \theta) - q(v_a, v_b)\log q(v_a, v_b) &(33)\\ &= E_{q_a(v_a)}\left[\log p(\mathbf{A}|v_a, \theta) + \log p(v_a|\theta) - \log q_a(v_a)\right] + &(34)\\ &\quad E_{q_b(v_b)}\left[\log p(\mathbf{B}|v_b, \theta) + \log p(v_b|\theta) - \log q_b(v_b)\right] + \log p(\theta)\\ &= E_{q_a(v_a)}\left[\frac{S_a^{\text{old}} - S_a}{2v_a}\right] + E_{q_b(v_b)}\left[\frac{S_b^{\text{old}} - S_b}{2v_b}\right] &(35)\\ &\quad -\frac{N_a}{2}\log\left(\pi S_a^{\text{old}}\right) - \frac{N_b}{2}\log\left(\pi S_b^{\text{old}}\right) + \log\Gamma\left(\frac{N_a}{2}\right)\Gamma\left(\frac{N_b}{2}\right)p(\theta)\\ &= -\frac{N_a S_a}{2S_a^{\text{old}}} - \frac{N_b S_b}{2S_b^{\text{old}}} - \frac{N_a}{2}\log\left(\frac{\pi}{e}S_a^{\text{old}}\right) - \frac{N_b}{2}\log\left(\frac{\pi}{e}S_b^{\text{old}}\right) + \log\Gamma\left(\frac{N_a}{2}\right)\Gamma\left(\frac{N_b}{2}\right)p(\theta) &(36)\end{aligned}$$

For the M-step, we compute a new guess for θ. Only the first two terms in (36) depend on θ, so we get

$$\begin{aligned} \frac{\mathrm{d}G}{\mathrm{d}\theta} &= -\frac{N_a}{2S_a^{\text{old}}}\frac{\mathrm{d}S_a}{\mathrm{d}\theta} - \frac{N_b}{2S_b^{\text{old}}}\frac{\mathrm{d}S_b}{\mathrm{d}\theta} && (37) \\ &= \frac{N_a}{S_a^{\text{old}}}\sum_i (a_i - \theta) + \frac{N_b}{S_b^{\text{old}}}\sum_i (b_i - \theta) = 0 && (38) \\ \theta &= \frac{\frac{N_a \sum_i a_i}{S_a^{\text{old}}} + \frac{N_b \sum_i b_i}{S_b^{\text{old}}}}{\frac{N_a^2}{S_a^{\text{old}}} + \frac{N_b^2}{S_b^{\text{old}}}} && (39) \end{aligned}$$

The EM algorithm for data pooling reduces to iteratively computing (31), (32), and (39) until θ stops changing.

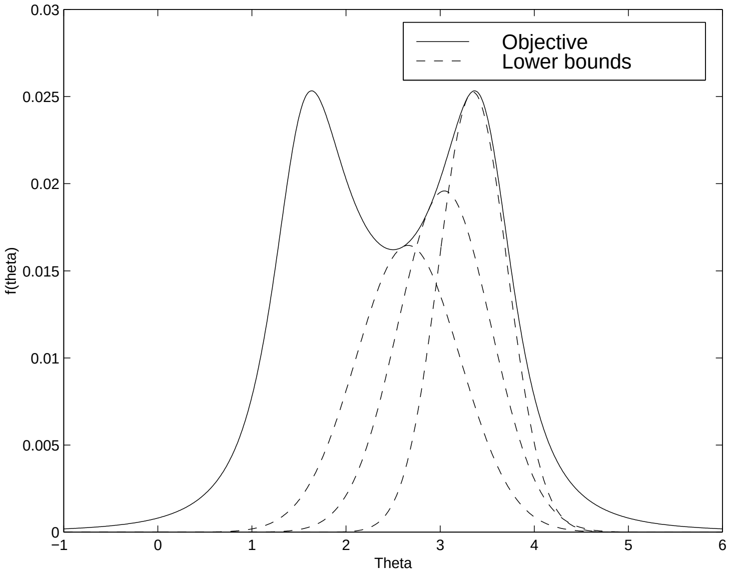

Figure 2: The objective $f(\theta)$ and successive lower bounds $g(\theta, q)$ after 1, 4, and 9 iterations of EM starting from $\theta = 2.6$.

Figure 2 illustrates the algorithm in action. The function $f(\theta)$ can be determined analytically as the product of two T densities. In general, it will be bimodal, and the EM algorithm will converge to the local maximum nearest to the starting guess for θ. A pathological case occurs when θ is started exactly on the valley of the function, where the gradient is zero: the algorithm gets stuck there.

4 EM for a mixture model

A finite mixture model is a density for $\mathbf{x}$ which has the form of a weighted sum of component densities:

$$p(\mathbf{x}|\theta) = \sum_{c=1}^{K} p(\mathbf{x}|c,\theta)p(c|\theta) \tag{40}$$

For example, if $p(\mathbf{x}|c,\theta)$ is Gaussian then $p(\mathbf{x}|\theta)$ is a weighted sum of K Gaussians. Note that we are treating c here as a random variable whose value we don't know. If we knew c, then the density for $\mathbf{x}$ is just the cth component density. Hence c is called the *hidden assignment* for $\mathbf{x}$ to one of the component densities. If we have several independent samples $\mathbf{x}_i$ then each has its own assignment variable c_i. You can think of the $\mathbf{x}_i$'s and c_i's as labeled training data where some or all of the labels are missing.

There are two main independence assumptions implicit in the finite mixture model. First, if θ is known then the observed data points are statistically independent. Second, if θ is known then the hidden assignments are independent.

These two properties create a simplification in the EM algorithm. The independence of the c_i allows us to represent the multidimensional q distribution with a product of one-dimensional distributions:

$$q(c_1..c_N) = q_1(c_1)..q_N(c_N)$$

Since c_i only takes on values $j = 1..K$, the functions $q_i(c_i)$ can be represented by the $N \times K$ matrix $\mathbf{Q}$ where $q_{ij} = q_i(j)$. Furthermore, the independence of the data points $\mathbf{x}_i$ means $\log p(\mathbf{X}, \mathbf{c}, \theta) = \sum_i \log p(\mathbf{x}_i, \mathbf{c}_i, \theta)$.

The EM algorithm simplifies to:

E-step From (8), compute

$$q_{ij} = \frac{p(\mathbf{x}_i|c_i = j,\theta)p(c_i = j|\theta)}{\sum_j p(\mathbf{x}_i|c_i = j,\theta)p(c_i = j|\theta)} = p(c_i = j|\mathbf{x}_i,\theta)$$

M-step From (13), maximize over θ:

$$\sum_{ij} q_{ij} \log p(\mathbf{x}_i, c_i = j, \theta)$$

For a further discussion of EM applied to a mixture model see Bishop (1995) and Redner and Walker (1984).

Acknowledgements

Rosalind Picard helped improve the presentation.

References

[1] M. S. Bazaraa and C. M. Shetty. *Nonlinear Programming.* John Wiley and Sons, New York, 1979.

[2] C. Bishop. *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press, 1995.

[3] Wray Buntine. Computation with the exponential family and graphical models. Tutorial given at NATO Workshop on Learning in Graphical Models, Erice, Italy, September 1996. `http://www.ultimode.com/~wray`, 1996.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society B*, 39:1–38, 1977.

[5] Stephen P. Luttrell. Partitioned mixture distribution: an adaptive bayesian network for low-level image processing. *IEE Proc on Vision, Image and Signal Processing*, 141(4):251–260, August 1994.

[6] Radford M. Neal and Geoffrey E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. Technical report, University of Toronto, Dept of Computer Science, 1993. `http://www.cs.toronto.edu/~radford/em.abstract.html`.

[7] Richard A. Redner and Homer F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, April 1984.