Expectation-Maximization

Shanthi Devara, Ron Kneusel

CSCI 5454 Lecture 19 April 11, 2011

1 Introduction

The Expectation-Maximization (EM) algorithm is a powerful tool for estimating the parameters of a probability distribution from given data in cases where the Maximum-Likelihood (ML, see below) is difficult to calculate. To get a feel for EM and its use, consider the following situation from [3]:

> Say that the probability of the temperature outside your window for each of the 24 hours of a day $x \in \mathbb{R}^{24}$ depends on the season $\theta \in \{\text{summer, fall, winter, spring}\}$, and that you know the seasonal temperature distribution $p(x|\theta)$. But say you can only measure the average temperature $y = \bar{x}$ for the day, and you'd like to guess what season it is (for example, is spring here yet?). The maximum likelihood estimate of θ maximizes $p(y|\theta)$, but in some cases this may be hard to find. Thats when EM is useful it takes your observed data y, iteratively makes guesses about the complete data x, and iteratively finds the θ that maximizes $p(x|\theta)$ over θ. In this way, EM tries to find the maximum likelihood estimate of θ given y. [...] EM doesn't actually promise to find you the θ that maximizes $p(y|\theta)$, but there are some theoretical guarantees, and it often does a good job in practice, though it may need a little help in the form of multiple random starts.

The concept of Maximum-Likelihood will be explained below. From this example, it is easy to see that EM is a powerful technique which can be used to generate desired outcomes from partially complete data.

Though EM algorithm was first explained by many different authors it was only in 1977 that this algorithm was named and generalized by Dempster, Laird, and Rubin. The DLR (Dempster-Laird-Rubin) paper [4] sketched a convergence analysis for a wider class of problems. This paper received an enthusiastic discussion at the Royal Statistical Society and was applauded by many scholars.

However, the convergence analysis of [4] was flawed and a correct convergence analysis was published by Wu in 1983 [9]. Wu's proof established the EM method's convergence outside of the exponential family, as claimed by [4]. Later many research papers and books described the application of EM algorithm in different fields like statistics and computational biology for analysing and solving real time problems.

To describe the EM algorithm briefly we can say that, EM is an iterative algorithm that is one of the broadly applicable statistical techniques for maximizing complex likelihoods and enabling parameter estimation in probabilistic models with incomplete data. At each iteration step of the algorithm, two steps are performed. The E step computes the expectation of the log-likelihood evaluated using the current estimate for the latent variables, and the M step computes parameters for maximizing the expected log-likelihood found in the E step. These parameter estimates are then used to determine the distribution of the latent variables in the next E step. The EM algorithm relates to MCMC and Bayesian networks as a forerunner by its data augmentation step that replaces simulation by maximization.

1.1 Some Prerequisites

Before jumping into the EM algorithm some initial prerequisites need to be discussed. These are the concept of maximum-likelihood, which is foundational to EM, and the notions of convex/concave functions plus Jensen's inequality, which is important in the derivation of the EM algorithm.

1.1.1 Maximum-Likelihood

The concept of maximum-likelihood is central to understanding expectation-maximization. It is when the maximum-likelihood is difficult to find that EM is most useful. The description here is based on that in [5].

Assume that we have c classes and that we have samples, $D_1, \ldots, D_c$, of each class. The samples are i.i.d. - independent and identically distributed - random variables. This means that the samples are drawn from some probability distribution, $p(\mathbf{x}|\omega_j)$, for some class j. Also, assume that we have some knowledge of the form of this probability distribution and that it can be parameterized by a vector $\boldsymbol{\theta}_j$. This parameter might be the mean and covariance matrix of a normal distribution for example. The goal of maximum-likelihood is to find $\boldsymbol{\theta}_j$ for $j = 1, \ldots, c$ using the data given in the samples, $D_1, \ldots, D_c$. If we assume that D_i tells us nothing about class j if $i \neq j$, then we have c separate problems to consider.

Therefore, for some set of samples D which are i.i.d., we have,

$$p(D|\boldsymbol{\theta}) = \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ is an unknown parameter vector that we need to find and $p(D|\boldsymbol{\theta})$ is the likelihood of the samples D given a particular parameterization of the assumed probability density given by $\boldsymbol{\theta}$. The $\hat{\boldsymbol{\theta}}$ that maximizes $p(D|\boldsymbol{\theta})$ is the *maximum-likelihood* estimate and is therefore the parameterization of the probablity density that most likely explains the measured data. Finding $\hat{\boldsymbol{\theta}}$ may be nontrivial. If so, the EM algorithm is called for.

Note that it is simpler to work with the *log-likelihood*, $l(\boldsymbol{\theta}) \equiv \ln p(D|\boldsymbol{\theta})$ which is maximized when the likelihood is maximized since $\ln x$ is a monotonically increasing function. Therefore, the goal of maximum-likelihood estimation can be written as,

$$\hat{\boldsymbol{\theta}} = \arg\max_{x} l(\boldsymbol{\theta})$$

If we define $\boldsymbol{\nabla}_{\boldsymbol{\theta}} \equiv \left[\frac{\partial}{\partial\theta_1}, \ldots, \frac{\partial}{\partial\theta_n}\right]^T$, then the maximum-likelihood estimate can be found by solving,

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} l = \mathbf{0}$$

Again, if this solution is difficult to find, expectation-maximization is appropriate. Likewise, any algorithm that searches the space of the parameters for a maximum is fair game. This includes things like simulated annealing, genetic algorithms or particle swarm optimization.

1.1.2 Convex, Concave Functions

It is very important to understand the definitions, properties and differences between convex and concave functions as they are used in the EM algorithm.

A real valued function $f(x)$ defined on an interval or any convex subset[1] of some vector space is called convex, if for any two points x_1 and x_2 in its domain X and any $\lambda \in [0,1]$,

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2)$$

and strictly convex if,

$$f(\lambda x_1 + (1-\lambda)x_2) < \lambda f(x_1) + (1-\lambda)f(x_2)$$

Figure 1 shows this situation graphically. In this case, the interval is $[a,b]$ on which the convex function, $f(x)$, is defined. There are many properties of convex functions but only those necessary to the EM algorithm will be discussed here:

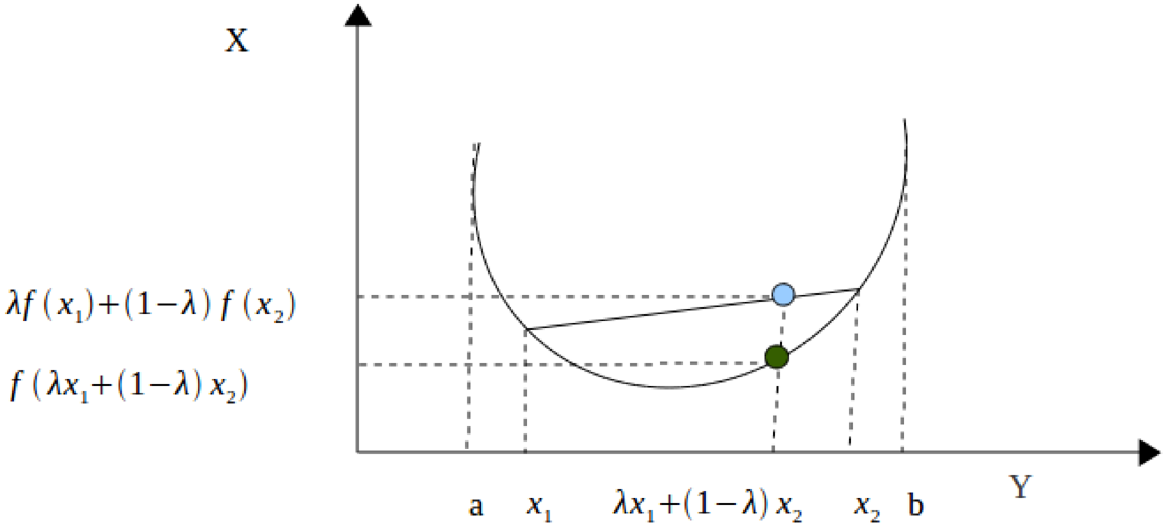

X
$\lambda f(x_1)+(1-\lambda)f(x_2)$
$f(\lambda x_1+(1-\lambda)x_2)$
a
x_1
$\lambda x_1+(1-\lambda)x_2$
x_2
b
Y

Figure 1: A graphical representation of the meaning of a convex function.

1. A function f is said to be strictly concave if the function $-f$ is strictly convex.
2. If $f(x)$ is twice differentiable on $[a,b]$ and if $f''(x) \geq 0$ on $[a,b]$, then $f(x)$ is convex on $[a,b]$. If $f''(x) < 0$ on $[a,b]$, then $f(x)$ is concave on $[a,b]$.
3. The function $-\ln(x)$ is strictly convex as shown by property 2: $f''(x) = \frac{1}{x^2} > 0$.

[1] In Euclidean space, an object is said to be convex if for every pair of points within the object, the straight line joining the points is inside the object as well.

1.1.3 Jensen's Inequality

Jensen's inequality was proposed by the Danish mathematician Johan Jensen in 1906 [6]. Many problems use this property, including the EM algorithm. The proof here is based on that found in [1].

Theorem 1.1 (Jensen's inequality) *Let f be a convex function defined on an interval I. If $x_1, x_2, \ldots, x_n \in I$ and $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$,*

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i)$$

Jensen's inequality means that the weighted means of a convex function is always greater or equal to the convex function of the weighted means. This generalizes so that the secant line of a convex function always lies above the graph of the convex function. If we consider the convex function graph in Figure 1, the secant line consists of the weighted means of the convex function (the blue circle) which is $\lambda f(x_1) + (1-\lambda) f(x_2)$, while the graph consists of a convex function of the weighted means (the green circle), $f(\lambda x_1 + (1-\lambda)x_2)$. Therefore,

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

Let $\lambda = 0.5$, $x_1 = 2$ and $x_2 = 4$ and consider the convex function $f(x) = x^2$. First, calculate the weighted means,

$$\begin{aligned} \lambda f(x_1) + (1-\lambda) f(x_2) = & \quad 0.5f(2) + (1-0.5)f(4) \\ = & \quad 0.5(2^2) + 0.5(4^2) \\ = & \quad 0.5(4) + 0.5(16) \\ = & \quad 10 \end{aligned}$$

Second, calculate the function of the weighted means,

$$\begin{aligned} f(\lambda x_1 + (1-\lambda)x_2) = & \quad f(0.5(2) + (1-0.5)(4)) \\ = & \quad f(1+2) \\ = & \quad f(3) = 9 \end{aligned}$$

Since $10 > 9$ we have $\lambda f(x_1) + (1-\lambda) f(x_2) > f(\lambda x_1 + (1-\lambda)x_2)$ thus demonstrating Jensen's inequality. It should be pointed out that if $f(x) = x$ the inequality becomes an equality. Now, let us consider a more formal proof.

Let λ_1 and λ_2 be two arbitrary, real valued numbers such that $\lambda_1 + \lambda_2 = 1$ and f be a convex function. Since f is convex we have $f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$ for any x_1 and x_2. This can be generalized for additional λ and x values provided $\lambda_1 + \lambda_2 + \cdots + \lambda_n = 1$. At least one λ must be positive for the sum to be 1. Let that λ be λ_1. Then, we have, $f(\lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n) \leq \lambda_1 f(x_1) + \lambda_2 f(x_2) + \cdots + \lambda_n f(x_n)$ for any $x_1, x_2, \ldots, x_n$.

We will prove this by induction according to the convexity hypothesis for $n = 2$ that the statement is true. Assume that it is true for some n and prove that it is true for $n+1$.

$$f\left(\sum_{i=1}^{n+1} \lambda_i x_i\right) = f\left(\lambda_1 x_1 + \sum_{i=2}^{n+1} \lambda_i x_i\right)$$

Introducing the term $(1-\lambda_1)$ and multiplying and dividing the RHS inside the function gives,

$$f\left(\sum_{i=1}^{n+1}\lambda_i x_i\right) = f\left(\lambda_1 x_1 + (1-\lambda_1)\sum_{i=2}^{n+1}\frac{\lambda_i x_i}{(1-\lambda_1)}\right)$$

If we consider the term $\sum_{i=2}^{n+1}\frac{\lambda_i x_i}{(1-\lambda_1)}$ to be x_2 then the equation above becomes,

$$f\left(\sum_{i=1}^{n+1}\lambda_i x_i\right) = f\left(\lambda_1 x_1 + (1-\lambda_1)x_2\right)$$

which, since f is convex we can write as,

$$f\left(\sum_{i=1}^{n+1}\lambda_i x_i\right) \leq \lambda_1 f(x_1) + (1-\lambda_1) f\left(\sum_{i=2}^{n+1}\frac{\lambda_i x_i}{(1-\lambda_1)}\right)$$

Since $\sum_{i=2}^{n+1}\frac{\lambda_i}{(1-\lambda_1)} = 1$ we have,

$$f\left(\sum_{i=1}^{n+1}\lambda_i x_i\right) \leq \lambda_1 f(x_1) + (1-\lambda_1) f\left(\sum_{i=2}^{n+1} x_i\right)$$

We apply the same induction hypothesis as above to the last term in the equation above to obtain the end result of Jensen's inequality, thus proving it.

In order to obtain the general inequality, where the start and end points are not defined, we will use a density argument. Consider,

$$f\left(\int x \, \mathrm{d}\mu_n(x)\right) \leq \int f(x) \, \mathrm{d}\mu_n(x)$$

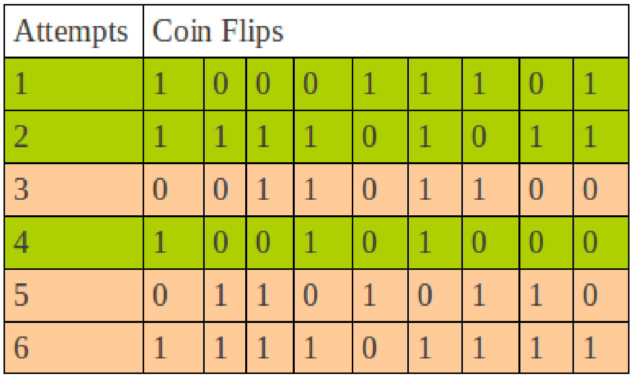
where μ_n is a convex combination of Dirac deltas given by $\mu_n = \sum_{i=1}^{n}\lambda_i\delta_i$. Since convex functions are continuous and since convex combinations of Dirac deltas are weakly dense in the set of probability measures, the general statement could be obtained by a limiting procedure.

Jensen's inequality, along with the proof that $-ln(x)$ is convex, will be used in the formal presentation of the EM algorithm below.

2 A Quick Example

To build a better understanding of the EM algorithm, consider a simple experiment: the flipping of two coins. Lets us take 2 coins A and B of unknown biases θ_A and θ_B respectively. Then the probability of heads and tails for coin A will be $P_A(H) = \theta_A$ and $P_A(T) = 1 - \theta_A$ and likewise for coin B.

Our aim is to find a parameter estimation $\theta = (\theta_A, \theta_B)$ that maximizes the likelihood. Maximum likelihood (ML) can be thought of as measuring the quality of a statistical model based on the probability of the observed data. We can find this ML estimate by flipping the coin n times and



Attempts	Coin Flips								
1	1	0	0	0	1	1	1	0	1
2	1	1	1	1	0	1	0	1	1
3	0	0	1	1	0	1	1	0	0
4	1	0	0	1	0	1	0	0	0
5	0	1	1	0	1	0	1	1	0
6	1	1	1	1	0	1	1	1	1

Figure 2: The result of the experiment conducted 6 times with 9 flips in each attempt selecting either coin A (green) or B (orange).

selecting either coin A or coin B arbitrarily in each toss. Figure 2 gives more insight (let $H = 1$ and $T = 0$. Also, $n = 9$ and $k = 6$).

Figure 3 shows the count of heads and tails for each coin in every attempt. Define two vectors $x = (x_1, x_2, \ldots, x_n)$ and $z = (z_1, z_2, \ldots, z_k)$ where x_i denotes the number of heads observed in the i-th set of tosses and $z_i \in \{\text{A,B}\}$ gives the identity of the coin in the i-th set of tosses. We can estimate θ_A and θ_B with the following formulas,

$$\hat{\theta}_A = \frac{\#\ of\ heads\ using\ coin\ A}{total\ \#\ of\ flips\ using\ coin\ A}$$

$$\hat{\theta}_B = \frac{\#\ of\ heads\ using\ coin\ B}{total\ \#\ of\ flips\ using\ coin\ B}$$

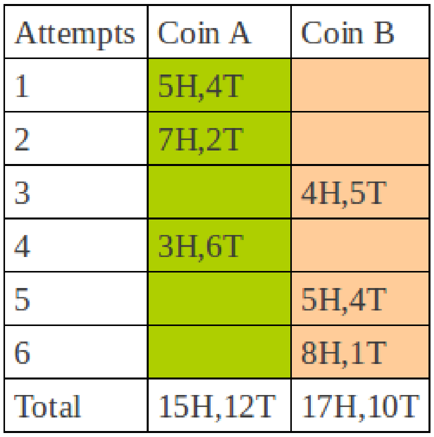
which for the example in the figures gives,

$$\hat{\theta}_A = \frac{15}{15 + 12} \approx 0.555$$

$$\hat{\theta}_B = \frac{17}{17 + 10} \approx 0.629$$

If we consider the joint probability/log-likelihood of obtaining any particular vector of observed head counts x and coin types z as $\log P(x, z; \theta)$ then parameter estimation that we get by solving $\hat{\theta}_A$ and $\hat{\theta}_B$ maximizes the log-likelihood $\log P(x, z; \theta)$.

The experiment above is straightforward since we have complete data of the number of attempts, the coins selected in each attempt, the number of flips in each attempt and the result of each flip. So calculating a maximum-likelihood estimate in this case is very easy. But consider the case where we have missing data i.e. either the head counts x is missing or the identities of the coins z is missing. Missing variables are known as *hidden variables* or *latent factors*. For example, let z be missing. The problem is still is a solvable problem where we could apply the EM algorithm to find the maximum-likelihood estimation for maximizing the head counts x_i. This comes under the case of incomplete data so our goal is to fill in this incomplete data (z in this case) in order to attain



Attempts	Coin A	Coin B
1	5H,4T	
2	7H,2T	
3		4H,5T
4	3H,6T	
5		5H,4T
6		8H,1T
Total	15H,12T	17H,10T

Figure 3: The count of heads and tails for each coin.

the maximum-likelihood. We can do this by guessing correctly which coin was used in each of the k sets.

We do this by a recursive operation i.e. let us start from some initial parameters, $\hat{\theta}^{(t)} = (\hat{\theta}_A^{(t)}, \hat{\theta}_B^{(t)})$, determine for each of the k sets which of the coins A or B would have more likely generated the observed coin flip pattern by using the current parameter estimates as a base. We then estimate that our coin assumptions are correct and apply the next step which is to calculate the maximum-likelihood estimation to get $\hat{\theta}^{(t+1)}$. Finally, repeat these two steps until convergence.

As the estimated model improves, so too will the quality of the resulting completions. The EM algorithm is truly based on the refinement of this basic idea. Instead of picking the single most likely completion of the missing coin assignments on each iteration, the EM algorithm computes probabilities for each possible completion of the missing data, using the current parameters $\hat{\theta}^{(t)}$. These probabilities are used to create a weighted training set consisting of all possible completions of the data. Finally, a modified version of maximum-likelihood estimation that deals with weighted training examples provides new parameter estimates, $\hat{\theta}^{(t+1)}$. By using weighted training examples rather than choosing the single best completion, the EM algorithm accounts for the confidence of the model in each completion of the data.

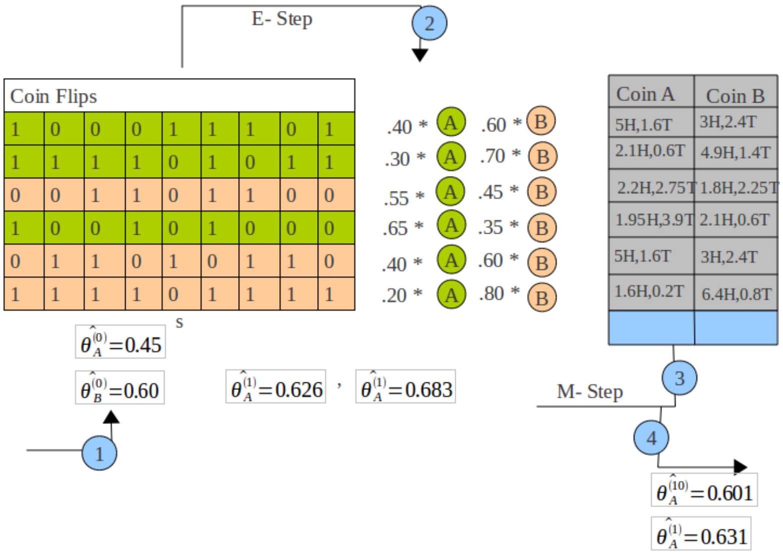
Figure 4 shows the EM process graphically for this example.

3 The EM Algorithm

Preliminaries aside, it is time to present the EM algorithm and to discuss its convergence properties as well as its runtime properties.

3.1 Formal Description

Formally, the EM algorithm is the iteration of the following two steps [10],


E- Step
2
.40 * A .60 * B
.30 * A .70 * B
.55 * A .45 * B
.65 * A .35 * B
.40 * A .60 * B
.20 * A .80 * B
s
$\hat{\theta}_A^{(0)}=0.45$
$\hat{\theta}_B^{(0)}=0.60$
1
$\hat{\theta}_A^{(1)}=0.626$, $\hat{\theta}_A^{(1)}=0.683$
M- Step
3
4
$\hat{\theta}_A^{(10)}=0.601$
$\hat{\theta}_A^{(1)}=0.631$

Coin Flips

1	0	0	0	1	1	1	0	1
1	1	1	1	0	1	0	1	1
0	0	1	1	0	1	1	0	0
1	0	0	1	0	1	0	0	0
0	1	1	0	1	0	1	1	0
1	1	1	1	0	1	1	1	1

Coin A	Coin B
5H,1.6T	3H,2.4T
2.1H,0.6T	4.9H,1.4T
2.2H,2.75T	1.8H,2.25T
1.95H,3.9T	2.1H,0.6T
5H,1.6T	3H,2.4T
1.6H,0.2T	6.4H,0.8T

Figure 4: The EM process for the coin flipping experiment. 1. Initial estimates of the coin biases. 2. Expected outcomes for each coin given the current (initial) biases. 3. Maximization of the bias values. 4. Convergence reached, final θ values.

1. *E-step*: compute $Q(\boldsymbol{\theta};\boldsymbol{\theta^t})$, which is the **expectation** of the complete-data log-likelihood, $\log p(X,Z|\boldsymbol{\theta^t})$ and the expectation is w.r.t. $p(Z|\boldsymbol{\theta}^t,X)$.
2. *M-step*: **maximize** $Q(\boldsymbol{\theta};\boldsymbol{\theta}^t)$ w.r.t. $\boldsymbol{\theta}$ to obtain $\boldsymbol{\theta^{t+1}}$.

where,

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^t) \equiv \int p(Z|\boldsymbol{\theta}^t,X)\log p(X,Z|\boldsymbol{\theta})\,\mathrm{d}Z$$

Here we have assumed that the direct computation of the maximum-likelihood is difficult and have augmented the measured data X with "hidden" data Z so that the maximization of $p(X,Z|\theta)$ is easy to compute, if both X and Z have been observed.

Now, assume we have an initial guess for the parameters, $\boldsymbol{\theta}^t$ and we want to find a new $\boldsymbol{\theta}$ so that $p(X|\boldsymbol{\theta}) \geq p(X|\boldsymbol{\theta}^t)$. If we look at the log-likelihood we see that,

$$\Delta L = L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}^t) = \log\frac{p(X|\boldsymbol{\theta})}{p(X|\boldsymbol{\theta}^t)}$$

This is where the "hidden" variable Z comes in so that $p(X,Z|\boldsymbol{\theta})$ is easy to compute. Then, we can compute the difference between the log-likelihoods by marginalizing Z,

$$\begin{aligned} L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}^t) &= \log\tfrac{\int p(X,Z|\boldsymbol{\theta})\,\mathrm{d}Z}{p(X|\boldsymbol{\theta}^t)} \\ &= \log\left[\int\frac{p(Z|\boldsymbol{\theta}^t,X)}{p(Z|\boldsymbol{\theta}^t)}\frac{p(X,Z|\boldsymbol{\theta})}{p(X|\boldsymbol{\theta}^t)}\,\mathrm{d}Z\right] \\ &\geq \int\left[p(Z|\boldsymbol{\theta}^t,X)\log\frac{p(X,Z|\boldsymbol{\theta})}{p(Z,\boldsymbol{\theta}^t,X)p(X|\boldsymbol{\theta}^t)}\right]\mathrm{d}Z \\ &\equiv \underline{\Delta}L(\boldsymbol{\theta};\boldsymbol{\theta}^t) \end{aligned}$$

where the inequality makes use of Jensen's inequality and the concavity of $\log x$ as noted above in the preliminaries.

The definition above, then, is of a quantity that will be added to the existing log-likelihood for the current parameter estimates $\boldsymbol{\theta}^t$, so that if we can maximize $\underline{\Delta}L(\boldsymbol{\theta};\boldsymbol{\theta}^t)$ we obtain a new estimate, $\boldsymbol{\theta}^t$.

To maximize $\underline{\Delta}L(\boldsymbol{\theta};\boldsymbol{\theta}^t)$ with respect to $\boldsymbol{\theta}$ we get:

$$\begin{aligned}
\boldsymbol{\theta}^{t+1} &= \arg\max_{\boldsymbol{\theta}} \underline{\Delta}L(\boldsymbol{\theta};\boldsymbol{\theta}^t) \\
&= \arg\max_{\boldsymbol{\theta}} \int \left[p(Z|\boldsymbol{\theta}^t, X) \log \frac{p(X, Z|\boldsymbol{\theta})}{p(Z|\boldsymbol{\theta}^t, X) p(X|\boldsymbol{\theta}^t)} \right] \mathrm{d}Z \\
&= \arg\max_{\boldsymbol{\theta}} \int p(Z|\boldsymbol{\theta}^t, X) \log p(X, Z|\boldsymbol{\theta})\, \mathrm{d}Z
\end{aligned}$$

Finally, define,

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^t) \equiv \int p(Z|\boldsymbol{\theta}^t, X) \log p(X, Z|\boldsymbol{\theta})\, \mathrm{d}Z$$

which is the **expectation** function for the complete log-likelihood as given in the *E-step* above. The entire EM algorithm, then, repeats the two steps until some convergence criteria, like no appreciable change in Q, is reached.

3.2 Convergence

Does the EM algorithm converge? Perhaps. What can be shown is that each new iteration of the algorithm will not make things worse. To see this, consider the following theorem,

Theorem 3.1 *Let $L(\boldsymbol{\theta})$ be the log-likelihood function. If $Q(\boldsymbol{\theta};\boldsymbol{\theta}^t) \geq Q(\boldsymbol{\theta}^t;\boldsymbol{\theta}^t)$, then $L(\boldsymbol{\theta}) \geq L(\boldsymbol{\theta}^t)$.*

With this theorem, the proof is outlined nicely in [3], we can see that since the *M-step* maximizes the Q function to give a new estimate of the parameters we will have,

$$Q(\boldsymbol{\theta}^{t+1};\boldsymbol{\theta}^t) \geq Q(\boldsymbol{\theta}^t;\boldsymbol{\theta}^t)$$

which therefore means,

$$L(\boldsymbol{\theta}^{t+1}) \geq L(\boldsymbol{\theta}^t)$$

indicating that iterations of the EM algorithm will not make things worse, even if they do not necessarily make things better.

3.3 Complexity

From an algorithms point of view, what can we say about the time and space requirements of the EM algorithm? A literature search turned up little in this regard aside from some work on the convergence rate for Gaussian mixtures. Before looking at this, let's consider the space requirements.

Neither the *E-step* nor the *M-step* introduce any new amount of data to that already present in the system (the X and any introduced Z). Therefore, it seems reasonable to claim that the space requirements of the EM algorithm are simply $O(n)$ where n is a measure of the size of the input data being used to determine the $\boldsymbol{\theta}$ values.

The time complexity is harder to pin down. Any particular implementation will have its own way of doing the expectation and maximization steps which may increase in run time as the data size, n, increases, but a general statement cannot be made. This becomes even harder to discuss when considering a generalized EM algorithm (see Section 3.4) because of the possibly stochastic way in which the log-likelihood is improved at each step.

The convergence rate for Gaussian mixtures was examined in [7] and found for this particular case to be,

$$o(e^{0.5-\epsilon}(\boldsymbol{\theta}^*))$$

where $\epsilon > 0$ is a small number, $\boldsymbol{\theta}^*$ is the true solution (parameter set) and $e(\boldsymbol{\theta}^*)$ is a measure of the overlap of the Gaussians. The implication being, for large samples, that the convergence rate for EM is superlinear when $e(\boldsymbol{\theta}^*) \to 0$. Bear in mind that this result is for a particular model type and is independent of the implementation used for the algorithm.

3.4 Generalized EM

The *M-step* of standard EM is to maximize $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^t)$ w.r.t. $\boldsymbol{\theta}$ to obtain $\boldsymbol{\theta^{t+1}}$ but the maximization requirement can be relaxed. The maximization requirement gives the largest increase in the log-likelihood for each step, but as long as there is an increase, largest or otherwise, the algorithm will still work. All that is required is that

$$Q(\boldsymbol{\theta}^{t+1}; \boldsymbol{\theta}^t) \geq Q(\boldsymbol{\theta}^t; \boldsymbol{\theta}^t)$$

be satisfied. Therefore, any means of finding a $\boldsymbol{\theta}^{t+1}$ that improves the parameter estimate will work though the algorithm will take longer to converge.

4 Examples

We conclude with two examples of the EM algorithm in action.

4.1 Two Gaussians with Missing Data

This example is from [5] and illustrates a computation involving missing data.

We are given a data set of four points in two dimensions, $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4\} = \{(0,2), (1,0), (2,2), (*,4)\}$ where the x coordinate of the fourth data point is missing. Additionally, we are told that the model is Gaussian with a diagonal covariance matrix $\boldsymbol{\Sigma}$ and an arbitrary mean $\boldsymbol{\mu}$. Therefore, $\boldsymbol{\theta} = \{\mu_1, \mu_2, \sigma_1^2, \sigma_2^2\}$. Finally, the EM algorithm needs an initial guess for $\boldsymbol{\theta}$. In this case we take the 2D Gaussian centered on the origin with $\boldsymbol{\Sigma} = \boldsymbol{I}$ so that $\boldsymbol{\theta}^0 = \{0, 0, 1, 1\}$. In this case, the "bad" or "hidden" data D_b is x_{41} and the measured data is all the rest, D_g.

To find $\boldsymbol{\theta}^1$ we must calculate Q and find the maximum,

$$\begin{aligned}
Q(\boldsymbol{\theta};\boldsymbol{\theta}^0) &= E_{x_{41}}[\log p(\boldsymbol{x}_g, \boldsymbol{x}_b; \boldsymbol{\theta})|\boldsymbol{\theta}^0; D_g] \\
&= \int_{-\infty}^{+\infty} \left[\sum_{k=1}^{3} \log p(\boldsymbol{x}_k|\boldsymbol{\theta}) + \log p(\boldsymbol{x}_4|\boldsymbol{\theta})\right] p(x_{41}|\boldsymbol{\theta}^0; x_{42} = 4)\,\mathrm{d}x_{41} \\
&= \sum_{k=1}^{3} \log p(\boldsymbol{x}_4|\boldsymbol{\theta}) + \int_{-\infty}^{+\infty} \log p\left(\begin{pmatrix} x_{41} \\ 4 \end{pmatrix}|\boldsymbol{\theta})\right) \underbrace{\frac{p(\left(\begin{smallmatrix} x_{41} \\ 4 \end{smallmatrix}\right)|\boldsymbol{\theta}^0)}{\int_{-\infty}^{+\infty} p\left(\begin{pmatrix} x'_{41} \\ 4 \end{pmatrix}|\boldsymbol{\theta}^0\right)\,\mathrm{d}x'_{41}}}_{\equiv K} \mathrm{d}x_{41}
\end{aligned}$$

Pull K out from the integral and substitute the general equation for a Gaussian,

$$\begin{aligned}
Q(\boldsymbol{\theta};\boldsymbol{\theta}^0) &= \sum_{k=1}^{3} \log p(\boldsymbol{\theta}_k|\boldsymbol{\theta}) + \frac{1}{K}\int_{-\infty}^{+\infty} \log p\left(\begin{pmatrix} x_{41} \\ 4 \end{pmatrix}|\boldsymbol{\theta}\right) \frac{1}{2\pi \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}} \exp\left[-\frac{1}{2}(x_{41}^2 + 4^2)\right] \mathrm{d}x_{41} \\
&= \sum_{k=1}^{3} \log p(\boldsymbol{\theta}_k|\boldsymbol{\theta}) - \frac{1+\mu_1^2}{2\sigma_1^2} - \frac{(4-\mu_2)^2}{2\sigma_2^2} - \log(2\pi\sigma_1\sigma_2)
\end{aligned}$$

thereby completing the *E-step*. Next, find the values of $\boldsymbol{\theta} = \{\mu_1, \mu_2, \sigma_1, \sigma_2\}$ that maximize this Q. This is $\boldsymbol{\theta}^1$, the next estimate. After some algebra, it is found that,

$$\boldsymbol{\theta}^1 = \begin{pmatrix} 0.75 \\ 2.0 \\ 0.938 \\ 2.0 \end{pmatrix}$$

This $\boldsymbol{\theta}^1$ can be used to find $\boldsymbol{\theta}^2$ and so forth until convergence is reached. In this case, convergence is reached after three iterations as can be seen graphically in Figure 5. The final solution is found to be $\boldsymbol{\mu} = \begin{pmatrix} 1.0 \\ 2.0 \end{pmatrix}$ and $\boldsymbol{\Sigma} = \begin{pmatrix} 0.667 & 0 \\ 0 & 2.0 \end{pmatrix}$.

4.2 A Cookbook Recipe for EM and GMM

This example is drawn from [3]. Here we present only the recipe for using EM to determine the parameters of a Gaussian Mixture Model (a sum of multiple Gaussians). The derivation is in [3].

Given n i.i.d. samples, $y_1, y_2, \ldots, y_n \in \mathbb{R}^d$ from a GMM with k components, estimate the parameter set $\boldsymbol{\theta} = \{(\omega_j, \mu_j, \Sigma_j)\}_{i=1}^{k}$ using the recipe below.

1. **Initialization:** Choose initial estimates for $\omega_j^{(0)}, \mu_j^{(0)}, \Sigma_j^{(0)}, j = 1, \ldots, k$ and find the initial log-likelihood

$$L^{(0)} = \frac{1}{n}\sum_{i=1}^{n} \log\left(\sum_{j=1}^{k} \omega_j^{(0)} \phi(y_i|\mu_i^{(0)}, \Sigma_j^{(0)})\right)$$

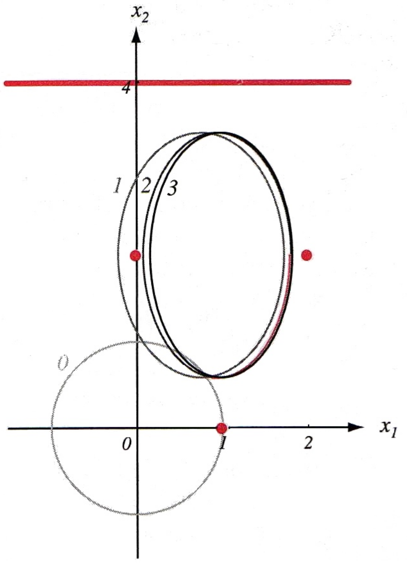


Figure 5: The iterations of the EM algorithm converging on the most likely set of parameters. The initial estimate is the circle centered at the origin. Subsequent iterations are labeled and marked showing the $1/e$ ellipse position. The initial values measured are shown in red with $\boldsymbol{x}_4$ shown as a horizontal line since the x_1 coordinate is missing. See [5], p 127.

2. **E-step:** Compute

$$\gamma_{ij}^{(m)} = \frac{\omega_j^{(m)} \phi(y_i | \mu_j^{(m)}, \Sigma_j^{(m)})}{\sum_{l=1}^{k} \omega_l^{(m)} \phi(y_i | \mu_i^{(m)}, \Sigma_l^{(m)})},\ i = 1, \ldots, n,\ j = 1, \ldots, k$$

   and

$$n_j^{(m)} = \sum_{i=1}^{n} \gamma_{ij}^{(m)},\ j = 1, \ldots, k$$

3. **M-step:** Compute the new estimate

$$\begin{aligned}
\omega_j^{(m+1)} &= \frac{n_j^{(m)}}{n},\ j = 1, \ldots, k, \\
\mu_j^{(m+1)} &= \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{ij}^{(m)} y_i,\ j = 1, \ldots, k, \\
\Sigma_j^{(m+1)} &= \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{ij}^{(m)} \left(y_i - \mu_j^{(m+1)}\right) \left(y_i - \mu_j^{(m+1)}\right)^T,\ j = 1, \ldots, k. \qquad (1)
\end{aligned}$$

4. **Convergence check:** Compute the new log-likelihood

$$L^{(m+1)} = \frac{1}{n} \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \omega_j^{(m+1)} \phi(y_i | \mu_j^{(m+1)}, \Sigma_j^{(m+1)}) \right)$$

   Return to step 2 if $|L^{(m+1)} - L^{(m)}| > \delta$ for a preset threshold, δ.

where

$$\phi(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \equiv \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right)$$

for a Gaussian of dimension d, mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

EM is a very general algorithm that can be applied to any situation where there is a model with unknown parameters or latent variables and one wishes to estimate the parameters. For example, EM has been applied to image segmentation [2] and graph analysis [8].

References

[1] S. Borman, *The Expectation-Maximization Algorithm: A short tutorial*, Unpublished paper available at *http://www.seanborman.com/publications*, version updated 2009.

[2] C. Carson, S. Belongie, H. Greenspan and J. Malik, *Blobworld: Image segmentation using Expectation-Maximization and its application to image querying*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, pp. 1026-1038, 1999.

[3] Y. Chen and M. Gupta, *EM Demystified: An Expectation-Maximization Tutorial*, Technical Report UWEETR-2010-0002, University of Washington, Seattle, WA, February 2010.

[4] A. Dempster, N. Laird, D. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society. Series B (Methodological), Vol. 39, No. 1. pp. 1-38, 1977.

[5] R. Duda, P. Hart, D. Stork, Pattern Classification, 2nd ed., Wiley, 2001.

[6] J. Jensen, *Sur les fonctions convexes et les ingalits entre les valeurs moyennes*, Acta Mathematica 30 (1): 175-193, 1906.

[7] J. Ma, L. Xu, M. Jordan, *Asymptotic Convergence Rate of the EM Algorithm for Gaussian Mixtures*, Neural Computation, Vol. 12, No. 12, pp. 2881-2907, December 2000.

[8] M. Newman, E. Leicht, *Mixture Models and Exploratory Analysis in Networks*, PNAS, Vol. 104, No. 23, pp. 9564-9569, 2007.

[9] C. Wu, *On the Convergence Properties of the EM Algorithm*, The Annals of Statistics, Vol. 11, No. 1, pp. 95-103, March 1983.

[10] J. Zhang, *Expectation Maximization*, Unpublished paper available at *http://www.stat.purdue.edu/ jianzhan/notes/EM.pdf.*