EM Demystified: An Expectation-Maximization Tutorial

Yihua Chen and Maya R. Gupta
Department of Electrical Engineering
University of Washington
Seattle, WA 98195
{yhchen,gupta}@ee.washington.edu

UWEE Technical Report
Number UWEETR-2010-0002
February 2010

Department of Electrical Engineering
University of Washington
Box 352500
Seattle, Washington 98195-2500
PHN: (206) 543-2150
FAX: (206) 543-3842
URL: http://www.ee.washington.edu

EM Demystified: An Expectation-Maximization Tutorial

Yihua Chen and Maya R. Gupta
Department of Electrical Engineering
University of Washington
Seattle, WA 98195
{yhchen,gupta}@ee.washington.edu

University of Washington, Dept. of EE, UWEETR-2010-0002
February 2010

Abstract

After a couple of disastrous experiments trying to teach EM, we carefully wrote this tutorial to give you an intuitive and mathematically rigorous understanding of EM and why it works. We explain the standard applications of EM to learning Gaussian mixture models (GMMs) and hidden Markov models (HMMs), and prepare you to apply EM to new problems. This tutorial assumes you have an advanced undergraduate understanding of probability and statistics.

1 Introduction

Expectation-maximization (EM) is a method to find the maximum likelihood estimator of a parameter θ of a probability distribution. Let's start with an example. 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 θ 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. That's 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. We'll see in later sections that 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*.

First, we go over the steps of EM, breaking down the usual two-step description into a six-step description. Table 1 summarizes the key notation. Then we present a number of examples, including Gaussian mixture model (GMM) and hidden Markov model (HMM), to show you how EM is applied. In Section 4 we walk you through the proof that the EM estimate never gets worse as it iterates. To understand EM more deeply, we show in Section 5 that EM is iteratively maximizing a tight lower bound to the true likelihood surface. In Section 6, we provide details and examples for how to use EM for learning a GMM. Lastly, we consider using EM for maximum *a posteriori* (MAP) estimation.

2 The EM Algorithm

To use EM, you must be given some observed data y, a parametric density $p(y \,|\, \theta)$, a description of some complete data x that you wish you had, and the parametric density $p(x \,|\, \theta)$. Later we'll show you how to define the complete data x for some standard EM applications. At this point, we'll just assume you've already decided what the complete data x is, and that it can be modeled as a continuous[1] random variable X with density $p(x \,|\, \theta)$, where $\theta \in \Theta$.[2] You do

[1] The treatment of discrete random variables is very similar: one only need to replace the probability density function with probability mass function and integral with summation.

[2] We assume that the support $\mathcal{X}$ of X, where $\mathcal{X}$ is the closure of the set $\{x \mid p(x \,|\, \theta) > 0\}$, does not depend on θ, for example, we do not address the case that θ is the end point of a uniform distribution.

Table 1: Notation Summary

$y \in \mathbb{R}^{d_1}$	measurement or observation you have
$Y \in \mathbb{R}^{d_1}$	random measurement; we assume you have a realization y of Y
$x \in \mathbb{R}^{d_2}$	complete data you wish you had, but instead you have $y = T(x)$
$X \in \mathbb{R}^{d_2}$	random complete data; a realization of X is x
$z \in \mathbb{R}^{d_3}$	missing data; in some problems $x = (y, z)$
$Z \in \mathbb{R}^{d_3}$	random missing data; a realization of Z is z
$\theta \in \Theta$	parameter you'd like to estimate
$\theta^{(m)} \in \Theta$	mth estimate of θ
$p(y \mid \theta)$	density of y given θ, sometimes we more explicitly but equivalently write: $p_Y(Y = y \mid \theta)$
$\mathcal{X}$	support of X, that is, the closure of the set of x such that $p(x \mid \theta) > 0$
$\mathcal{X}(y)$	support of X conditioned on y, that is, the closure of the set of x such that $p(x \mid y, \theta) > 0$
$\triangleq$	means "is defined to be"
$E_{X \mid y}[X]$	$= \int_{\mathcal{X}(y)} x p(x \mid y) dx$, you will also see this integral denoted by $E_{X \mid Y}[X \mid Y = y]$
$D_{\mathrm{KL}}(P \,\\|\, Q)$	Kullback-Leibler divergence between probability distributions P and Q

not observe X directly; instead, you observe a realization y of the random variable $Y = T(X)$ for some function T. For example, the function T might map a set X to its mean, or if X is a complex number you see only its magnitude, or T might return only the l_1 norm of some vector X, etc.

Given that you only have y, you may want to form the maximum likelihood estimate (MLE) of θ:

$$\hat{\theta}_{\mathrm{MLE}} = \arg\max_{\theta \in \Theta} p(y \mid \theta). \tag{2.1}$$

It is often easier to calculate the θ that maximizes the log-likelihood of y,

$$\hat{\theta}_{\mathrm{MLE}} = \arg\max_{\theta \in \Theta} \log p(y \mid \theta). \tag{2.2}$$

and because $\log$ is monotonic, the solution to (2.2) will be the same as the solution to (2.1).

However, for some problems it is difficult to solve either (2.1) or (2.2). Then we can try EM: we make a guess about the complete data X and solve for the θ that maximizes the (expected) log-likelihood of X. And once we have a guess for θ, we can make a better guess about the complete data X, and iterate.

EM is usually described as two steps (the E-step and the M-step), but we think it's helpful to think of EM as six distinct steps:

Step 1: Pick an initial guess $\theta^{(m=0)}$ for θ.

Step 2: Given the observed data y and pretending for the moment that your current guess $\theta^{(m)}$ is correct, calculate how likely it is that the complete data is exactly x, that is, calculate the conditional distribution $p(x \mid y, \theta^{(m)})$.

Step 3: Throw away your guess $\theta^{(m)}$, but keep Step 2's guess of the probability of the complete data $p(x \mid y, \theta^{(m)})$.

Step 4: In Step 5 we will make a new guess of θ that maximizes (the expected) $\log p(x \mid \theta)$. We'll have to maximize the *expected* $\log p(x \mid \theta)$ because we don't really know x, but luckily in Step 2 we made a guess of the probability distribution of x. So, we will integrate over all possible values of x, and for each possible value of x, we weight $\log p(x \mid \theta)$ by the *probability of seeing that* x. However, we don't really know the probability of seeing each x, all we have is the guess that we made in Step 2, which was $p(x \mid y, \theta^{(m)})$. The expected $\log p(x \mid \theta)$ is called the Q-function:[3]

$$Q(\theta \mid \theta^{(m)}) = \text{expected } \log p(x|\theta) = E_{X|y,\theta^{(m)}}[\log p(X \mid \theta)] = \int_{\mathcal{X}(y)} \log p(x \mid \theta) p(x \mid y, \theta^{(m)}) dx, \tag{2.3}$$

[3] Note this Q-function has NOTHING to do with the sum of the tail of a Gaussian, which is sometimes also called the Q-function. In EM it's called the Q-function because the original paper [1] used a Q to notate it. We like to say that the Q stands for *quixotic* because it's a bit crazy and hopeful and beautiful to think you can find the maximum likelihood estimate of θ this way.

where you integrate over the support of X given y, $\mathcal{X}(y)$, which is the closure of the set $\{x \,|\, p(x \,|\, y) > 0\}$. Note that θ is a free variable in (2.3), so the Q-function is a function of θ, and also depends on your old guess $\theta^{(m)}$.

Step 5: Make a new guess $\theta^{(m+1)}$ for θ by choosing the θ that maximizes the expected log-likelihood given in (2.3).

Step 6: Let $m = m + 1$ and go back to Step 2.

Note in Section 1 we said *EM tries to find* not that *EM finds*, because the EM estimate is *only guaranteed to never get worse* (see Section 4 for details), which often means it can find a peak in the likelihood $p(y \,|\, \theta)$, but if the likelihood function $p(y \,|\, \theta)$ has multiple peaks, EM won't necessarily find the global optimum of the likelihood. In practice, it is common to start EM from multiple random initial guesses, and choose the one with the largest likelihood as the final guess for θ.

The traditional description of the EM algorithm consists of only two steps. The above steps 2, 3, and 4 combined are called the *E-step*, and Step 5 is called the *M-step*:

E-step: Given the estimate from the mth iteration $\theta^{(m)}$, form for $\theta \in \Theta$ the Q-function given in (2.3).

M-step: The $(m + 1)$th guess of θ is:

$$\theta^{(m+1)} = \arg\max_{\theta \in \Theta} Q(\theta \,|\, \theta^{(m)}).$$

It is sometimes helpful to write the Q-function integral in a different way. Note that

$$\begin{aligned} p(x \,|\, y, \theta) &= \frac{p(x, y \,|\, \theta)}{p(y \,|\, \theta)} && \text{by Bayes rule} \\ &= \frac{p(x \,|\, \theta)}{p(y \,|\, \theta)} && \text{because } Y = T(X) \text{ is a deterministic function.} \end{aligned}$$

If the last line isn't clear, note that because $Y = T(X)$ is a deterministic function, knowing x means you know y, and thus asking for the probability of the pair (x, y) is like asking for the probability that your favorite composer is $x =$ *Bach* and $y =$ *male* – it's equivalent to the probability that your favorite composer is $x =$ *Bach*. Thus, (2.3) can be written as:

$$Q(\theta \,|\, \theta^{(m)}) = \int_{\mathcal{X}(y)} \log p(x \,|\, \theta) \frac{p(x \,|\, \theta^{(m)})}{p(y \,|\, \theta^{(m)})} dx.$$

2.1 A Toy Example

We work out an example of EM that is heavily based on an example from the original EM paper [1].[4]

Imagine you ask n kids to choose a toy out of four choices. Let $Y = \begin{bmatrix} Y_1 & \dots & Y_4 \end{bmatrix}^T$ denote the histogram of their n choices where Y_1 is the number of kids that chose toy 1, etc. We can model this random histogram Y as being distributed according to a multinomial distribution. The multinomial has two parameters: the *number of trials* $n \in \mathbb{N}$ and the probability that a kid will choose each of the four toys, which we'll call $p \in (0, 1)^4$, where $p_1 + p_2 + p_3 + p_4 = 1$. Then the probability of seeing some particular histogram y is:

$$P(y \,|\, \theta) = \frac{n!}{y_1! y_2! y_3! y_4!} p_1^{y_1} p_2^{y_2} p_3^{y_3} p_4^{y_4}.$$

For this example we assume that the unknown probability p of choosing each of the toys is parameterized by some hidden value $\theta \in (0, 1)$ such that

$$p_\theta = \begin{bmatrix} \frac{1}{2} + \frac{1}{4}\theta & \frac{1}{4}(1 - \theta) & \frac{1}{4}(1 - \theta) & \frac{1}{4}\theta \end{bmatrix}^T, \ \theta \in (0, 1).$$

The estimation problem is to guess the θ that maximizes the probability of the observed histogram of toy choices. Because we assume Y is multinomial, we can write the probability of seeing the histogram $y = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \end{bmatrix}$ as

$$P(y \,|\, \theta) = \frac{n!}{y_1! y_2! y_3! y_4!} \left(\frac{1}{2} + \frac{\theta}{4}\right)^{y_1} \left(\frac{1 - \theta}{4}\right)^{y_2} \left(\frac{1 - \theta}{4}\right)^{y_3} \left(\frac{\theta}{4}\right)^{y_4}.$$

[4] There were some historical precedents to EM before [1], but [1] is generally considered the original EM paper, and we leave a full discussion of the historical development of EM to others.

For this simple example, one could directly maximize the log-likelihood $\log P(y \mid \theta)$, but here we will instead illustrate how to use the EM algorithm to find the maximum likelihood estimate of θ.

To use EM, we need to specify what the complete data X is. We will choose the complete data to enable us to specify the probability mass function (pmf) in terms of only θ and $1-\theta$. To that end, we define the complete data to be $X = \begin{bmatrix} X_1 & \dots & X_5 \end{bmatrix}^T$, where X has a multinomial distribution with number of trials n and the probability of each event is:

$$q_\theta = \begin{bmatrix} \frac{1}{2} & \frac{1}{4}\theta & \frac{1}{4}(1-\theta) & \frac{1}{4}(1-\theta) & \frac{1}{4}\theta \end{bmatrix}^T, \; \theta \in (0,1).$$

By defining X this way, we can then write the observed data Y as:

$$Y = T(X) = \begin{bmatrix} X_1 + X_2 & X_3 & X_4 & X_5 \end{bmatrix}^T.$$

Then the likelihood of a realization x of the complete data is

$$P(x \mid \theta) = \frac{n!}{\prod_{i=1}^{5} x_i!} \left(\frac{1}{2}\right)^{x_1} \left(\frac{\theta}{4}\right)^{x_2+x_5} \left(\frac{1-\theta}{4}\right)^{x_3+x_4}. \tag{2.4}$$

For EM, we must maximize the Q-function:

$$\begin{aligned} \theta^{(m+1)} &= \arg \max_{\theta \in (0,1)} Q(\theta \mid \theta^{(m)}) \\ &\equiv \arg \max_{\theta \in (0,1)} E_{X|y,\theta^{(m)}} [\log p(X \mid \theta)] \end{aligned} \tag{2.5}$$

To solve (2.5) we actually only need the terms of $\log p(x \mid \theta)$ that depend on θ, because the other terms are irrelevant as far as maximizing over θ is concerned. Take the log of (2.4) and ignore terms that don't depend on θ to make (2.5) into:

$$\begin{aligned} \theta^{(m+1)} &= \arg \max_{\theta \in (0,1)} E_{X|y,\theta^{(m)}} [(X_2 + X_5) \log \theta + (X_3 + X_4) \log(1-\theta)] \\ &\equiv \arg \max_{\theta \in (0,1)} \left(\log \theta \left(E_{X|y,\theta^{(m)}}[X_2] + E_{X|y,\theta^{(m)}}[X_5] \right) + \log(1-\theta) \left(E_{X|y,\theta^{(m)}}[X_3] + E_{X|y,\theta^{(m)}}[X_4] \right) \right). \end{aligned} \tag{2.6}$$

To solve (2.6) we need the conditional expectation of the complete data X conditioned on already knowing the incomplete data y, which only leaves the uncertainty about X_1 and X_2. Since we do know that $X_1 + X_2 = y_1$, we can say that given y_1 the pair X_1, X_2 is binomially distributed as follows:

$$P(x \mid y, \theta) = \frac{y_1!}{x_1! x_2!} \left(\frac{2}{2+\theta}\right)^{x_1} \left(\frac{\theta}{2+\theta}\right)^{x_2} 1_{\{x_1+x_2=y_1\}} \prod_{i=3}^{5} 1_{\{x_i = y_{i-1}\}}, \tag{2.7}$$

where the $1_{\{\cdot\}}$ is the indicator function. Recognizing that (2.7) is a binomial distribution over the first two events, we know that the binomial mean (which is the expectation we need to solve (2.6)) is

$$E_{X|y,\theta}[X] = \begin{bmatrix} \frac{2}{2+\theta} y_1 & \frac{\theta}{2+\theta} y_1 & y_2 & y_3 & y_4 \end{bmatrix}^T,$$

and thus (2.6) becomes

$$\theta^{(m+1)} = \arg \max_{\theta \in (0,1)} \left(\log \theta \left(\frac{\theta^{(m)} y_1}{2+\theta^{(m)}} + y_4 \right) + \log(1-\theta)(y_2 + y_3) \right) = \frac{\frac{\theta^{(m)}}{2+\theta^{(m)}} y_1 + y_4}{\frac{\theta^{(m)}}{2+\theta^{(m)}} y_1 + y_2 + y_3 + y_4}.$$

If we choose the initial estimate as $\theta^{(0)} = 0.5$, then the algorithm reaches $\hat{\theta}_{\text{MLE}}$ to numerical precision on the 18th iteration.

2.2 Can $Y = T(X)$ be a Random Function?

Can $Y = T(X)$ be a random function? No, to get the EM guarantees, Y has to be a deterministic function of X. But that doesn't mean you can't deal with cases like $Y = X + N$, where N is some random noise drawn from some distribution. The trick is to treat $\tilde{X} = (X, N)$ as the complete data, then $Y = T(\tilde{X})$ is a deterministic function of $\tilde{X}$, and all is well.

3 EM for Missing Data

A common use of EM is to let the complete data X be the observed data Y plus some missing (also called latent or hidden) data Z, so that $X = (Y, Z)$. Two examples of EM for missing data are fitting a Gaussian mixture model (GMM) and fitting a hidden Markov model (HMM). We describe how to apply EM to these two problems in the next subsections.

In general when using EM with missing data, one can write the Q-function as an integral over the domain of Z because the only random part of X is Z:

$$\begin{aligned} Q(\theta \,|\, \theta^{(m)}) &= E_{X|y,\theta^{(m)}}\left[\log p_X(X \,|\, \theta)\right] \\ &= E_{Z|y,\theta^{(m)}}\left[\log p_X(y, Z \,|\, \theta)\right] \\ &= \int_{\mathcal{Z}} \log p_X(y, z \,|\, \theta) p_{Z|Y}(z \,|\, y, \theta^{(m)}) dz. \end{aligned}$$

3.1 Specifying the Complete Data X for Fitting a Gaussian Mixture Model

In this section we explain how to specify the complete data X for the problem of fitting a GMM using EM. This is sometimes called *EM clustering*. In words, you observe a bunch of points that you pretend were generated by k Gaussians, and you want to find the means and covariances of the Gaussians, and the probability (weight) that a point comes from each of the Gaussians. To make this estimation problem easier, you probabilistically assign each of the observed points to each of the generating Gaussians – this is the hidden information.

Now we will say the same thing in math. For a GMM with k components, the density of $Y \in \mathbb{R}^d$ is a sum of weighted Gaussian densities $\{\phi(\mu_i, \Sigma_i)\}_{i=1}^k$:

$$\begin{aligned} p(y \,|\, \theta) &= \sum_{i=1}^{k} w_i \phi(\mu_i, \Sigma_i) \\ &= \sum_{i=1}^{k} w_i \frac{\exp\left(-\frac{1}{2}(y-\mu_i)^T \Sigma_i^{-1}(y-\mu_i)\right)}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}, \end{aligned} \tag{3.1}$$

where $w_i > 0$, $i = 1, \ldots, k$, and $\sum_{i=1}^k w_i = 1$. To fit the model, one must estimate the set of k means and k covariance matrices, so one sets $\theta = \{(w_i, \mu_i, \Sigma_i)\}_{i=1}^k$.

Usually when fitting a GMM you have n observations of d-dimensional vectors drawn from (3.1), including hopefully a few observations from each Gaussian component. But since each of these samples is assumed independent, we can consider for simplicity the case of just one random observation Y from the GMM.

For one observation, we let the complete data be $X = (Y, Z)$, where $Z \in \{1, \ldots, k\}$ is a discrete random variable that defines which Gaussian component the data Y came from, so $P(Z = i) = w_i$, $i = 1, \ldots, k$, and $(Y \,|\, Z = i) \sim \mathcal{N}_d(\mu_i, \Sigma_i)$, $i = 1, \ldots, k$. Then the density of the complete data X is

$$p_X(Y = y, Z = i \,|\, \theta) = w_i \frac{\exp\left(-\frac{1}{2}(y-\mu_i)^T \Sigma_i^{-1}(y-\mu_i)\right)}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}.$$

If you then marginalize $p_X(y, i \,|\, \theta)$ over Z, you get

$$p(y \,|\, \theta) = \sum_{i=1}^{k} w_i p_X(Y = y, Z = i \,|\, \theta),$$

which is in fact (3.1), as it should be.

The details for how to fit a GMM with EM are given in Section 6.

3.2 Specifying the Complete Data X for Fitting a Hidden Markov Model

A hidden Markov model (HMM) is used to model random sequences [2]. Here we explain how to specify the complete data X to fit an HMM. For a more detailed description of EM for HMMs, see for example Bilmes' tutorial [3].

Just like with a GMM, to fit an HMM you need many observations, in this case, many observed sequences. But each of these observed sequences is considered independent and identically distributed, so we can consider the case of just one sequence.

Say you observe one sequence of length T:

$$Y = \begin{bmatrix} Y_1 & Y_2 & \dots & Y_T \end{bmatrix},$$

where each observation in the sequence is a d-dimensional vector: $Y_t \in \mathbb{R}^d$, $t = 1, \dots, T$. Then the complete data $X = (Y, Z)$ is the observed sequence Y plus the (hidden) state sequence Z:

$$Z = \begin{bmatrix} Z_1 & Z_2 & \dots & Z_T \end{bmatrix},$$

where $Z_t \in \{1, 2, \dots, k\}$, $t = 1, \dots, T$.

For example, in genomics one might be modeling a DNA sequence as an HMM, where there are two possible hidden state values: coding region or non-coding region. Thus each $Z_t \in \{\text{coding}, \text{non-coding}\}$, $k = 2$, and each observation is $Y_t \in \{A, T, C, G\}$.

In phoneme recognition, usually the first step of speech recognition, it is common to process the original time signal into MFCC (mel-filtered cepstral coefficients) features $Y_t \in \mathbb{R}^d$, and then model the sequence of MFCC features for each phoneme as a realization of an HMM, where the hidden states are the more detailed sub-phone units.

An HMM makes two assumptions. First, the conditional probability distribution of each hidden state z_t given all its previous states is equal to its conditional probability distribution given only its immediately previous state z_{t-1} (this is called the Markov property). Second, the observation y_t at time t is *conditionally independent* of other observations and states given the hidden state z_t at time t. These two assumptions can be formalized by stating:

$$p(x) = p(y, z) = \prod_{\tau=1}^{T} p(y_\tau \mid z_\tau) P(z_1) \prod_{t=2}^{T} P(z_t \mid z_{t-1}).$$

To completely specify an HMM, we need three things:

1. An initial probability distribution over the k states: $\pi = \begin{bmatrix} \pi_1 & \dots & \pi_k \end{bmatrix}^T$, where $\pi_i = P(Z_1 = i)$.

2. A transition probability matrix $\mathbf{P} \in \mathbb{R}^{k \times k}$ that specifies the probability of transitioning from state i to state j: $\mathbf{P}_{ij} = P(Z_t = j \mid Z_{t-1} = i)$.

3. The probability distribution of observations $Y \in \mathbb{R}^d$ given hidden state i; we parameterize this with parameter set θ_i, such that one can write $p(Y_t = y \mid Z_t = i) = p(y \mid \theta_i)$. For example, in modeling a DNA sequence, the θ_i is a pmf parameter that specifies the probabilities of A, T, C, and G being observed if the hidden state is $Z_t = i$. In modeling speech sequences, it is common to assume that given a particular hidden state, the observed MFCC feature vector Y_t is drawn from a GMM that corresponds to that hidden state. Then the parameter set θ_i for the ith hidden state includes all the parameters for the corresponding GMM, so $\theta_i = \{(w_{ij}, \mu_{ij}, \Sigma_{ij})\}_{j=1}^{M_i}$, where M_i is the number of components for the ith hidden state's GMM.

Thus for an HMM the complete set of parameters to estimate is $\theta = \{\pi, \mathbf{P}, \theta_1, \dots, \theta_k\}$.

4 Monotonicity

What theoretical guarantees does EM have? Here's what you can prove: as the EM algorithm iterates, the $(m+1)$th guess $\theta^{(m+1)}$ will never be less likely than the mth guess $\theta^{(m)}$. This property is called the *monotonicity* of the EM algorithm, and follows from the following theorem, which states that improving the Q-function will at least not make the log-likelihood worse:

Theorem 4.1. *Let $L(\theta) = \log p(y \mid \theta)$ be the log-likelihood function. For $\theta \in \Theta$, if $Q(\theta \mid \theta^{(m)}) \geq Q(\theta^{(m)} \mid \theta^{(m)})$, then $L(\theta) \geq L(\theta^{(m)})$.*

We first discuss the theorem, then prove it. For the EM algorithm, the M-step ensures that

$$\theta^{(m+1)} = \arg\max_{\theta \in \Theta} Q(\theta \,|\, \theta^{(m)}),$$

and hence it must be that

$$Q(\theta^{(m+1)} \,|\, \theta^{(m)}) \geq Q(\theta^{(m)} \,|\, \theta^{(m)}).$$

Therefore we can apply Theorem 4.1 and conclude that

$$L(\theta^{(m+1)}) \geq L(\theta^{(m)}).$$

The monotonicity of the EM algorithm guarantees that as EM iterates, its guesses won't get worse in terms of their likelihood, but the monotonicity alone cannot guarantee the convergence of the sequence $\{\theta^{(m)}\}$.[5] Indeed, there is no general convergence theorem for the EM algorithm: the convergence of the sequence $\{\theta^{(m)}\}$ depends on the characteristics of $L(\theta)$ and $Q(\theta \,|\, \theta')$, and also the starting point $\theta^{(0)}$. Under certain regularity conditions, we can prove that $\{\theta^{(m)}\}$ converges to a stationary point (not necessarily a local maximum) of $L(\theta)$. See [4] for a detailed discussion; other discussions on the convergence can be found in [1, 5, 6].

We have seen that instead of solving the potentially difficult problem of directly maximizing $L(\theta)$, the EM algorithm chooses to repeatedly maximize $Q(\theta \,|\, \theta^{(m)})$, but sometimes this maximization problem is still difficult. There are many ways to deal with a hard or intractable M-step [7], and one of them is to merely seek to increase $Q(\theta \,|\, \theta^{(m)})$ by finding a $\theta^{(m+1)} \in \Theta$ that satisfies $Q(\theta^{(m+1)} \,|\, \theta^{(m)}) > Q(\theta^{(m)} \,|\, \theta^{(m)})$. This is called the *generalized* EM or GEM algorithm. By Theorem 4.1, the GEM algorithm retains the monotonicity.

We encourage you to read the proof of Theorem 4.1 below, which we have written in a tutorial style:

Proof. We first derive a lower bound on the log-likelihood function:[6]

$$\begin{aligned}
L(\theta) &= \log p(y \,|\, \theta) && \text{by definition} \\
&= \log \int_{\mathcal{X}(y)} p(x, y \,|\, \theta) dx && \text{by the law of total probability} \\
&= \log \int_{\mathcal{X}(y)} p(x \,|\, \theta) dx && \text{because } y = T(x) \text{ is deterministic} \\
&= \log \int_{\mathcal{X}(y)} \frac{p(x \,|\, \theta)}{p(x \,|\, y, \theta^{(m)})} p(x \,|\, y, \theta^{(m)}) dx && \text{multiply top and bottom by the same thing} \\
&= \log E_{X|y,\theta^{(m)}} \left[\frac{p(X \,|\, \theta)}{p(X \,|\, y, \theta^{(m)})} \right] && \text{rewrite as an expectation} \\
&\geq E_{X|y,\theta^{(m)}} \left[\log \frac{p(X \,|\, \theta)}{p(X \,|\, y, \theta^{(m)})} \right] && \text{by Jensen's inequality} \\
&= E_{X|y,\theta^{(m)}} \left[\log p(X \,|\, \theta) \right] \\
&\quad + E_{X|y,\theta^{(m)}} \left[-\log p(X \,|\, y, \theta^{(m)}) \right] \\
&= Q(\theta \,|\, \theta^{(m)}) + E_{X|y,\theta^{(m)}} \left[-\log p(X \,|\, y, \theta^{(m)}) \right] && (4.1)
\end{aligned}$$

where in the last line we used the definition of the Q-function in (2.3). The second term of (4.1) is called the *differential entropy* of X given $Y = y$ and $\theta^{(m)}$, which we denote as $h(X \,|\, y, \theta^{(m)})$, and thus we can conclude the first part of the proof by restating (4.1) as a lower bound on the log-likelihood:

$$L(\theta) \geq Q(\theta \,|\, \theta^{(m)}) + h(X \,|\, y, \theta^{(m)}). \tag{4.2}$$

Notice that in this lower bound, $Q(\theta \,|\, \theta^{(m)})$ is the only term that depends on θ.

[5] If $L(\theta)$ is bounded above on Θ, then the monotonicity implies the convergence of the sequence $\{L(\theta^{(m)})\}$, but not of the sequence $\{\theta^{(m)}\}$.

[6] This proof is one place where we need the previous assumption (see footnote 2 on page 1) that the parameter θ does not affect the support of X, because (for example) in the 4th line of the proof you multiply top and bottom by the same factor, but if the support depends on θ then you could get a $0/0$ factor when you do that, and the rest of the proof won't follow.

Next, consider the lower bound given by (4.2) for the special case that $\theta = \theta^{(m)}$:

$$\begin{aligned}
&Q(\theta^{(m)} \,|\, \theta^{(m)}) + h(X \,|\, y, \theta^{(m)}) \\
&= E_{X|y,\theta^{(m)}}\left[\log p(X \,|\, \theta^{(m)})\right] + E_{X|y,\theta^{(m)}}\left[-\log p(X \,|\, y, \theta^{(m)})\right] \\
&= \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m)}) \log p(x \,|\, \theta^{(m)}) dx - \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m)}) \log p(x \,|\, y, \theta^{(m)}) dx \\
&= \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m)}) \log \frac{p(x \,|\, \theta^{(m)})}{p(x \,|\, y, \theta^{(m)})} dx \\
&= \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m)}) \log p(y \,|\, \theta^{(m)}) dx && \text{because } p(x \,|\, \theta^{(m)}) = p(x, y \,|\, \theta^{(m)}) \\
&= \log p(y \,|\, \theta^{(m)}) && \text{because } \log p(y \,|\, \theta^{(m)}) \text{ can be pulled out of integral} \\
&\triangleq L(\theta^{(m)}) && \text{by definition.} \qquad (4.3)
\end{aligned}$$

The theorem assumes that $Q(\theta \,|\, \theta^{(m)}) \geq Q(\theta^{(m)} \,|\, \theta^{(m)})$, and thus we can conclude that:

$$\begin{aligned}
L(\theta) &\geq Q(\theta \,|\, \theta^{(m)}) + h(X \,|\, y, \theta^{(m)}) && \text{by (4.2)} \\
&\geq Q(\theta^{(m)} \,|\, \theta^{(m)}) + h(X \,|\, y, \theta^{(m)}) && \text{by theorem's assumption} \\
&= L(\theta^{(m)}) && \text{by (4.3),}
\end{aligned}$$

which completes the proof. □

5 Maximization-Maximization

Another way to view the EM algorithm is as a joint maximization procedure that iteratively maximizes a better and better lower bound F to the true likelihood $L(\theta)$ you would like to maximize [8]. Specifically, let $\tilde{P}$ denote a distribution of X with support $\mathcal{X}(y)$ and density $\tilde{p}(x)$. Let P_θ denote the conditional distribution with density $p(x \,|\, y, \theta)$. Then we will show that EM maximizes the following objective function alternately with respect to $\tilde{P}$ and θ:

$$F(\tilde{P}, \theta) = L(\theta) - D_{\mathrm{KL}}(\tilde{P} \,\|\, P_\theta),$$

where $D_{\mathrm{KL}}(\tilde{P} \,\|\, P_\theta)$ is the Kullback-Leibler divergence (a.k.a. relative entropy) between our current guess of the distribution over the complete data $\tilde{P}$ and the likelihood of the complete data given the parameter θ.

The alternating maximization steps are:

Max Step 1: Given the estimate from the previous iteration $\theta^{(m-1)}$, maximize $F(\tilde{P}, \theta^{(m-1)})$ over $\tilde{P}$ to find

$$\tilde{P}^{(m)} = \arg\max_{\tilde{P}} F(\tilde{P}, \theta^{(m-1)}). \qquad (5.1)$$

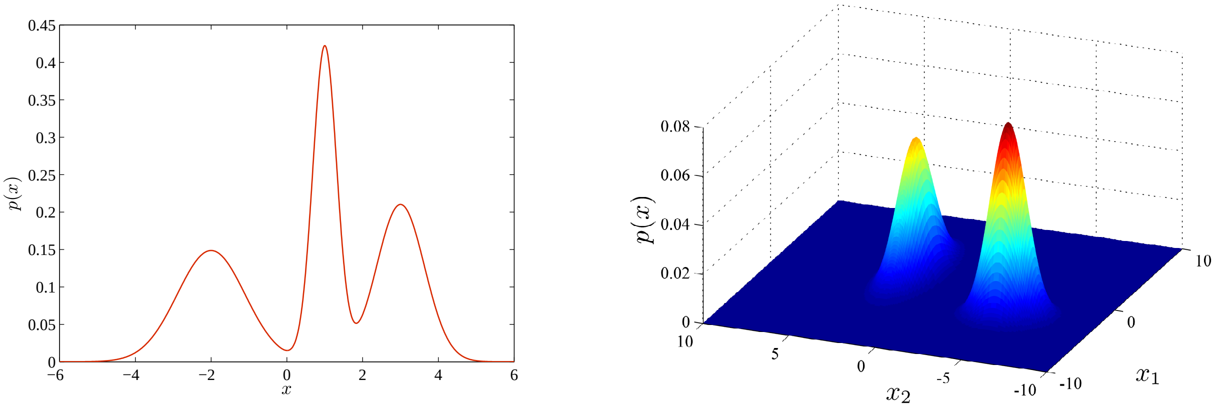
Max Step 2: Maximize $F(\tilde{P}^{(m)}, \theta)$ over θ to find

$$\theta^{(m)} = \arg\max_{\theta \in \Theta} F(\tilde{P}^{(m)}, \theta). \qquad (5.2)$$

Since both steps perform maximization, this view of the EM algorithm is called *maximization-maximization*. This joint maximization view of EM is useful as it has led to variants of the EM algorithm that use alternative strategies to maximize $F(\tilde{P}, \theta)$, for example by performing partial maximization in the first maximization step (see [8] for details).

Next we show that this really is the same as the EM algorithm. First, note that (5.1) can be simplified:

$$\tilde{P}^{(m)} = \arg\max_{\tilde{P}} \left(L(\theta^{(m-1)}) - D_{\mathrm{KL}}(\tilde{P} \,\|\, P_{\theta^{(m-1)}})\right) = \arg\min_{\tilde{P}} D_{\mathrm{KL}}(\tilde{P} \,\|\, P_{\theta^{(m-1)}}) = P_{\theta^{(m-1)}},$$



(a)

(b)

Figure 1: (a) Probability density of a 1-dimensional GMM with three components: $\mu_1 = -2$, $\mu_2 = 1$, $\mu_3 = 3$, $\sigma_1^2 = 0.8$, $\sigma_2^2 = 0.1$, $\sigma_3^2 = 0.4$, $w_1 = w_2 = w_3 = 1/3$. (b) Probability density of a 2-dimensional GMM with two components: $\mu_1 = [1 \;\; 2]^T$, $\mu_2 = [-3 \;\; -5]^T$, $\Sigma_1 = \text{diag}(4, 0.5)$, $\Sigma_2 = I_2$, $w_1 = w_2 = 0.5$.

that is, $\tilde{P}^{(m)}$ has density $p(x \,|\, y, \theta^{(m-1)})$. Second, (5.2) can be rewritten as the Q-function:

$$\begin{aligned}
\theta^{(m)} &= \arg\max_{\theta \in \Theta} \; L(\theta) - D_{\text{KL}}(\tilde{P}^{(m)} \,\|\, P_\theta) \\
&= \arg\max_{\theta \in \Theta} \; \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m-1)}) \log p(y \,|\, \theta) dx - D_{\text{KL}}(\tilde{P}^{(m)} \,\|\, P_\theta) \\
&= \arg\max_{\theta \in \Theta} \; \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m-1)}) \log \frac{p(x \,|\, \theta)}{p(x \,|\, y, \theta)} dx - \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m-1)}) \log \frac{p(x \,|\, y, \theta^{(m-1)})}{p(x \,|\, y, \theta)} dx \\
&= \arg\max_{\theta \in \Theta} \; \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m-1)}) \log p(x \,|\, \theta) dx - \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m-1)}) \log p(x \,|\, y, \theta^{(m-1)}) dx \\
&= \arg\max_{\theta \in \Theta} \; \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m-1)}) \log p(x \,|\, \theta) dx \\
&= \arg\max_{\theta \in \Theta} \; E_{X|y,\theta^{(m-1)}} [\log p(X \,|\, \theta)] \\
&= \arg\max_{\theta \in \Theta} \; Q(\theta \,|\, \theta^{(m-1)}),
\end{aligned}$$

which is just the standard M-step shown in Section 2.

6 Gaussian Mixture Model

We have introduced GMM in Section 3.1. Figure 1 shows the probability density funcitions of a 1-dimensional GMM with three components and a 2-dimensional GMM with two components, respectively. In this section, we illustrate how to use the EM algorithm to estimate the parameters of a GMM.

6.1 A Helpful Proposition

Before we proceed, we need to mention a proposition that can help simplify the computation of $Q(\theta \,|\, \theta^{(m)})$.

Proposition 6.1. *Let the complete data X consist of n i.i.d. samples: X_1, . . ., X_n, that is, $p(x \,|\, \theta) = \prod_{i=1}^n p(x_i \,|\, \theta)$ for all $x \in \mathcal{X}$ and for all $\theta \in \Theta$, and let $y_i = T(x_i)$, $i, = 1, \ldots, n$, then*

$$Q(\theta \,|\, \theta^{(m)}) = \sum_{i=1}^{n} Q_i(\theta \,|\, \theta^{(m)}),$$

where

$$Q_i(\theta \,|\, \theta^{(m)}) = E_{X_i|y_i,\theta^{(m)}}\left[\log p(X_i \,|\, \theta)\right], \quad i = 1, \ldots, n.$$

Proof. This is because

$$\begin{aligned}
Q(\theta \,|\, \theta^{(m)}) &= E_{X|y,\theta^{(m)}}[\log p(X \,|\, \theta)] \\
&= E_{X|y,\theta^{(m)}}\left[\log \prod_{i=1}^{n} p(X_i \,|\, \theta)\right] && \text{by the i.i.d. assumption} \\
&= E_{X|y,\theta^{(m)}}\left[\sum_{i=1}^{n} \log p(X_i \,|\, \theta)\right] \\
&= \sum_{i=1}^{n} E_{X_i|y,\theta^{(m)}}\left[\log p(X_i \,|\, \theta)\right] \\
&= \sum_{i=1}^{n} E_{X_i|y_i,\theta^{(m)}}\left[\log p(X_i \,|\, \theta)\right] && \text{because } p(x_i \,|\, y, \theta^{(m)}) = p(x_i \,|\, y_i, \theta^{(m)}),
\end{aligned}$$

where $p(x_i \,|\, y, \theta^{(m)}) = p(x_i \,|\, y_i, \theta^{(m)})$ is because of the i.i.d. assumption, $y_i = T(x_i)$, and Bayes' rule. □

6.2 Derivation of EM for GMM Fitting

Now given n i.i.d. samples $y_1, y_2, \ldots, y_n \in \mathbb{R}^d$ from a GMM with k components, consider the problem of estimating its parameter set $\theta = \{(w_j, \mu_j, \Sigma_j)\}_{j=1}^{k}$. Let

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

and define $\gamma_{ij}^{(m)}$ to be your guess at the mth iteration of the probability that the ith sample belongs to the jth Gaussian component, that is,

$$\gamma_{ij}^{(m)} \triangleq P(Z_i = j \,|\, Y_i = y_i, \theta^{(m)}) = \frac{w_j^{(m)} \phi(y_i \,|\, \mu_j^{(m)}, \Sigma_j^{(m)})}{\sum_{l=1}^{k} w_l^{(m)} \phi(y_i \,|\, \mu_l^{(m)}, \Sigma_l^{(m)})},$$

which satisfies $\sum_{j=1}^{k} \gamma_{ij}^{(m)} = 1$.

First, we have

$$\begin{aligned}
Q_i(\theta \,|\, \theta^{(m)}) &= E_{Z_i|y_i,\theta^{(m)}}\left[\log p_X(y_i, Z_i \,|\, \theta)\right] \\
&= \sum_{j=1}^{k} \gamma_{ij}^{(m)} \log p_X(y_i, j \,|\, \theta) \\
&= \sum_{j=1}^{k} \gamma_{ij}^{(m)} \log w_j \phi(y_i \,|\, \mu_j, \Sigma_j) \\
&= \sum_{j=1}^{k} \gamma_{ij}^{(m)} \left(\log w_j - \frac{1}{2}\log|\Sigma_j| - \frac{1}{2}(y_i - \mu_j)^T \Sigma_j^{-1} (y_i - \mu_j)\right) + \text{constant}.
\end{aligned}$$

Then according to Proposition 6.1, we obtain[7]

$$Q(\theta \,|\, \theta^{(m)}) = \sum_{i=1}^{n}\sum_{j=1}^{k} \gamma_{ij}^{(m)} \left(\log w_j - \frac{1}{2}\log|\Sigma_j| - \frac{1}{2}(y_i - \mu_j)^T \Sigma_j^{-1} (y_i - \mu_j)\right),$$

[7] We drop the constant term in $Q_i(\theta \,|\, \theta^{(m)})$ when we sum $Q_i(\theta \,|\, \theta^{(m)})$ to get $Q(\theta \,|\, \theta^{(m)})$.

which completes the E-step. Let

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

and we can rewrite $Q(\theta \mid \theta^{(m)})$ as

$$Q(\theta \mid \theta^{(m)}) = \sum_{j=1}^{k} n_j^{(m)} \left(\log w_j - \frac{1}{2} \log|\Sigma_j| \right) - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij}^{(m)} (y_i - \mu_j)^T \Sigma_j^{-1} (y_i - \mu_j).$$

The M-step is to solve

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & Q(\theta \mid \theta^{(m)}) \\ \text{subject to} \quad & \sum_{j=1}^{k} w_j = 1,\ w_j \geq 0,\ j = 1, \ldots, k, \\ & \Sigma_j \succ 0,\ j = 1, \ldots, k, \end{aligned}$$

where $\Sigma_j \succ 0$ means that Σ_j is positive definite. The above optimization problem turns out to be much easier to solve than directly maximizing the following log-likelihood function

$$L(\theta) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} w_j \phi(y_i \mid \mu_j, \Sigma_j) \right).$$

To solve for the weights, we form the Lagrangian[8]

$$J(w, \lambda) = \sum_{j=1}^{k} n_j^{(m)} \log w_j + \lambda \left(\sum_{j=1}^{k} w_j - 1 \right),$$

and the optimal weights satisfy

$$\frac{\partial J}{\partial w_j} = \frac{n_j^{(m)}}{w_j} + \lambda = 0, \quad j = 1, \ldots, k. \tag{6.1}$$

Combine (6.1) with the constraint that $\sum_{j=1}^{k} w_j = 1$, and we have

$$w_j^{(m+1)} = \frac{n_j^{(m)}}{\sum_{j=1}^{k} n_j^{(m)}} = \frac{n_j^{(m)}}{n}, \quad j = 1, \ldots, k.$$

To solve for the means, we let

$$\frac{\partial Q(\theta \mid \theta^{(m)})}{\partial \mu_j} = \Sigma_j^{-1} \left(\sum_{i=1}^{n} \gamma_{ij}^{(m)} y_i - n_j^{(m)} \mu_j \right) = 0, \quad j = 1, \ldots, k,$$

which yields

$$\mu_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{ij}^{(m)} y_i, \quad j = 1, \ldots, k.$$

To solve for the covariance matrix, we let[9]

$$\begin{aligned} \frac{\partial Q(\theta \mid \theta^{(m)})}{\partial \Sigma_j} &= -\frac{1}{2} n_j^{(m)} \frac{\partial}{\partial \Sigma_j} \log|\Sigma_j| - \frac{1}{2} \sum_{i=1}^{n} \gamma_{ij}^{(m)} \frac{\partial}{\partial \Sigma_j} (y_i - \mu_j)^T \Sigma_j^{-1} (y_i - \mu_j) \\ &= -\frac{1}{2} n_j^{(m)} \Sigma_j^{-1} + \frac{1}{2} \sum_{i=1}^{n} \gamma_{ij}^{(m)} \Sigma_j^{-1} (y_i - \mu_j)(y_i - \mu_j)^T \Sigma_j^{-1} \\ &= 0, \quad j = 1, \ldots, k, \end{aligned}$$

[8] Not comfortable with the method of Lagrange multipliers? There are a number of excellent tutorials on this topic; see for example `http://www.slimy.com/~steuard/teaching/tutorials/Lagrange.html`.

[9] See [9] for matrix derivatives.

and get

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

We summarize the whole procedure below.

EM algorithm for estimating GMM parameters

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

$$L^{(0)} = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} w_j^{(0)} \phi(y_i \,|\, \mu_j^{(0)}, \Sigma_j^{(0)}) \right).$$

2. **E-step:** Compute

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

and

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

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

$$w_j^{(m+1)} = \frac{n_j^{(m)}}{n}, \ j = 1, \ldots, k,$$

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

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

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

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

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

6.3 Convergence and Initialization

For an analysis of the convergence of the EM algorithm for fitting GMMs see [10].

The k-means algorithm is often used to find a good initialization. Basically, the k-means algorithm provides a coarse estimate of $P(Z_i = j \,|\, Y_i = y_i)$, which can be stated as

$$\gamma_{ij}^{(-1)} = \begin{cases} 1, & x_i \text{ is in cluster } j, \\ 0, & \text{otherwise.} \end{cases}$$

With $\gamma_{ij}^{(-1)}$, $i = 1, \ldots, n$, $j = 1, \ldots, k$, we can use the same formulas in the M-step to obtain $w_j^{(0)}$, $\mu_j^{(0)}$, $\Sigma_j^{(0)}$, $j = 1, \ldots, k$.

6.4 An Example of GMM Fitting

Consider a 2-component GMM in $\mathbb{R}^2$ with the following parameters

$$\mu_1 = \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \ \mu_2 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}, \ \Sigma_1 = \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \ \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \ w_1 = 0.6, \ w_2 = 0.4.$$

Its density is shown in Figure 2(a). Figure 2(b) shows 1000 samples randomly drawn from this distribution; samples from the 1st and 2nd components are marked red and blue, respectively. We ran the k-means algorithm on these samples and used the centroids of the two clusters as the initial estimates of the means:

$$\mu_1^{(0)} = \begin{bmatrix} 0.0823 \\ 3.9189 \end{bmatrix}, \quad \mu_2^{(0)} = \begin{bmatrix} -2.0706 \\ -0.2327 \end{bmatrix}.$$

Also, we let $w_1^{(0)} = w_2^{(0)} = 0.5$ and $\Sigma_1^{(0)} = \Sigma_2^{(0)} = I_2$. The density corresponding to these initial estimates is shown in Figure 2(c). We set $\delta = 10^{-3}$, and in this example, the EM algorithm only needs three iterations to converge. Figure 2(d)–(f) show the estimated density at each iteration. The final estimates are

$$\mu_1^{(3)} = \begin{bmatrix} 0.0806 \\ 3.9445 \end{bmatrix}, \quad \mu_1^{(3)} = \begin{bmatrix} -2.0181 \\ -0.1740 \end{bmatrix}, \quad \Sigma_1^{(3)} = \begin{bmatrix} 2.7452 & 0.0568 \\ 0.0568 & 0.4821 \end{bmatrix}, \quad \Sigma_2^{(3)} = \begin{bmatrix} 0.8750 & -0.0153 \\ -0.0153 & 1.7935 \end{bmatrix},$$

and $w_1^{(3)} = 0.5966$ and $w_2^{(3)} = 0.4034$. These estimates are close to the true ones as can be confirmed by visually comparing Figure 2(f) with Figure 2(a).

6.5 Singularity Problem in Using EM for GMM Fitting

The EM algorithm does well in the previous example, but sometimes it can fail by approaching singularities of the log-likelihood function, especially when the number of components k is large. This is an inherent problem with applying maximum likelihood estimation to GMM due to the fact that the log-likelihood function $L(\theta)$ is not bounded above. For example, let $\mu_1 = y_1$, $\Sigma_1 = \sigma_1^2 I_d$ and $0 < w_1 < 1$, and we have

$$\begin{aligned} L(\theta) &= \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} w_j \phi(y_i \,|\, \mu_j, \Sigma_j) \right) \\ &= \log \left(\sum_{j=1}^{k} w_j \phi(y_1 \,|\, \mu_j, \Sigma_j) \right) + \sum_{i=2}^{n} \log \left(\sum_{j=1}^{k} w_j \phi(y_i \,|\, \mu_j, \Sigma_j) \right) \\ &\geq \log \left(w_1 \phi(y_1 \,|\, \mu_1, \Sigma_1) \right) + \sum_{i=2}^{n} \log \left(\sum_{j=2}^{k} w_j \phi(y_i \,|\, \mu_j, \Sigma_j) \right) \\ &= \log \left(w_1 \phi(y_1 \,|\, y_1, \sigma_1^2 I_d) \right) + \sum_{i=2}^{n} \log \left(\sum_{j=2}^{k} w_j \phi(y_i \,|\, \mu_j, \Sigma_j) \right) \\ &= \log w_1 - \frac{d}{2} \log(2\pi) - \frac{d}{2} \log \sigma_1^2 + \sum_{i=2}^{n} \log \left(\sum_{j=2}^{k} w_j \phi(y_i \,|\, \mu_j, \Sigma_j) \right). \end{aligned}$$

If we let $\sigma_1^2 \to 0$ and keep everything else fixed, then the above lower bound of $L(\theta)$ diverges to infinity and thus $L(\theta) \to \infty$. So for GMM, maximizing the likelihood is an ill-posed problem. However, heuristically, we can still find meaningful solutions at finite local maxima of the log-likelihood function. In order to avoid such singularities when applying the EM algorithm, one can resort to ad hoc techniques such as reinitializing the algorithm after detecting that one component is "collapsing" onto a data sample; one can also adopt the Bayesian approach (discussed in Section 7) as a more principled way to deal with this problem.

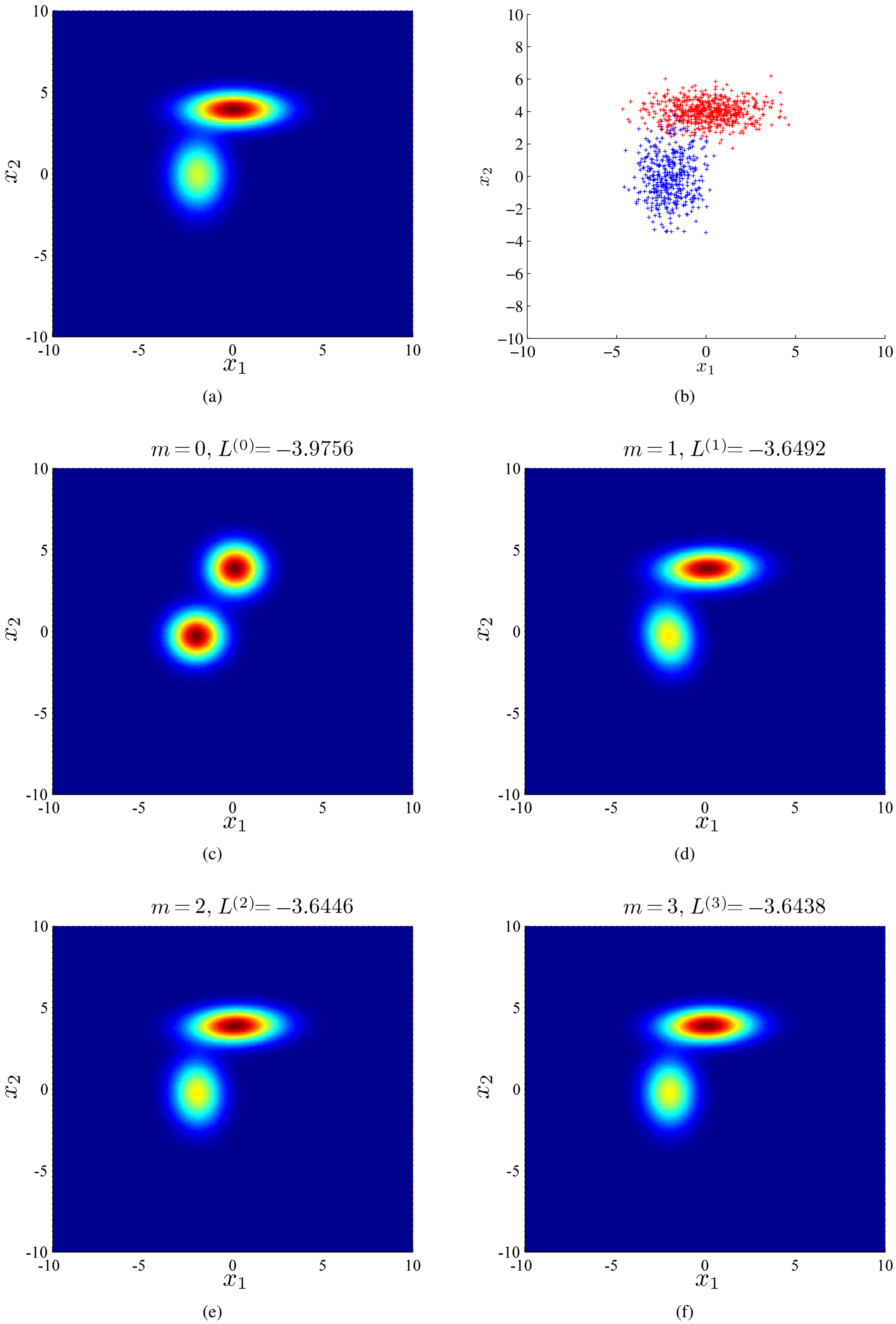


Figure 2: GMM fitting example in Section 6.4: (a) shows the density of a 2-component GMM in $\mathbb{R}^2$; (b) shows 1000 i.i.d. samples from this distribution; (c)–(f) show the estimated density at each iteraton.

7 Maximum A Posteriori

In maximum *a posteriori* (MAP) estimation, one tries to maximize the posterior instead of the likelihood, so we can write the MAP estimator of θ as

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta\in\Theta} \log p(\theta \mid y) = \arg\max_{\theta\in\Theta} \left(\log p(y \mid \theta) + \log p(\theta)\right) = \arg\max_{\theta\in\Theta} \left(L(\theta) + \log p(\theta)\right),$$

where $p(\theta)$ is a prior probability distribution of θ. By modifying the M-step, the EM algorithm can be easily extended for MAP estimation:

E-step: Given the estimate from the previous iteration $\theta^{(m)}$, compute for $\theta \in \Theta$ the conditional expectation

$$Q(\theta \mid \theta^{(m)}) = E_{X|y,\theta^{(m)}} \left[\log p(X \mid \theta)\right].$$

M-step: Maximize $Q(\theta \mid \theta^{(m)}) + \log p(\theta)$ over $\theta \in \Theta$ to find

$$\theta^{(m+1)} = \arg\max_{\theta\in\Theta} \Big(Q(\theta \mid \theta^{(m)}) + \log p(\theta)\Big).$$

Again we have the following theorem to show the monotonicity of the modified EM algorithm:

Theorem 7.1. *Let $L(\theta) = \log p(y \mid \theta)$ be the log-likelihood function and $p(\theta)$ a prior probability distribution of θ on Θ. For $\theta \in \Theta$, if*

$$Q(\theta \mid \theta^{(m)}) + \log p(\theta) \geq Q(\theta^{(m)} \mid \theta^{(m)}) + \log p(\theta^{(m)}), \tag{7.1}$$

then

$$L(\theta) + \log p(\theta) \geq L(\theta^{(m)}) + \log p(\theta^{(m)}).$$

Proof. Add $\log p(\theta)$ to both sides of (4.2), and we have

$$L(\theta) + \log p(\theta) \geq Q(\theta \mid \theta^{(m)}) + \log p(\theta) + h(X \mid y, \theta^{(m)}). \tag{7.2}$$

Similarly, by adding $\log p(\theta^{(m)})$ to both sides of (4.3), we have

$$L(\theta^{(m)}) + \log p(\theta^{(m)}) = Q(\theta^{(m)} \mid \theta^{(m)}) + \log p(\theta^{(m)}) + h(X \mid y, \theta^{(m)}). \tag{7.3}$$

If (7.1) holds, then we can combine (7.2) and (7.3), and have

$$\begin{aligned} L(\theta) + \log p(\theta) &\geq Q(\theta \mid \theta^{(m)}) + \log p(\theta) + h(X \mid y, \theta^{(m)}) \\ &\geq Q(\theta^{(m)} \mid \theta^{(m)}) + \log p(\theta^{(m)}) + h(X \mid y, \theta^{(m)}) \\ &= L(\theta^{(m)}) + \log p(\theta^{(m)}), \end{aligned}$$

which ends the proof. □

In Section 6.5, we mentioned that the EM algorithm might fail at finding meaningful parameters for GMM due to the singularities of the log-likelihood function. However, when using the extended EM algorithm for MAP estimation, one can choose an appropriate prior $p(\theta)$ to avoid such singularities. We refer the reader to [11] for details.

8 Last Words and Acknowledgements

EM is a handy tool, but please use it responsibly. Keep in mind its limitations, and always check the concavity of your log-likelihood; if your log-likelihood isn't concave, don't trust one run of EM to find the optimal solution. Non-concavity can happen to you! In fact, the most popular applications of EM, such as GMM and HMM, are usually *not* concave, and can benefit from multiple initializations. For non-concave likelihood functions, it might be helpful to use EM in conjunction with a global optimizer designed to explore the space more randomly: the global optimizer provides the exploration strategy while EM does the actual local searches. For more on state-of-the-art global optimization, see for example [12, 13, 14, 15, 16, 17].

This tutorial was supported in part by the United States Office of Naval Research. We thank the following people for their suggested edits and proofreading: Ji Cao, Sergey Feldman, Bela Frigyik, Eric Garcia, Adam Gustafson, Amol Kapila, Nicole Nichols, Mikyoung Park, Nathan Parrish, Tien Re, Eric Swanson, and Kristi Tsukida.

References

[1] A. P. Dempster, N. M. Laird, and D. B. 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.

[2] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, pp. 257–286, Feb. 1989.

[3] J. A. Bilmes, “A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models,” Tech. Rep. TR-97-021, International Computer Science Institute, April 1998.

[4] C. F. J. Wu, “On the convergence properties of the EM algorithm,” *The Annals of Statistics*, vol. 11, pp. 95–103, March 1983.

[5] R. A. Boyles, “On the convergence of the EM algorithm,” *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 45, no. 1, pp. 47–50, 1983.

[6] R. A. Redner and H. F. Walker, “Mixture densities, maximum likelihood and the EM algorithm,” *SIAM Review*, vol. 26, pp. 195–239, April 1984.

[7] A. Roche, “EM algorithm and variants: An informal tutorial.” Unpublished (available online at `ftp://ftp.cea.fr/pub/dsv/madic/publis/Roche_em.pdf`), 2003.

[8] R. M. Neal and G. E. Hinton, “A view of the EM algorithm that justifies incremental, sparse, and other variants,” in *Learning in Graphical Models* (M. I. Jordan, ed.), MIT Press, Nov. 1998.

[9] K. B. Petersen and M. S. Pedersen, “The matrix cookbook,” Nov. 2008. `http://matrixcookbook.com/`.

[10] L. Xu and M. I. Jordan, “On convergence properties of the EM algorithm for Gaussian mixtures,” *Neural Computation*, vol. 8, pp. 129–151, Jan. 1996.

[11] C. Fraley and A. E. Raftery, “Bayesian regularization for normal mixture estimation and model-based clustering,” *Journal of Classification*, vol. 24, pp. 155–181, Sept. 2007.

[12] D. R. Jones, “A taxonomy of global optimization methods based on response surfaces,” *Journal of Global Optimization*, vol. 21, pp. 345–383, Dec. 2001.

[13] R. Mendes, J. Kennedy, and J. Neves, “The fully informed particle swarm: simpler, maybe better,” *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 204–210, June 2004.

[14] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky, “A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems,” *Journal of Global Optimization*, vol. 31, pp. 635–672, April 2005.

[15] C. Khompatraporn, J. D. Pintér, and Z. B. Zabinsky, “Comparative assessment of algorithms and software for global optimization,” *Journal of Global Optimization*, vol. 31, pp. 613–633, April 2005.

[16] M. Hazen and M. R. Gupta, “A multiresolutional estimated gradient architecture for global optimization,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 3013–3020, 2006.

[17] M. Hazen and M. R. Gupta, “Gradient estimation in global optimization algorithms,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1841–1848, 2009.