Foundations and Trends® in
Signal Processing
Vol. 4, No. 3 (2010) 223–296

DOI: 10.1561/2000000034

now
the essence of knowledge

Theory and Use of the EM Algorithm

By Maya R. Gupta and Yihua Chen

Contents

Foundations and Trends® in
Signal Processing
Vol. 4, No. 3 (2010) 223–296

DOI: 10.1561/2000000034

now
the essence of knowledge

Theory and Use of the EM Algorithm

Maya R. Gupta[1] and Yihua Chen[2]

[1] *Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA, gupta@ee.washington.edu*
[2] *Department of Electrical Engineering, University of Washington, Seattle, WA 98195, USA, yhchen@ee.washington.edu*

Abstract

This introduction to the expectation–maximization (EM) algorithm provides an intuitive and mathematically rigorous understanding of EM. Two of the most popular applications of EM are described in detail: estimating Gaussian mixture models (GMMs), and estimating hidden Markov models (HMMs). EM solutions are also derived for learning an optimal mixture of fixed models, for estimating the parameters of a compound Dirichlet distribution, and for dis-entangling superimposed signals. Practical issues that arise in the use of EM are discussed, as well as variants of the algorithm that help deal with these challenges.

1

The Expectation-Maximization Method

Expectation–maximization (EM) is an iterative method that attempts to find the maximum likelihood estimator of a parameter θ of a parametric probability distribution. Let us begin with an example. Consider the temperature outside your window for each of the 24 hours of a day, represented by $x \in \mathbb{R}^{24}$, and say that this temperature depends on the season $\theta \in \{\text{summer, fall, winter, spring}\}$, and that you know the seasonal temperature distribution $p(x\,|\,\theta)$. But what if you could only measure the average temperature $y = \bar{x}$ for some day, and you would like to estimate what season θ it is (for example, is spring here yet?). In particular, you might seek the maximum likelihood estimate of θ, that is, the value $\hat{\theta}$ that maximizes $p(y\,|\,\theta)$. If this is not a trivial maximum likelihood problem, you might call upon EM. EM iteratively alternates between making guesses about the complete data x, and finding 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 will see in later sections that EM does not actually promise to find 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.

This exposition is designed to be useful to both the EM novice and the experienced EM user looking to better understand the method and its use. To this end, we err on the side of providing too many explicit details rather than too few.

First, we go over the steps of EM, breaking down the usual two-step description into a five-step description. Table 1.1 summarizes the key notation. We recommend reading this document linearly up through Section 1.4, after which sections can generally be read out-of-order. Section 1 ends with a detailed version of a historical toy example for EM. In Section 2 we show that EM never gets worse as it iterates in terms of the likelihood of the estimate it produces, and we explain the *maximization–maximization* interpretation of EM. We also explain the general advantages and disadvantages of EM compared to other options for maximizing the likelihood, like the Newton–Raphson method. The

Table 1.1. Notation summary.

$\mathbb{R}$	Set of real numbers
$\mathbb{R}_+$	Set of positive real numbers
$\mathbb{N}$	Set of natural numbers
$y \in \mathbb{R}^d$	Given measurement or observation
$Y \in \mathbb{R}^d$	Random measurement; y is a realization of Y
$x \in \mathbb{R}^{d_1}$	Complete data you wish you had
$X \in \mathbb{R}^{d_1}$	Random complete data; x is a realization of X
$z \in \mathbb{R}^{d_2}$	Missing data; in some problems $x = (y, z)$
$Z \in \mathbb{R}^{d_2}$	Random missing data; z is a realization of Z
$\theta \in \Omega$	Parameter(s) to estimate, Ω is the parameter space
$\theta^{(m)} \in \Omega$	mth estimate of θ
$p(y\,\vert\,\theta)$	Density of y given θ; also written as $p(Y = y\,\vert\,\theta)$
$\mathcal{X}$	Support of X (closure of the set of x where $p(x\,\vert\,\theta) > 0$)
$\mathcal{X}(y)$	Support of X conditioned on y (closure of the set of x where $p(x\,\vert\,y,\theta) > 0$)
$\triangleq$	"Is defined to be"
$L(\theta)$	Likelihood of θ given y, that is, $p(y\,\vert\,\theta)$
$\ell(\theta)$	Log-likelihood of θ given y, that is, $\log p(y\,\vert\,\theta)$
$E_{X\vert y,\theta}[X]$	Expectation of X conditioned on y and θ, that is, $\int_{\mathcal{X}(y)} x p(x\,\vert\,y,\theta) dx$
$1_{\{\cdot\}}$	Indicator function: equals 1 if the expression $\{\cdot\}$ is true, and 0 otherwise
$\mathbf{1}$	Vector of ones
$D_{\mathrm{KL}}(P \,\Vert\, Q)$	Kullback–Leibler divergence (a.k.a. relative entropy) between distributions P and Q

advantages of EM are made clearer in Sections 3 and 4, in which we derive a number of popular applications of EM and use these applications to illustrate practical issues that can arise with EM. Section 3 covers learning the optimal combination of fixed models to explain the observed data, and fitting a Gaussian mixture model (GMM) to the data. Section 4 covers learning hidden Markov models (HMMs), separating superimposed signals, and estimating the parameter for the compound Dirichlet distribution. In Section 5, we categorize and discuss some of the variants of EM and related methods, and we conclude this manuscript in Section 6 with some historical notes.

1.1 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 $p(x\,|\,\theta)$.[1] In Sections 3 and 4 we will explain how to define the complete data x for some standard EM applications.

We assume that the complete data can be modeled as a continuous[2] random vector X with density $p(x\,|\,\theta)$,[3] where $\theta \in \Omega$ for some set Ω. You do not observe X directly; instead, you observe a realization y of the random vector Y that depends[4] on X. For example, X might be a random vector and Y the mean of its components, or if X is a complex number then Y might be only its magnitude, or Y might be the first component of the vector X.

[1] A different standard choice of notation for a parametric density would be $p(y;\theta)$, but we prefer $p(y\,|\,\theta)$ because this notation is clearer when one wants to find the maximum *a posteriori* estimate rather than the maximum likelihood estimate—we will talk more about the maximum *a posteriori* estimate of θ in Section 1.3.

[2] The treatment of discrete random vectors is a straightforward special case of the continuous treatment: one only needs to replace the probability density function with probability mass function and integral with summation.

[3] We assume that the support of X, denoted by $\mathcal{X}$, which is the closure of the set $\{x \mid p(x\,|\,\theta) > 0\}$, does not depend on θ. An example where the support does depend on θ is if X is uniformly distributed on the interval $[0,\theta]$. If the support does depend on θ, then the monotonicity of the EM algorithm might not hold. See Section 2.1 for details.

[4] A rigorous description of this dependency is deferred to Section 1.4.

Given that you only have y, the goal here is to find the maximum likelihood estimate (MLE) of θ:

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \Omega} p(y \,|\, \theta). \tag{1.1}$$

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

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \Omega} \log p(y \,|\, \theta). \tag{1.2}$$

Because log is a monotonically increasing function, the solution to (1.1) will be the same as the solution to (1.2). However, for some problems it is difficult to solve either (1.1) or (1.2). 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 an estimate 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 let us first break it down into five steps:

Step 1: Let $m = 0$ and make an initial estimate $\theta^{(m)}$ for θ.

Step 2: Given the observed data y and pretending for the moment that your current guess $\theta^{(m)}$ is correct, formulate the conditional probability distribution $p(x \,|\, y, \theta^{(m)})$ for the complete data x.

Step 3: Using the conditional probability distribution $p(x \,|\, y, \theta^{(m)})$ calculated in Step 2, form the *conditional expected log-likelihood*, which is called the Q-function[5]:

$$\begin{aligned} Q(\theta \,|\, \theta^{(m)}) &= \int_{\mathcal{X}(y)} \log p(x \,|\, \theta) p(x \,|\, y, \theta^{(m)}) dx \\ &= E_{X|y,\theta^{(m)}}[\log p(X \,|\, \theta)], \end{aligned} \tag{1.3}$$

[5] Note this Q-function has nothing to do with the sum of the tail of a Gaussian, which is also called the Q-function. People call (1.3) the Q-function because the original paper [11] used a Q to notate it. We like to say that the Q stands for *quixotic* because it is a bit crazy and hopeful and beautiful to think you can find the maximum likelihood estimate of θ in this way that iterates round-and-round like a windmill, and if Don Quixote had been a statistician, it is just the sort of thing he might have done.

where the integral is over the set $\mathcal{X}(y)$, which is the closure of the set $\{x \mid p(x\,|\,y,\theta) > 0\}$, and we assume that $\mathcal{X}(y)$ does not depend on θ.

Note that θ is a free variable in (1.3), so the Q-function is a function of θ, but also depends on your current guess $\theta^{(m)}$ implicitly through the $p(x\,|\,y,\theta^{(m)})$ calculated in Step 2.

Step 4: Find the θ that maximizes the Q-function (1.3); the result is your new estimate $\theta^{(m+1)}$.

Step 5: Let $m := m + 1$ and go back to Step 2. (The EM algorithm does not specify a stopping criterion; standard criteria are to iterate until the estimate stops changing: $\|\theta^{(m+1)} - \theta^{(m)}\| < \epsilon$ for some $\epsilon > 0$, or to iterate until the log-likelihood $\ell(\theta) = \log p(y\,|\,\theta)$ stops changing: $|\ell(\theta^{(m+1)}) - \ell(\theta^{(m)})| < \epsilon$ for some $\epsilon > 0$.)

The EM estimate is *only guaranteed to never get worse* (see Section 2.1 for details). Usually, it will find a peak in the likelihood $p(y\,|\,\theta)$, but if the likelihood function $p(y\,|\,\theta)$ has multiple peaks, EM will not necessarily find the global maximum 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 and 3 combined are called the *E-step* for *expectation*, and Step 4 is called the *M-step* for *maximization*:

E-step: Given the estimate from the previous iteration $\theta^{(m)}$, compute the conditional expectation $Q(\theta\,|\,\theta^{(m)})$ given in (1.3).

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

$$\theta^{(m+1)} = \arg\max_{\theta\in\Omega} Q(\theta\,|\,\theta^{(m)}). \tag{1.4}$$

Since the E-step is just to compute the Q-function which is used in the M-step, EM can be summarized as just iteratively solving the M-step given by (1.4). When applying EM to a particular problem, this is usually the best way to think about EM because then one does not waste time computing parts of the Q-function that do not depend on θ.

1.2 Contrasting EM with a Simple Variant

As a comparison that may help illuminate EM, we next consider a simple variant of EM. In Step 2 above, one computes the conditional distribution $p(x \,|\, y, \theta^{(m)})$ over all possible values of x, and this entire conditional distribution is taken into account in the M-step. A simple variant is to instead use only the mth maximum likelihood estimate $x^{(m)}$ of the complete data x:

$$\text{E-like-step:} \qquad x^{(m)} = \arg\max_{x \in \mathcal{X}(y)} p(x \,|\, y, \theta^{(m)}),$$

$$\text{M-like-step:} \qquad \theta^{(m+1)} = \arg\max_{\theta \in \Omega} p(x^{(m)} \,|\, \theta).$$

We call this variant the *point-estimate variant of EM*; it has also been called *classification EM*. More on this variant can be found in [7, 9].

Perhaps the most famous example of this variant is *k-means clustering*[6] [21, 35]. In k-means clustering, we have n observed data points $y = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}^{\mathrm{T}}$, where each $y_i \in \mathbb{R}^d$, and it is believed that the data points belong to k clusters. Let the complete data be the observed data points and the missing information that specifies which of the k clusters each observed data point belongs to. The goal is to estimate the k cluster centers θ. First, one makes an initial guess $\hat{\theta}^0$ of the k cluster centers. Then in the E-like step, one assigns each of the n points to the closest cluster based on the estimated cluster centers $\theta^{(m)}$. Then in the M-like step, one takes all the points assigned to each cluster, and computes the mean of those points to form a new estimate of the cluster's centroid. Underlying k-means is a model that the clusters are defined by Gaussian distributions with unknown means (the θ to be estimated) and identity covariance matrices.

EM clustering differs from k-means clustering in that at each iteration you do not choose a single $x^{(m)}$, that is, one does not force each observed point y_i to belong to only one cluster. Instead, each observed point y_i is probabilistically assigned to the k clusters by estimating $p(x \,|\, y, \theta^{(m)})$. We treat EM clustering in more depth in Section 3.2.

[6] The k-means clustering algorithm dates to 1967 [35] and is a special case of *vector quantization*, which was first proposed as Lloyd's algorithm in 1957 [32]. See [17] for details.

1.3 Using a Prior with EM (MAP EM)

The EM algorithm can fail due to singularities of the log-likelihood function — for example, for learning a GMM with 10 components, it may decide that the most likely solution is for one of the Gaussians to only have one data point assigned to it, with the bad result that the Gaussian is estimated as having zero covariance (see Section 3.2.5 for details).

A straightforward solution to such degeneracies is to take into account or impose some prior information on the solution for θ. One approach would be to restrict the set of possible θ. Such a restriction is equivalent to putting a uniform prior probability over the restricted set. More generally, one can impose any prior $p(\theta)$, and then modify EM to maximize the posterior rather than the likelihood:

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta\in\Omega} \log p(\theta\,|\,y) = \arg\max_{\theta\in\Omega} (\log p(y\,|\,\theta) + \log p(\theta)).$$

The EM algorithm is easily extended to maximum *a posteriori* (MAP) estimation by modifying the M-step:
E-step: Given the estimate from the previous iteration $\theta^{(m)}$, compute as a function of $\theta \in \Omega$ the conditional expectation

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

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

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

An example of MAP EM is given in Section 3.3.

1.4 Specifying the Complete Data

Practically, the complete data should be defined so that given x it is relatively easy to maximize $p(x\,|\,\theta)$ with respect to θ. Theoretically, the complete data X must satisfy the Markov relationship $\theta \to X \to Y$ with respect to the parameter θ and the observed data Y, that is, it must be that

$$p(y\,|\,x,\theta) = p(y\,|\,x).$$

A special case is when Y is a function of X, that is, $Y = T(X)$; in this case, $X \to Y$ is a deterministic function, and thus the Markov relationship always holds.

1.4.1 EM for Missing Data Problems

For many applications of EM, including GMM and HMM, the complete data X is the observed data Y plus some missing (sometimes called *latent* or *hidden*) data Z, such that $X = (Y, Z)$. This is a special case of $Y = T(X)$, where the function T simply removes Z from X to produce Y. In general when using EM with missing data, one can write the Q-function as an integral over the domain of Z, denoted by $\mathcal{Z}$, rather than over the domain of X, because the only random part of the complete data X is the missing data Z. Then, for missing data problems where $x = (y, z)$,

$$\begin{aligned} Q(\theta \,|\, \theta^{(m)}) &= \int_{\mathcal{X}} \log p(x \,|\, \theta) p(x \,|\, y, \theta^{(m)}) dx \\ &= \int_{\mathcal{X}} \log p(y, z \,|\, \theta) p(y, z \,|\, y, \theta^{(m)}) dx \\ &= \int_{\mathcal{Z}} \log p(y, z \,|\, \theta) p(z \,|\, y, \theta^{(m)}) dz \\ &= E_{Z|y,\theta^{(m)}}[\log p(y, Z \,|\, \theta)]. \end{aligned} \tag{1.5}$$

1.4.2 EM for Independently, Identically Distributed Samples

For many common applications such as learning a GMM or HMM, the complete data X is a set of n independent and identically distributed (i.i.d.) random vectors, $X = \begin{bmatrix} X_1 & X_2 & \dots & X_n \end{bmatrix}^{\mathrm{T}}$ and the ith observed sample y_i is only a function of x_i. Then the following proposition is useful for decomposing the Q-function into a sum:

Proposition 1.1. Suppose $p(x \,|\, \theta) = \prod_{i=1}^{n} p(x_i \,|\, \theta)$ for all $x \in \mathcal{X}^n$ and all $\theta \in \Omega$, and the Markov relationship $\theta \to X_i \to Y_i$ holds for all $i = 1, \dots, n$, that is,

$$p(y_i \,|\, x, y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n, \theta) = p(y_i \,|\, x_i), \tag{1.6}$$

then

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

where

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

Proof. First, we show that given θ, the elements of the set $\{(X_i, Y_i)\}$, $i = 1,\ldots,n$, are mutually independent, that is,

$$p(x, y\,|\,\theta) = \prod_{i=1}^{n} p(x_i, y_i\,|\,\theta). \tag{1.7}$$

This mutual independence holds because

$$\begin{aligned}
p(x, y\,|\,\theta) &= p(y_1\,|\,y_2,\ldots,y_n,x,\theta)\cdots p(y_n\,|\,x,\theta)p(x\,|\,\theta) \\
&\quad \text{(by the chain rule)} \\
&= p(y_1\,|\,x_1,\theta)\cdots p(y_n\,|\,x_n,\theta)p(x\,|\,\theta) \\
&\quad \text{(by (1.6), but keep } \theta \text{ in the condition)} \\
&= p(y_1\,|\,x_1,\theta)\cdots p(y_n\,|\,x_n,\theta)\prod_{i=1}^{n} p(x_i\,|\,\theta) \\
&\quad \text{(by the independence assumption on } X) \\
&= \prod_{i=1}^{n} p(y_i\,|\,x_i,\theta)p(x_i\,|\,\theta) \\
&= \prod_{i=1}^{n} p(x_i, y_i\,|\,\theta).
\end{aligned}$$

Then we show that for all $i = 1,\ldots,n$, we have

$$p(x_i\,|\,y,\theta) = p(x_i\,|\,y_i,\theta). \tag{1.8}$$

This is because

$$\begin{aligned}
p(x_i\,|\,y,\theta) &= \frac{p(x_i, y\,|\,\theta)}{p(y\,|\,\theta)} \\
&\quad \text{(by Bayes' rule)} \\
&= \frac{\int_{\mathcal{X}^{n-1}} p(x, y\,|\,\theta)dx_1\ldots dx_{i-1}dx_{i+1}\ldots dx_n}{\int_{\mathcal{X}^n} p(x, y\,|\,\theta)dx}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\int_{\mathcal{X}^{n-1}} \prod_{j=1}^{n} p(x_j, y_j \,|\, \theta) dx_1 \ldots dx_{i-1} dx_{i+1} \ldots dx_n}{\int_{\mathcal{X}^n} \prod_{j=1}^{n} p(x_j, y_j \,|\, \theta) dx_1 \ldots dx_n} \\
&\quad \text{(by (1.7))} \\
&= \frac{p(x_i, y_i \,|\, \theta) \prod_{j=1, j\neq i}^{n} \int_{\mathcal{X}} p(x_j, y_j \,|\, \theta) dx_j}{\prod_{j=1}^{n} \int_{\mathcal{X}} p(x_j, y_j \,|\, \theta) dx_j} \\
&= \frac{p(x_i, y_i \,|\, \theta) \prod_{j=1, j\neq i}^{n} p(y_j \,|\, \theta)}{\prod_{j=1}^{n} p(y_j \,|\, \theta)} \\
&= \frac{p(x_i, y_i \,|\, \theta)}{p(y_i \,|\, \theta)} \\
&= p(x_i \,|\, y_i, \theta).
\end{aligned}
$$

Then,

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

where the last line holds because of (1.8). □

1.5 A Toy Example

We next present a fully worked-out version of a "toy example" of EM that was used in the seminal EM paper [11]. Here, we give more details, and we have changed it to literally be a toy example.

Imagine you ask n kids to choose a toy out of four choices. Let $Y = \begin{bmatrix} Y_1 & \ldots & Y_4 \end{bmatrix}^{\mathrm{T}}$ denote the histogram of their n choices, where Y_i is the number of the kids that chose toy i, for $i = 1, \ldots, 4$. We can model this

random histogram Y as being distributed according to a multinomial distribution. The multinomial has two parameters: the *number of kids asked*, denoted by $n \in \mathbb{N}$, and the *probability that a kid will choose each of the four toys*, denoted by $p \in [0,1]^4$, where $p_1 + p_2 + p_3 + p_4 = 1$. Then the probability of seeing some particular histogram y is:

$$P(y \,|\, p) = \frac{n!}{y_1! y_2! y_3! y_4!} p_1^{y_1} p_2^{y_2} p_3^{y_3} p_4^{y_4}. \tag{1.9}$$

Next, say that we have reason to believe that the unknown probability p of choosing each of the toys is parameterized by some hidden value $\theta \in (0,1)$ such that

$$p_\theta = \begin{bmatrix} \frac{1}{2} + \frac{1}{4}\theta & \frac{1}{4}(1-\theta) & \frac{1}{4}(1-\theta) & \frac{1}{4}\theta \end{bmatrix}^{\mathrm{T}}, \quad \theta \in (0,1). \tag{1.10}$$

The estimation problem is to guess the θ that maximizes the probability of the observed histogram y of toy choices.

Combining (1.9) and (1.10), we can write the probability of seeing the histogram $y = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 \end{bmatrix}^{\mathrm{T}}$ as

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

For this simple example, one could directly maximize the log-likelihood $\log P(y \,|\, \theta)$, 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 $X = \begin{bmatrix} X_1 & \dots & X_5 \end{bmatrix}^{\mathrm{T}}$, where X has a multinomial distribution with number of trials n and the probability of each event is:

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

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

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

The likelihood of a realization x of the complete data is

$$P(x\,|\,\theta) = \frac{n!}{\prod_{i=1}^{5} x_i!}\left(\frac{1}{2}\right)^{x_1}\left(\frac{\theta}{4}\right)^{x_2+x_5}\left(\frac{1-\theta}{4}\right)^{x_3+x_4}. \tag{1.11}$$

For EM, we need to maximize the Q-function:

$$\theta^{(m+1)} = \arg\max_{\theta\in(0,1)} Q(\theta\,|\,\theta^{(m)}) = \arg\max_{\theta\in(0,1)} E_{X|y,\theta^{(m)}}[\log p(X\,|\,\theta)].$$

To solve the above equation, we actually only need the terms of $\log p(x\,|\,\theta)$ that depend on θ, because the other terms are irrelevant as far as maximizing over θ is concerned. Take the log of (1.11) and ignore those terms that do not depend on θ, then

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

To solve the above maximization problem, we need the expectation of the complete data X conditioned on the already known incomplete data y, which only leaves the uncertainty about X_1 and X_2. Since we know that $X_1 + X_2 = y_1$, we can use the indicator function $1_{\{\cdot\}}$ to write that given y_1, the pair (X_1, X_2) is binomially distributed with X_1 "successes" in y_1 events:

$$\begin{aligned}&P(x\,|\,y,\theta^{(m)})\\ &\quad= \frac{y_1!}{x_1!x_2!}\left(\frac{\frac{1}{2}}{\frac{1}{2}+\frac{\theta^{(m)}}{4}}\right)^{x_1}\left(\frac{\frac{\theta^{(m)}}{4}}{\frac{1}{2}+\frac{\theta^{(m)}}{4}}\right)^{x_2} 1_{\{x_1+x_2=y_1\}}\prod_{i=3}^{5} 1_{\{x_i=y_{i-1}\}}\\ &\quad= \frac{y_1!}{x_1!x_2!}\left(\frac{2}{2+\theta^{(m)}}\right)^{x_1}\left(\frac{\theta^{(m)}}{2+\theta^{(m)}}\right)^{x_2} 1_{\{x_1+x_2=y_1\}}\prod_{i=3}^{5} 1_{\{x_i=y_{i-1}\}}.\end{aligned}$$

Then the conditional expectation of X given y and $\theta^{(m)}$ is

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

and the M-step becomes

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

Given an initial estimate $\theta^{(0)} = 0.5$, the above algorithm reaches $\hat{\theta}_{\text{MLE}}$ to MATLAB's numerical precision on the 18th iteration.

2

Analysis of EM

How good are the estimates produced by EM? How exactly is the Q-function related to the true log-likelihood? In this section we discuss EM convergence, show that the Q-function provides a lower bound to the true log-likelihood, and describe the maximization–maximization interpretation of EM.

2.1 Convergence

Here is what can be proved without extra conditions: 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 results from the following theorem, which states that improving the Q-function will at least not make the log-likelihood $\ell(\theta)$ worse:

Theorem 2.1. Let random variables X and Y have parametric densities with parameter $\theta \in \Omega$. Suppose the support of X does not depend on θ, and the Markov relationship $\theta \to X \to Y$, that is,

$$p(y\,|\,x,\theta) = p(y\,|\,x) \tag{2.1}$$

holds for all $\theta \in \Omega$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Then for $\theta \in \Omega$ and any $y \in \mathcal{Y}$ with $\mathcal{X}(y) \neq \emptyset$, $\ell(\theta) \geq \ell(\theta^{(m)})$ if $Q(\theta \,|\, \theta^{(m)}) \geq Q(\theta^{(m)} \,|\, \theta^{(m)})$.

We first discuss the theorem, then prove it later in Section 2.1.1. For the EM algorithm, the M-step ensures that

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

and hence it must be that $Q(\theta^{(m+1)} \,|\, \theta^{(m)}) \geq Q(\theta^{(m)} \,|\, \theta^{(m)})$. Therefore we can apply Theorem 2.1 and conclude that $\ell(\theta^{(m+1)}) \geq \ell(\theta^{(m)})$.

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 $\{\theta^{(m)}\}$.[1] Indeed, *there is no general convergence theorem for the EM algorithm*[2]: the convergence of the sequence $\{\theta^{(m)}\}$ depends on the characteristics of $\ell(\theta)$ and $Q(\theta \,|\, \theta')$, and also the starting point $\theta^{(0)}$.

Under certain regularity conditions, one can prove that $\{\theta^{(m)}\}$ converges to a stationary point (for example, a local maximum or saddle point) of $\ell(\theta)$. However, this convergence is only linear.[3] Instead of using the EM algorithm one could (locally) maximize the likelihood using Newton–Raphson updates, which requires calculating the inverse of the Hessian matrix, but has quadratic convergence.[4] Superlinear convergence[5] could instead be achieved using conjugate gradient methods or quasi-Newton updates such as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) update, which only require computing the gradient of the log-likelihood [27, 45]. The Newton–Raphson method can be expected to hone in on $\theta^\star$ fast once $\theta^{(m)}$ is close, but EM may be more effective given a poor initial guess, in part because the Hessian matrix for the Newton–Raphson method may not be positive definite and hence makes the inversion unstable.

---

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

[2] Theorem 2 in [11] appears to be a general convergence theorem for EM; however, its proof is flawed as pointed out in [6, 62].

[3] Linear convergence means that there exist $M > 0$ and $0 < C < 1$ such that $\|\theta^{(m+1)} - \theta^\star\| \leq C\|\theta^{(m)} - \theta^\star\|$ for all $m \geq M$, where $\theta^\star$ is the optimal value of θ.

[4] Quadratic convergence means that there exist $M > 0$ and $0 < C < 1$ such that $\|\theta^{(m+1)} - \theta^\star\| \leq C\|\theta^{(m)} - \theta^\star\|^2$ for all $m \geq M$.

[5] Superlinear convergence means $\|\theta^{(m+1)} - \theta^\star\| / \|\theta^{(m)} - \theta^\star\| \to 0$ as $m \to \infty$.

See [62] for a detailed discussion on EM convergence; other discussions on EM convergence can be found in [6, 11, 50], and [39] specifically addresses the rate of convergence of the EM algorithm. For an analysis of the convergence of the EM algorithm for fitting GMMs, see [63]. Note that some authors (such as [50]) use the term *global convergence* to refer to *convergence to a local maximum from almost any starting point*, and not to imply that one will actually converge to the global maximum.

2.1.1 Proof of the Monotonicity Theorem

Next, we prove Theorem 2.1.

Proof. We first derive a lower bound on the log-likelihood function $\ell(\theta)$:

$$
\begin{aligned}
\ell(\theta) &= \log p(y \,|\, \theta) \\
&\quad \text{(by definition)} \\
&= \log \int_{\mathcal{X}(y)} p(x, y \,|\, \theta) dx \\
&\quad \text{(by the law of total probability)} \\
&= \log \int_{\mathcal{X}(y)} \frac{p(x, y \,|\, \theta)}{p(x \,|\, y, \theta^{(m)})} p(x \,|\, y, \theta^{(m)}) dx \qquad (2.2) \\
&\quad \text{(multiply the top and bottom by the same factor)} \\
&= \log E_{X|y,\theta^{(m)}} \left[\frac{p(X, y \,|\, \theta)}{p(X \,|\, y, \theta^{(m)})} \right] \\
&\quad \text{(rewrite the integral as an expectation)} \\
&\geq E_{X|y,\theta^{(m)}} \left[\log \frac{p(X, y \,|\, \theta)}{p(X \,|\, y, \theta^{(m)})} \right] \\
&\quad \text{(by Jensen's inequality)} \\
&= E_{X|y,\theta^{(m)}} \left[\log \frac{p(X \,|\, \theta) p(y \,|\, X)}{p(X \,|\, \theta^{(m)}) p(y \,|\, X) / p(y \,|\, \theta^{(m)})} \right] \\
&\quad \text{(by Bayes' rule and the assumed Markov relationship)} \\
&= E_{X|y,\theta^{(m)}} \left[\log \frac{p(X \,|\, \theta) p(y \,|\, \theta^{(m)})}{p(X \,|\, \theta^{(m)})} \right]
\end{aligned}
$$

$$
\begin{aligned}
&= E_{X|y,\theta^{(m)}}[\log p(X\,|\,\theta)] - E_{X|y,\theta^{(m)}}[\log p(X\,|\,\theta^{(m)})] \\
&\quad + \log p(y\,|\,\theta^{(m)}) \\
&= Q(\theta\,|\,\theta^{(m)}) - Q(\theta^{(m)}\,|\,\theta^{(m)}) + \ell(\theta^{(m)}),
\end{aligned} \tag{2.3}
$$

where the Q-function in the last line is defined in (1.3). Note that because of the assumption that the support of X does not depend on θ, combined with the assumed Markov relationship, we can easily conclude that $\mathcal{X}(y)$ does not depend on θ, either, and thus perform the trick in (2.2); otherwise, if $\mathcal{X}(y)$ does depend on θ, (2.2) can lead to $\frac{0}{0}$ and the rest of the proof won't follow.

We can conclude the first part of the proof by restating (2.3) as a lower bound on the log-likelihood function:

$$
\ell(\theta) \geq \ell(\theta^{(m)}) + Q(\theta\,|\,\theta^{(m)}) - Q(\theta^{(m)}\,|\,\theta^{(m)}). \tag{2.4}
$$

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

Next, since we assume that $Q(\theta\,|\,\theta^{(m)}) \geq Q(\theta^{(m)}\,|\,\theta^{(m)})$, we can simply conclude that:

$$
\ell(\theta) \geq \ell(\theta^{(m)}) + (Q(\theta\,|\,\theta^{(m)}) - Q(\theta^{(m)}\,|\,\theta^{(m)})) \geq \ell(\theta^{(m)}),
$$

which completes the proof. □

2.1.2 Monotonicity of MAP EM

The EM algorithm for the MAP estimation given in Section 1.3 also has the monotonicity property:

Theorem 2.2. Let random variables X and Y have parametric densities with parameter $\theta \in \Omega$, where θ is distributed according to the density $p(\theta)$ on Ω. Suppose the support of X does not depend on θ, and the Markov relationship $\theta \to X \to Y$, that is,

$$
p(y\,|\,x,\theta) = p(y\,|\,x)
$$

holds for all $\theta \in \Omega$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Then for $\theta \in \Omega$ and any $y \in \mathcal{Y}$ with $\mathcal{X}(y) \neq \emptyset$,

$$
\ell(\theta) + \log p(\theta) \geq \ell(\theta^{(m)}) + \log p(\theta^{(m)}),
$$

if

$$Q(\theta \,|\, \theta^{(m)}) + \log p(\theta) \geq Q(\theta^{(m)} \,|\, \theta^{(m)}) + \log p(\theta^{(m)}). \tag{2.5}$$

Proof. Add $\log p(\theta)$ to both sides of (2.4), and we have

$$\begin{aligned} \ell(\theta) + \log p(\theta) &\geq \ell(\theta^{(m)}) + Q(\theta \,|\, \theta^{(m)}) - Q(\theta^{(m)} \,|\, \theta^{(m)}) + \log p(\theta) \\ &= \ell(\theta^{(m)}) + \log p(\theta^{(m)}) + Q(\theta \,|\, \theta^{(m)}) + \log p(\theta) \\ &\quad - Q(\theta^{(m)} \,|\, \theta^{(m)}) - \log p(\theta^{(m)}) \\ &\geq \ell(\theta^{(m)}) + \log p(\theta^{(m)}), \end{aligned}$$

where the last line follows from (2.5). □

2.2 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 log-likelihood function $\ell(\theta)$ [41]. Specifically, we will guess that X has distribution $\tilde{P}$ with support $\mathcal{X}(y)$ and density $\tilde{p}(x)$. Let P_θ denote the conditional distribution with density $p(x \,|\, y, \theta)$. Then consider maximizing the following objective function alternately with respect to $\tilde{P}$ and θ:

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

where $D_{\mathrm{KL}}(\tilde{P} \,\|\, P_\theta)$ is the Kullback–Leibler divergence (a.k.a. relative entropy) between the current guess $\tilde{P}$ of the distribution over the complete data, and the likelihood P_θ of the complete data given the parameter θ. Maximizing $F(\tilde{P}, \theta)$ with respect to θ maximizes a lower bound on the log-likelihood function $\ell(\theta)$ since the KL divergence is always nonnegative. Then maximizing $F(\tilde{P}, \theta)$ with respect to $\tilde{P}$ attempts to tighten the lower bound for your current estimate of θ. 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 [41] for details).

Further, this interpretation establishes EM as belonging to the class of methods called *alternating optimization* or *alternating minimization* methods. This class of methods also includes projection onto convex sets (POCS) and the Blahut–Arimoto algorithms; for more details on this class of algorithms, we recommend Stark and Yang's book [55] and Yeung's book [64].

Next we show that this maximization–maximization view truly is the same as the EM algorithm. Formally, 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

$$\tilde{P}^{(m)} = \arg\max_{\tilde{P}} F(\tilde{P}, \theta^{(m-1)}). \tag{2.6}$$

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

$$\theta^{(m)} = \arg\max_{\theta \in \Omega} F(\tilde{P}^{(m)}, \theta). \tag{2.7}$$

First, note that (2.6) can be simplified:

$$\begin{aligned}
\tilde{P}^{(m)} &= \arg\max_{\tilde{P}} (\ell(\theta^{(m-1)}) - D_{\mathrm{KL}}(\tilde{P} \,\|\, P_{\theta^{(m-1)}})) \\
&= \arg\min_{\tilde{P}} D_{\mathrm{KL}}(\tilde{P} \,\|\, P_{\theta^{(m-1)}}) \\
&= P_{\theta^{(m-1)}},
\end{aligned}$$

that is, $\tilde{P}^{(m)}$ has density $p(x \,|\, y, \theta^{(m-1)})$. Second, (2.7) can be rewritten using the Q-function:

$$\begin{aligned}
\theta^{(m)} &= \arg\max_{\theta \in \Omega} \ell(\theta) - D_{\mathrm{KL}}(\tilde{P}^{(m)} \,\|\, P_\theta) \\
&= \arg\max_{\theta \in \Omega} \log p(y \,|\, \theta) - D_{\mathrm{KL}}(\tilde{P}^{(m)} \,\|\, P_\theta) \\
&= \arg\max_{\theta \in \Omega} \log p(y \,|\, \theta) \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m-1)}) dx - D_{\mathrm{KL}}(\tilde{P}^{(m)} \,\|\, P_\theta) \\
&= \arg\max_{\theta \in \Omega} \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m-1)}) \log p(y \,|\, \theta) dx - D_{\mathrm{KL}}(\tilde{P}^{(m)} \,\|\, P_\theta) \\
&= \arg\max_{\theta \in \Omega} \int_{\mathcal{X}(y)} p(x \,|\, y, \theta^{(m-1)}) \log \frac{p(y \,|\, x) p(x \,|\, \theta)}{p(x \,|\, y, \theta)} dx \\
&\quad - D_{\mathrm{KL}}(\tilde{P}^{(m)} \,\|\, P_\theta)
\end{aligned}$$

(by Bayes' rule and the assumed Markov relationship)

$$= \arg\max_{\theta\in\Omega} \int_{\mathcal{X}(y)} p(x\,|\,y,\theta^{(m-1)}) \log \frac{p(x\,|\,\theta)}{p(x\,|\,y,\theta)} dx - D_{\mathrm{KL}}(\tilde{P}^{(m)} \,\|\, P_\theta)$$

(by removing the term that does not depend on θ)

$$= \arg\max_{\theta\in\Omega} \int_{\mathcal{X}(y)} p(x\,|\,y,\theta^{(m-1)}) \log \frac{p(x\,|\,\theta)}{p(x\,|\,y,\theta)} dx$$

$$- \int_{\mathcal{X}(y)} p(x\,|\,y,\theta^{(m-1)}) \log \frac{p(x\,|\,y,\theta^{(m-1)})}{p(x\,|\,y,\theta)} dx$$

$$= \arg\max_{\theta\in\Omega} \int_{\mathcal{X}(y)} p(x\,|\,y,\theta^{(m-1)}) \log p(x\,|\,\theta) dx$$

$$- \int_{\mathcal{X}(y)} p(x\,|\,y,\theta^{(m-1)}) \log p(x\,|\,y,\theta^{(m-1)}) dx$$

$$= \arg\max_{\theta\in\Omega} \int_{\mathcal{X}(y)} p(x\,|\,y,\theta^{(m-1)}) \log p(x\,|\,\theta) dx$$

(by removing the term that does not depend on θ)

$$= \arg\max_{\theta\in\Omega} E_{X|y,\theta^{(m-1)}}[\log p(X\,|\,\theta)]$$

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

which is just the standard M-step given in (1.4).

3

Learning Mixtures

This section details the use of EM for two popular problems of learning a mixture. First, we consider one of the simplest and nicest EM applications: learning the maximum likelihood mixture of a finite set of fixed models to explain some observed data y. Then in Section 3.2 we consider the harder problem of learning a GMM (also called *EM clustering*), where both the mixture weights of the Gaussians and the parameters for each component Gaussian must be learned. In Section 3.3 we illustrate using EM to learn a GMM when there are additional constraints on the GMM parameters. More examples of EM for mixture models can be found in McLachlan and Peel's book [37].

3.1 Learning an Optimal Mixture of Fixed Models

Consider the problem of learning an optimal convex combination of arbitrary models. Suppose you have n observations $y_1, y_2, \ldots, y_n$ and k Models that could have generated these observations $p_1, p_2, \ldots, p_k$. For example, p_1 could be a Gaussian with some fixed parameters $\mu = 3$, $\sigma^2 = 1$, and p_2 could be a Laplacian distribution with fixed parameter $\lambda = 1/5$, etc. This setup would apply to learning a GMM (treated later

in Section 3.2) if one fixed the component Gaussian models *a priori* such that you only needed to learn the relative weights.

Suppose further that you model the observed n samples as being drawn i.i.d. from a convex combination of the k models such that

$$p(Y_i = y_i) = \sum_{j=1}^{k} \theta_j p_j(y_i),$$

where $\sum_j \theta_j = 1$ and $\theta_j \in [0,1]$ for all j. Your goal is to learn the most likely combination θ of models to explain the observed data.

To use EM, let the hidden data $z = \begin{bmatrix} z_1 & z_2 & \dots & z_n \end{bmatrix}^{\mathrm{T}}$ denote which of the k models generated each corresponding observation, that is $z_i \in \{1,2,\dots,k\}$, $i = 1,\dots,n$. Then for any θ,

$$p(Y_i = y_i, Z_i = j \,|\, \theta) = \theta_j p_j(y_i), \tag{3.1}$$

and given an estimate $\theta^{(m)}$, it follows from (3.1) and Bayes' rule that

$$P(Z_i = j \,|\, Y_i = y_i, \theta^{(m)}) = \frac{p(Y_i = y_i, Z_i = j \,|\, \theta^{(m)})}{p(Y_i = y_i \,|\, \theta^{(m)})} = \frac{\theta_j^{(m)} p_j(y_i)}{\sum_{l=1}^{k} \theta_l^{(m)} p_l(y_i)}.$$

That is, if we know that the relative frequencies of the k models are $\theta^{(m)}$, then the probability that the ith observed sample y_i was generated by the jth model is proportional to both the probability $\theta_j^{(m)}$ of that model *a priori* and the likelihood $p_j(y_i)$ of the jth model producing the observed y_i.

Let Ω be the set of θ such that $\sum_j \theta_j = 1$ and $\theta_j \in [0,1]$, $j = 1,\dots,k$. Then the M-step is:

$$\begin{aligned}
\theta^{(m+1)} &= \arg\max_{\theta \in \Omega} E_{Z|y,\theta^{(m)}}[\log p(y, Z \,|\, \theta)] \\
&\quad \text{(by (1.5))} \\
&= \arg\max_{\theta \in \Omega} \sum_{i=1}^{n} E_{Z_i|y_i,\theta^{(m)}}[\log p(y_i, Z_i \,|\, \theta)] \\
&\quad \text{(by Proposition 1.1)} \\
&= \arg\max_{\theta \in \Omega} \sum_{i=1}^{n} E_{Z_i|y_i,\theta^{(m)}}[\log \theta_{Z_i} + \log p_{Z_i}(y_i)]
\end{aligned}$$

$$
\begin{aligned}
&= \arg\max_{\theta \in \Omega} \sum_{i=1}^{n} E_{Z_i|y_i,\theta^{(m)}}[\log \theta_{Z_i}] \\
&\quad \text{(by removing the terms that do not depend on } \theta) \\
&= \arg\max_{\theta \in \Omega} \sum_{i=1}^{n} \sum_{j=1}^{k} p(Z_i = j \,|\, y_i, \theta^{(m)}) \log \theta_j \\
&= \arg\max_{\theta \in \Omega} \sum_{j=1}^{k} \alpha_j^{(m)} \log \theta_j, \qquad (3.2)
\end{aligned}
$$

where in (3.2), we let

$$
\alpha_j^{(m)} = \sum_{i=1}^{n} p(Z_i = j \,|\, y_i, \theta^{(m)}) = \sum_{i=1}^{n} \frac{\theta_j^{(m)} p_j(y_i)}{\sum_{l=1}^{k} \theta_l^{(m)} p_l(y_i)}. \qquad (3.3)
$$

The constrained optimization in (3.2) can be solved in a straightforward manner with the method of Lagrange multipliers,[1] but a more elegant solution uses Gibbs' inequality,[2] which states that:

Gibbs' Inequality: Given two probability mass functions p and q for the same k events,

$$
\sum_{j=1}^{k} p_j \log q_j \leq \sum_{j=1}^{k} p_j \log p_j, \qquad (3.4)
$$

with equality if and only if $p_j = q_j$ for all j.

[1] Here is the solution to (3.2) using the method of Lagrange multipliers. Ignoring for the moment the constraint that $\theta_j \in [0,1]$, we use the method of Lagrange multipliers to enforce the constraint that $\sum_j \theta_j = 1$, and solve (3.2) analytically:

$$
0 = \frac{\partial}{\partial \theta_j} \left(\sum_{l=1}^{k} \alpha_l^{(m)} \log \theta_l - \lambda \left(\sum_{l=1}^{k} \theta_l - 1 \right) \right) = \frac{\alpha_j^{(m)}}{\theta_j} - \lambda,
$$

which leads to $\theta_j^\star = \alpha_j^{(m)}/\lambda$. By choosing the λ that satisfies the sum-to-one constraint, we have $\theta_j^\star = \alpha_j^{(m)} / \sum_{l=1}^{k} \alpha_l^{(m)}$. In this particular case, our gamble of ignoring the constraint $\theta_j \in [0,1]$ was okay since the solution happens to satisfy the constraint. Note that the problem in (3.2) is actually a convex optimization problem [5], and here we skip the details of verifying the optimality of the solution.

[2] Gibbs' inequality is also known as the log-sum divergence inequality, or by its equivalent result that relative entropy is always nonnegative.

To solve (3.2), let $p_j = \alpha_j^{(m)} / \sum_{l=1}^k \alpha_l^{(m)}$ and $q_j = \theta_j$. Then from Gibbs' inequality, the maximum of $\sum_j \alpha_j^{(m)} \log \theta_j$ occurs when $q_j = p_j$, that is, $\theta_j^\star = \alpha_j^{(m)} / \sum_{l=1}^k \alpha_l^{(m)}$.

To summarize, EM for learning an optimal mixture of fixed models reduces to iteratively solving for the k estimated weights:

$$\theta_j^{(m+1)} = \frac{\alpha_j^{(m)}}{\sum_{l=1}^k \alpha_l^{(m)}}, \quad j = 1, \ldots, k, \tag{3.5}$$

where $\alpha_j^{(m)}$ is given in (3.3). Here $\alpha_j^{(m)}$ is your best estimate at the mth iteration of the total relative probability of the jth model given your n observations. Then the updated estimate given in (3.5) normalizes the relative probability $\alpha_j^{(m)}$ to make it the absolute probability of the j^{th} model.

This is EM at its best: it provides a simple closed-form solution at each iteration. One is not always so lucky! But even in this case, one is not really so lucky: depending on your choice of fixed models $\{p_j\}_{j=1}^k$ and your random draw of data y, the likelihood surface may have multiple maxima, and EM may not converge to the global maximum.

3.2 Learning a GMM

In this section, we explain how to fit a GMM using EM. This is also called *EM clustering*. Figure 3.1 shows the probability density function of a one-dimensional GMM with three components. Fitting a GMM is a special case of the general problem of estimating a mixture of densities (for more on the general case, see [50]).

3.2.1 GMM Setup and Short Story

Suppose you are given n vectors $y_1, \ldots, y_n$ that you believe were generated i.i.d. by a mixture of k Gaussians,[3] and you want to find the means

[3] How did you know that your points came from a mixture of exactly k Gaussians? Sometimes one knows from side information. But if not, choosing the number of clusters k to assume is a difficult problem; see [56] for further discussion.

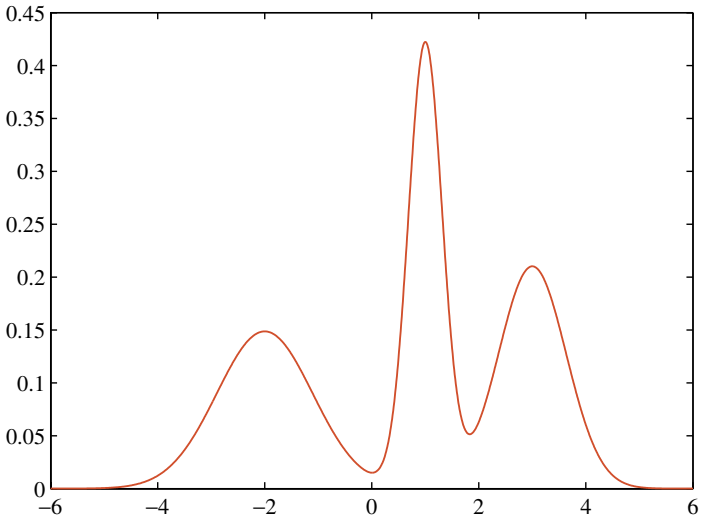

Fig. 3.1 Probability density of a one-dimensional GMM with three Gaussian components with means $\mu_1 = -2$, $\mu_2 = 1$, $\mu_3 = 3$, variances $\sigma_1^2 = 0.8$, $\sigma_2^2 = 0.1$, $\sigma_3^2 = 0.4$, and relative weights $w_1 = w_2 = w_3 = 1/3$.

and covariances of the k Gaussians, but you do not know which of the observed points came from which Gaussian. Your goal is to estimate the k means and k covariance matrices, and k weights that specify how likely each Gaussian is to be chosen; this entire set of parameters is θ. To find the maximum likelihood estimate of θ using EM, you define as the missing information z which of the k Gaussians each of the samples came from.

Spoiler alert! Before deriving the E-step and M-step, we summarize the outcome. The mth iteration of the E-step produces a guess of the $n \times k$ membership-weights $\{\gamma_{ij}^{(m)}\}$, where $\gamma_{ij}^{(m)}$ is the current guess of the probability that sample y_i came from the jth Gaussian, that is, $\gamma_{ij}^{(m)} = P(Z_i = j \,|\, y_i, \theta^{(m)})$. The M-step gives a closed-form solution for the new estimates of the mean and covariance for each Gaussian, and you complete your estimate of θ by setting the weight for the jth Gaussian to be proportional to the corresponding total membership-weight of the samples: $w_j \propto \sum_{i=1}^{n} \gamma_{ij}$.

3.2.2 Derivation of EM for the GMM

Given n i.i.d. samples $y_1, y_2, \ldots, y_n \in \mathbb{R}^d$ drawn from a GMM with k components, the goal is to estimate the parameter set $\theta = \{(w_j, \mu_j, \Sigma_j)\}_{j=1}^k$. For any y_i and any μ_j, Σ_j, denote the Gaussian

$$\phi(y_i \,|\, \mu_j, \Sigma_j) \triangleq \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(y - \mu_j)^T \Sigma_j^{-1}(y - \mu_j)\right).$$

The GMM has density

$$p(Y_i = y_i \,|\, \theta) = \sum_{j=1}^k w_j \phi(y_i \,|\, \mu_j, \Sigma_j),$$

where $w_j > 0$, $\sum_{j=1}^k w_j = 1$, and $\theta = \{(w_j, \mu_j, \Sigma_j)\}_{j=1}^k$.

Let $\gamma_{ij}^{(m)}$ be the estimate at the mth iteration of the probability that the ith sample was generated by the jth Gaussian component, that is,

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

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

Because the samples are i.i.d., we can apply Proposition 1.1,

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

where C is a constant that does not depend on θ and can thus be dropped without affecting the M-step. Then from Proposition 1.1,

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

which completes the E-step. For notational simplicity, denote the total membership weight of the jth Gaussian as

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

Then we can rewrite $Q(\theta\,|\,\theta^{(m)})$ as

$$\begin{aligned} Q(\theta\,|\,\theta^{(m)}) = &\sum_{j=1}^{k} n_j^{(m)}\left(\log w_j - \frac{1}{2}\log|\Sigma_j|\right) \\ &- \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{k}\gamma_{ij}^{(m)}(y_i-\mu_j)^T\Sigma_j^{-1}(y_i-\mu_j). \end{aligned} \tag{3.6}$$

The M-step is to solve

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & Q(\theta\,|\,\theta^{(m)}) \\ \text{subject to} \quad & \sum_{j=1}^{k} w_j = 1,\ w_j > 0, \quad j=1,\ldots,k, \\ & \Sigma_j \succ 0, \quad j=1,\ldots,k, \end{aligned} \tag{3.7}$$

where $\Sigma_j \succ 0$ means that Σ_j is positive definite.

From (3.6), one sees that we can independently maximize the Q-function with respect to the weights, and this requires maximizing the term $\sum_j n_j^{(m)}\log w_j$. This is the same problem we faced in Section 3.1, given in (3.2), and the solution is the same:

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

The optimal μ_j and Σ_j can be found by setting the corresponding derivatives to zero.[4] To solve for the means, we let

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

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 covariance matrices, we let[5]

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

and thus

$$\Sigma_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{ij}^{(m)} (y_i - \mu_j^{(m+1)})(y_i - \mu_j^{(m+1)})^T,$$

for $j = 1, \ldots, k$.

We summarize the whole procedure in Table 3.1.

3.2.3 Initialization

It is not uncommon to initialize EM clustering by randomly choosing k of the n samples and making these the first estimates of the cluster means, setting the first estimate of the covariances to be identity matrices, and the first guess at the weights $w_1 = \cdots = w_k = 1/k$.

[4] Note that the problem in (3.7) is a convex optimization problem, so setting the derivatives to zero corresponds to the optimal means and covariances. For details on how to analyze convexity and optimality conditions, see for example [5].

[5] See [47] for matrix derivatives.

Table 3.1. EM algorithm for estimating GMM parameters.

1. **Initialization:** Choose initial estimates $w_j^{(0)}$, $\mu_j^{(0)}$, $\Sigma_j^{(0)}$, $j = 1, \ldots, k$, and compute the initial log-likelihood
$$\ell^{(0)} = \frac{1}{n} \sum_{i=1}^{n} \log \Big(\sum_{j=1}^{k} w_j^{(0)} \phi(y_i \,|\, \mu_j^{(0)}, \Sigma_j^{(0)}) \Big).$$
2. **E-step:** For $j = 1, \ldots, k$, compute
$$\gamma_{ij}^{(m)} = \frac{w_j^{(m)} \phi(y_i \,|\, \mu_j^{(m)}, \Sigma_j^{(m)})}{\sum_{l=1}^{k} w_l^{(m)} \phi(y_i \,|\, \mu_l^{(m)}, \Sigma_l^{(m)})}, \quad i = 1, \ldots, n,$$
and
$$n_j^{(m)} = \sum_{i=1}^{n} \gamma_{ij}^{(m)}.$$
3. **M-step:** For $j = 1, \ldots, k$, compute the new estimates
$$w_j^{(m+1)} = \frac{n_j^{(m)}}{n},$$
$$\mu_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{ij}^{(m)} y_i,$$
$$\Sigma_j^{(m+1)} = \frac{1}{n_j^{(m)}} \sum_{i=1}^{n} \gamma_{ij}^{(m)} (y_i - \mu_j^{(m+1)})(y_i - \mu_j^{(m+1)})^{\mathrm{T}},$$
4. **Convergence check:** Compute the new log-likelihood
$$\ell^{(m+1)} = \frac{1}{n} \sum_{i=1}^{n} \log \Big(\sum_{j=1}^{k} w_j^{(m+1)} \phi(y_i \,|\, \mu_j^{(m+1)}, \Sigma_j^{(m+1)}) \Big).$$
Return to Step 2 if $|\ell^{(m+1)} - \ell^{(m)}| > \delta$ for a preset threshold δ; otherwise end the algorithm.

Common wisdom is that initializing by first doing a cheaper clustering will generally produce more desirable results. In particular, the k-means algorithm (see Section 1.2) is often used to find a good initialization for EM clustering. A common approach is to use the k-means clustering to provide a first estimate of $\gamma_{ij}^{(0)} = \hat{P}(Z_i = j \,|\, Y_i = y_i)$, where for the ith sample, $\gamma_{ij}^{(0)} = 1$ for only the one Gaussian that k-means assigns sample y_i to and $\gamma_{ij} = 0$ for all the other components for that y_i; then start EM at the M-step based on this $\gamma_{ij}^{(0)}$. In the example presented in Section 3.2.4, however, we simply initialize EM by using the cluster means from k-means as the estimated EM means and setting the covariance estimates to be identity matrices and the weights $w_1 = \cdots = w_k = 1/k$; then start EM at the E-step.

3.2.4 An Example of GMM Fitting

Consider a two-component GMM in $\mathbb{R}^2$ with the following parameters
$$\mu_1 = \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \quad \mu_2 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix},$$
and relative weights $w_1 = 0.6$ and $w_2 = 0.4$. Its density is shown in Figure 3.2, which also shows 1000 samples randomly drawn from this

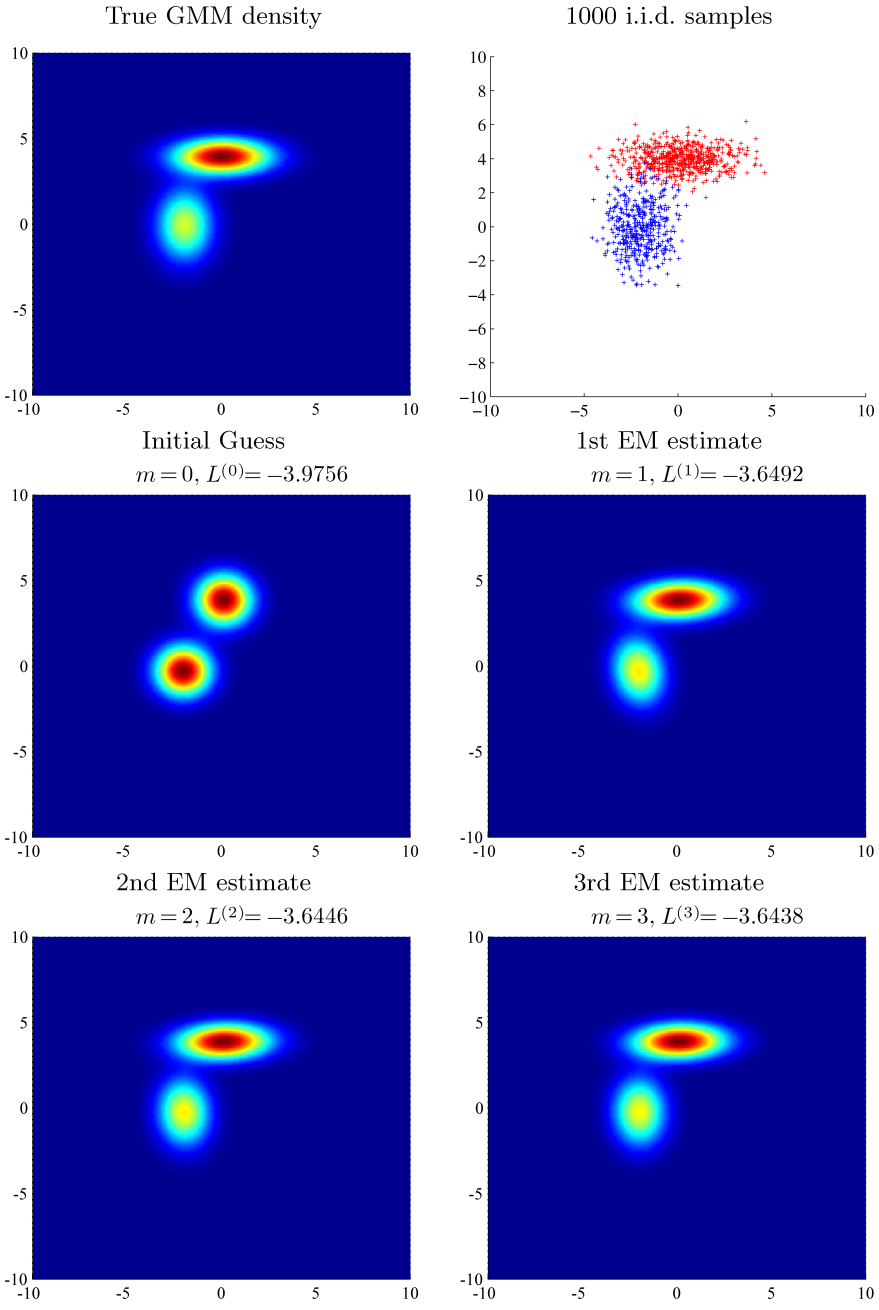

Fig. 3.2 GMM fitting example.

distribution; samples from the first and second components are marked red and blue, respectively.

We ran the k-means algorithm on the 1000 samples and used the centroids of the two k-means 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 $\Sigma_1^{(0)} = \Sigma_2^{(0)} = I_2$, where I_2 denotes the 2×2 identity matrix. The density corresponding to these initial estimates is shown in Figure 3.2. We set $\delta = 10^{-3}$, and in this example, the EM algorithm only needs three iterations to converge. Figure 3.2 shows the estimated density at each iteration. The final estimates are

$$w_1^{(3)} = 0.5966, \quad \mu_1^{(3)} = \begin{bmatrix} 0.0806 \\ 3.9445 \end{bmatrix}, \qquad \Sigma_1^{(3)} = \begin{bmatrix} 2.7452 & 0.0568 \\ 0.0568 & 0.4821 \end{bmatrix},$$
$$w_2^{(3)} = 0.4034, \quad \mu_2^{(3)} = \begin{bmatrix} -2.0181 \\ -0.1740 \end{bmatrix}, \quad \Sigma_2^{(3)} = \begin{bmatrix} 0.8750 & -0.0153 \\ -0.0153 & 1.7935 \end{bmatrix}.$$

3.2.5 Singularity Problem in Using EM for GMM Fitting

The EM algorithm does well in the previous example, but sometimes it fails by approaching singularities of the log-likelihood function, especially when the number of observations n is not large relative to the number of Gaussian components k. This is an inherent problem with applying maximum likelihood estimation to GMMs due to the fact that the log-likelihood function $\ell(\theta)$ is not bounded above, as we illustrate in the following example. First, let $\mu_1 = y_1$, $\Sigma_1 = \sigma_1^2 I_d$ and $0 < w_1 < 1$. Then the log-likelihood is

$$\begin{aligned} \ell(\theta) &= \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} w_j \phi(y_i \,|\, \mu_j, \Sigma_j) \right) \\ &= \log \left(\sum_{j=1}^{k} w_j \phi(y_1 \,|\, \mu_j, \Sigma_j) \right) + \sum_{i=2}^{n} \log \left(\sum_{j=1}^{k} w_j \phi(y_i \,|\, \mu_j, \Sigma_j) \right) \end{aligned}$$

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

So far, everything seems fine. But if we let $\sigma_1^2 \to 0$ and keep everything else fixed, then the above lower bound of $\ell(\theta)$ diverges to infinity, and thus $\ell(\theta) \to \infty$. So for a GMM, maximizing the likelihood is actually an ill-posed problem.

This problem most often arises in practice with EM when the number of components k is too large compared to the number of observations n, but it can also occur if one sample is relatively far from the bulk of the other samples. In both cases, a single Gaussian model becomes predominantly associated with one observation, and as the iterations progress, that Gaussian model shrinks its variance around that one single observation. More generally, this problem can arise if the samples predominantly assigned to a component Gaussian do not span the space, so that the estimated covariance of that Gaussian is not of full rank.

In order to avoid such singularities when applying the EM algorithm, one can resort to ad hoc techniques such as re-initializing the algorithm after detecting that one component is "collapsing" onto a data sample; one can also adopt the Bayesian approach (discussed in Section 1.3, and illustrated in the next subsection) as a more principled way to deal with this problem.

3.3 Estimating a Constrained GMM

In practice, one may wish to constrain the parameters of the GMM, either to incorporte prior information about what is being modeled, or to regularize the GMM to avoid the degenerate solutions as discussed in Section 3.2.5. In this subsection, we illustrate learning a restricted

GMM with a recent example by Chen and Krumm [10], where the set of restrictions was designed to model the GPS traces of cars driving along city streets. In addition to hard constraints, this example also uses the MAP variant of EM (see Section 1.3) to incorporate other prior information about the parameters through a prior distribution, and to ensure robust estimation.

The goal is to model the probability density of cars within the road based on observed GPS signals recorded in the cars. Then the probability model can be used to analyze multi-lane structure, especially at intersections, for automated mapping algorithms. For a given perpendicular cross-section of the road, the observed GPS traces were modeled as being generated i.i.d. from a one-dimensional GMM, where k, the number of Gaussian components, corresponds to the number of lanes, the weights $w_1, \ldots, w_k$ correspond to the relative traffic volume in each lane, and the Gaussian means $\mu_1, \ldots, \mu_k$ are used to model the center of each lane.

Restrictions to the GMM were added based on prior knowledge of roads. First, the widths of the lanes were expected to be approximately the same. This observation can be translated into the constraint that μ_j's are equally spaced, that is,

$$\mu_j = \mu + (j-1)\Delta\mu, \quad j = 1, \ldots, k, \tag{3.8}$$

where $\Delta\mu$ is the change between two adjacent μ_j's, and μ is the mean of either the leftmost or rightmost component along the sampling line, depending on the sign of $\Delta\mu$. Second, assume that the causes of the spread of the GPS traces are approximately the same for all the lanes, such that all the Gaussian components are restricted to have the same variance:

$$\sigma_j^2 = \sigma^2, \quad j = 1, \ldots, k. \tag{3.9}$$

In fact, forcing all the variances to be the same is a common restriction in learning a GMM even when there is no application-specific reason to assume it is true, but if the dimensionality of the observed data y_i is high compared to the number of samples n, restricting the Gaussian components to have the same covariance reduces the number of free variables to estimate, and this can reduce training time and lower

estimation error by reducing the estimation variance (see Chapter 2 of Hastie et al. [21] for a good discussion of estimation variance and bias). Combining (3.8) and (3.9) produces the following density for this one-dimensional restricted GMM:

$$p(y_i) = \sum_{j=1}^{k} w_j \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mu - (j-1)\Delta\mu)^2}{2\sigma^2}\right).$$

For robust estimation and to incorporate additional prior information, we use the MAP variant of the EM algorithm (see Section 1.3). For the shared variance σ^2, we use an inverse gamma prior:

$$\sigma^2 \sim \text{Inv-Gamma}\left(\frac{\nu}{2}, \frac{\varsigma^2}{2}\right),$$

which is a mathematically convenient choice because it serves as a conjugate prior of the variance of a Gaussian distribution. For the weights $w_1, \ldots, w_k$ and for the parameter μ, we use uniform priors.[6] However, for $\Delta\mu$ we use a normal prior conditioned on σ^2:

$$\Delta\mu \,|\, \sigma^2 \sim \mathcal{N}\left(\eta, \frac{\sigma^2}{\kappa}\right).$$

Then the prior distribution of the parameters has density:

$$p(\theta) \propto \left(\sigma^2\right)^{-\frac{\nu+3}{2}} \exp\left(-\frac{\varsigma^2 + \kappa(\Delta\mu - \eta)^2}{2\sigma^2}\right). \tag{3.10}$$

Now we show how to derive the MAP EM steps for learning the set of parameters $\theta = (w_1, \ldots, w_k, \mu, \Delta\mu, \sigma^2)$ for this restricted GMM. For the E-step, let

$$\gamma_{ij}^{(m)} = \frac{w_j^{(m)} \phi(x_i \,|\, \mu_j^{(m)}, \sigma^{(m)})}{\sum_{l=1}^{k} w_l^{(m)} \phi(x_i \,|\, \mu_l^{(m)}, \sigma^{(m)})},$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, k$, where

$$\mu_j^{(m)} = \mu^{(m)} + (j-1)\Delta\mu^{(m)}, \quad j = 1, \ldots, k.$$

[6] The uniform prior on μ is an improper prior since $\mu \in \mathbb{R}$.

Let

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

for $j = 1, \ldots, k$. The Q-function is

$$\begin{aligned} Q(\theta \,|\, \theta^{(m)}) &= \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij}^{(m)} \log(w_j \phi(x_i \,|\, \mu + (j-1)\Delta\mu, \sigma)) \\ &= \sum_{j=1}^{k} n_j^{(m)} \log w_j - \frac{n}{2} \log 2\pi - \frac{n}{2} \log \sigma^2 \\ &\quad - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij}^{(m)} (x_i - \mu - (j-1)\Delta\mu)^2. \end{aligned}$$

With (3.10) and the above Q-function, the MAP EM M-step is:

$$\begin{aligned} &\theta^{(m+1)} \\ &\quad = \arg\max_{\theta} (Q(\theta \,|\, \theta^{(m)}) + \log p(\theta)), \\ &\quad = \arg\max_{\theta} \left(\sum_{j=1}^{k} n_j^{(m)} \log w_j - \frac{n + \nu + 3}{2} \log \sigma^2 - \frac{\varsigma^2 + \kappa(\Delta\mu - \eta)^2}{2\sigma^2} \right. \\ &\qquad \left. - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij}^{(m)} (x_i - \mu - (j-1)\Delta\mu)^2 + C \right), \end{aligned}$$

where C is a constant that does not depend on θ. The weights that solve the above M-step are the same as the standard GMM:

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

To solve the M-step for μ and $\Delta\mu$, we let

$$\frac{\partial}{\partial \mu} (Q(\theta \,|\, \theta^{(m)}) + \log p(\theta)) = 0, \tag{3.11}$$

and

$$\frac{\partial}{\partial \Delta\mu} (Q(\theta \,|\, \theta^{(m)}) + \log p(\theta)) = 0. \tag{3.12}$$

Combining (3.11) and (3.12) produces the linear system of equations

$$A \begin{bmatrix} \mu \\ \Delta\mu \end{bmatrix} = b, \tag{3.13}$$

for the matrix $A = \begin{bmatrix} a_{ij} \end{bmatrix}_{2\times 2}$ with

$$\begin{aligned} a_{11} &= 1, \\ a_{12} = a_{21} &= \sum_{j=1}^{k-1} w_{j+1}^{(m+1)} j, \\ a_{22} &= \sum_{j=1}^{k-1} w_{j+1}^{(m+1)} j^2 + \frac{\kappa}{n}, \end{aligned}$$

and $b = \begin{bmatrix} b_1 & b_2 \end{bmatrix}^{\mathrm{T}}$ with

$$\begin{aligned} b_1 &= \frac{1}{n} \sum_{i=1}^{n} x_i, \\ b_2 &= \frac{\kappa\eta}{n} + \frac{1}{n} \sum_{i=1}^{n} \sum_{j=2}^{k} \gamma_{ij}^{(m)} (j-1) x_i. \end{aligned}$$

To confirm that (3.13) has a unique solution, consider

$$\begin{aligned} \left(\sum_{j=1}^{k-1} w_{j+1}^{(m+1)} j \right)^2 &= \left(\sum_{j=1}^{k-1} \sqrt{w_{j+1}^{(m+1)}} \sqrt{w_{j+1}^{(m+1)} j^2} \right)^2 \\ &\leq \left(\sum_{j=1}^{k-1} w_{j+1}^{(m+1)} \right) \left(\sum_{j=1}^{k-1} w_{j+1}^{(m+1)} j^2 \right) \\ &\quad \text{(follows from the Cauchy–Schwarz inequality)} \\ &\leq \sum_{j=1}^{k-1} w_{j+1}^{(m+1)} j^2 \\ &\quad \text{(follows from } 0 \leq \sum_{j=1}^{k-1} w_{j+1}^{(m+1)} \leq 1) \end{aligned}$$

$$< \sum_{j=1}^{k-1} w_{j+1}^{(m+1)} j^2 + \frac{\kappa}{n},$$

(follows from $\kappa > 0$).

Hence

$$\det A = a_{11}a_{22} - a_{12}a_{21} = \sum_{j=1}^{k-1} w_{j+1}^{(m+1)} j^2 + \frac{\kappa}{n} - \left(\sum_{j=1}^{k-1} w_{j+1}^{(m+1)} j \right)^2 > 0,$$

which confirms that (3.13) has a unique solution. Thus the new estimates are:

$$\mu^{(m+1)} = \frac{a_{22}b_1 - a_{12}b_2}{\det A},$$

and

$$\Delta\mu^{(m+1)} = \frac{a_{11}b_2 - a_{21}b_1}{\det A}.$$

Lastly, to solve for σ^2, we let

$$\frac{\partial}{\partial \sigma^2}(Q(\theta \,|\, \theta^{(m)}) + \log p(\theta)) = 0,$$

which yields

$$\sigma^{(m+1)} = \sqrt{\frac{\varsigma^2 + \kappa(\Delta\mu^{(m+1)} - \eta)^2 + \sum_{i=1}^{n}\sum_{j=1}^{k} \gamma_{ij}^{(m)} (x_i - \mu_j^{(m+1)})^2}{n + \nu + 3}}.$$

To illustrate the difference between the standard GMM and the restricted GMM, consider the following example. A standard GMM and the above restricted GMM were fit to 137 GPS traces with $k = 2$ Gaussian components. The standard GMM estimated relative weights of $\hat{w}_1 = 0.7$, $\hat{w}_2 = 0.3$, traffic lanes centered at $\hat{\mu}_1 = 4.7$ and $\hat{\mu}_2 = 8.2$, and variances within each lane of $\hat{\sigma}_1^2 = 4.5$, and $\hat{\sigma}_2^2 = 0.6$. The restricted GMM estimated more balanced relative weights of $\hat{w}_1 = 0.4$, $\hat{w}_2 = 0.6$, greater separation between the lane centers with $\hat{\mu}_1 = 3.5$, $\hat{\mu}_2 = 7.5$, and (by constraint) the same in-lane variance $\hat{\sigma}_1^2 = \hat{\sigma}_2^2 = 2.1$. Because of the restrictions it was faster to train the restricted GMM: its EM algorithm converged after 17 iterations, while the EM algorithm for the standard GMM parameter estimation required 136 iterations to reach the same stopping condition.

4

More EM Examples

We derive three more examples of EM. First, we learn a standard HMM. Second, we consider an example problem from the general class of signal superposition problems. Third, we show how to estimate the parameter for a compound Dirichlet distribution.

4.1 Learning a Hidden Markov Model

An HMM is used to model random sequences. Baum et al. and Welch derived the EM algorithm for this problem in the 1960s [4, 61] before EM was formalized as a general technique by Dempster et al. [11], and thus EM applied to learning an HMM is also called the Baum–Welch algorithm.

4.1.1 HMM Setup

Suppose you observe one sequence y of length T that is a realization of a random sequence Y such that

$$y = \begin{bmatrix} y_1 & y_2 & \dots & y_t & \dots & y_T \end{bmatrix}. \tag{4.1}$$

For example, y_t might be the cell strength of your cell phone at the beginning of the tth hour of the day, or y_t might be the tth word in a movie, or y_t could be the tth base (`A`, `T`, `G`, `C`) in a DNA sequence.

Note we have been using i or j to index our samples in the other EM examples, but in (4.1) we change the index to t to emphasize that in this case the situation is a little different. We usually have n i.i.d. samples, and for an HMM one could have n observed sequences that are independent realizations of a random sequence. However, you only need *one* observed sequence to learn an HMM (if you have multiple independent sequences, use Proposition 1.1 to sum the Q-function derived below). For simplicity, we assume only one observed sequence. We notate its components with the index t as in (4.1).

The HMM assumes there exists some other corresponding hidden sequence called the state sequence z:

$$z = \begin{bmatrix} z_1 & z_2 & \dots & z_t & \dots & z_T \end{bmatrix},$$

and that given the sequence z, the elements of Y are conditionally independent. For the cell phone strength example, the hidden state might be the distance to the nearest cell tower. For the word example, the z_t might specify which actor in the movie spoke the utterance y_t. For phoneme recognition, which is usually the first step of speech recognition, it is common to process the original acoustic time signal into a time-indexed sequence of MFCC (Mel-frequency cepstral coefficients) feature vectors $Y_t \in \mathbb{R}^d$, and then model the sequence of MFCC feature vectors for each phoneme as a realization of an HMM, where the hidden states are the more detailed sub-phone units. For simplicity of the derivations, we restrict ourselves to the case where the hidden states can take on one of G fixed values such that $Z_t \in \{1, 2, \dots, G\}$, though this is not a restriction of HMMs.

An HMM makes two assumptions. First, that 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} (the Markov property):

$$p(Z_t = g \,|\, z_{t-1}, z_{t-2}, \dots, z_1) = p(Z_t = g \,|\, z_{t-1}). \tag{4.2}$$

Second, the observation Y_t at time t does not depend on other observations nor other states given the hidden state z_t at time t:

$$p(Y_t = y_t \,|\, z, y_1, y_2, \ldots, y_{t-1}, y_{t+1}, \ldots, y_T) = p(y_t \,|\, z_t). \tag{4.3}$$

The HMM has the following parameters:

(1) An initial probability distribution over the G possible hidden states[1]: $\pi = \begin{bmatrix} \pi_1 & \ldots & \pi_G \end{bmatrix}^{\mathrm{T}}$, where $\pi_g = p(Z_1 = g)$.
(2) A hidden-state transition probability[2] matrix $\mathbf{P} \in \mathbb{R}^{G \times G}$ that specifies the probability of transitioning from state g to state h: $\mathbf{P}_{g,h} = p(Z_t = h \,|\, Z_{t-1} = g)$.
(3) The probability distribution of observations $Y \in \mathbb{R}^d$ given hidden state g; we parameterize this with parameter set b_g such that $p(Y_t = y \,|\, Z_t = g) = p(y \,|\, b_g)$. For example, in modeling a DNA sequence, the parameter b_g is the pmf that specifies the probabilities of `A`, `T`, `G` and `C` being observed if the hidden state is $Z_t = g$. In modeling speech sequences, it is common to assume that given a particular hidden state, an observed MFCC feature vector $Y_t \in \mathbb{R}^d$ is drawn from a GMM whose parameters depend on the hidden state. In this case the parameter set b_g for the gth hidden state includes all the parameters for the corresponding GMM, so $b_g = \{(w_{gj}, \mu_{gj}, \Sigma_{gj})\}_{j=1}^{k_g}$, where k_g is the number of Gaussian components in the GMM corresponding to the gth hidden state.

Thus for an HMM the complete set of parameters to estimate is $\theta = \{\pi, \mathbf{P}, b\}$, where $b = \{b_g\}_{g=1}^{G}$. Next, we describe EM for the HMM; for more introductory material about HMMs, see [49].

4.1.2 Estimating the Transition Probability Matrix P

The M-steps for π, $\mathbf{P}$ and b are each independent of the other parameters to be estimated, though each M-step depends on the

[1] HMMs with continuous state spaces analogously have a corresponding initial probability density π.
[2] We assume a time-homogenous HMM such that the transition probability matrix does not depend on t. HMMs with continuous state spaces analogously have a conditional transition density $\mathbf{P}$.

complete set of last-iteration parameter estimates $\theta^{(m)}$. We will show that the M-step for $\mathbf{P}$ has a closed-form solution, though it will take a few steps:

$$
\begin{aligned}
\mathbf{P}^{(m+1)} &= \arg\max_{\mathbf{P}} E_{Z|y,\theta^{(m)}}[\log p(y, Z \,|\, \theta)] \\
&= \arg\max_{\mathbf{P}} \sum_z p(z \,|\, y, \theta^{(m)}) \log p(y, z \,|\, \pi, \mathbf{P}, b) \\
&= \arg\max_{\mathbf{P}} \sum_z p(z \,|\, y, \theta^{(m)}) \log (p(y \,|\, z, b) p(z \,|\, \pi, \mathbf{P})) \\
&= \arg\max_{\mathbf{P}} \sum_z p(z \,|\, y, \theta^{(m)}) \log p(z \,|\, \pi, \mathbf{P}) \\
&\quad \text{(drop } p(y \,|\, z, b) \text{ since it does not depend on } \mathbf{P}) \\
&= \arg\max_{\mathbf{P}} \sum_z p(z \,|\, y, \theta^{(m)}) \log \left(\left(\prod_{t=2}^{T} p(z_t \,|\, z_{t-1}, \mathbf{P}) \right) p(z_1 \,|\, \pi) \right) \\
&\quad \text{(apply the chain rule and the Markov property)} \\
&= \arg\max_{\mathbf{P}} \sum_z p(z \,|\, y, \theta^{(m)}) \log \left(\prod_{t=2}^{T} p(z_t \,|\, z_{t-1}, \mathbf{P}) \right) \\
&\quad \text{(drop } p(z_1 \,|\, \pi) \text{ since it does not depend on } \mathbf{P}) \\
&= \arg\max_{\mathbf{P}} \sum_z p(z \,|\, y, \theta^{(m)}) \log \left(\prod_{t=2}^{T} \mathbf{P}_{z_{t-1}, z_t} \right) \\
&= \arg\max_{\mathbf{P}} \sum_z p(z \,|\, y, \theta^{(m)}) \log \left(\prod_{g=1}^{G} \prod_{h=1}^{G} \mathbf{P}_{g,h}^{\zeta_{gh}(z)} \right),
\end{aligned}
$$

where $\zeta_{gh}(z)$ is the number of transitions from state g to state h in z. With the notation $\zeta_{gh}(z)$, the above equation can be continued as follows,

$$
\begin{aligned}
\mathbf{P}^{(m+1)} &= \arg\max_{\mathbf{P}} \sum_z \left(p(z \,|\, y, \theta^{(m)}) \sum_{g=1}^{G} \sum_{h=1}^{G} \zeta_{gh}(z) \log \mathbf{P}_{g,h} \right) \\
&= \arg\max_{\mathbf{P}} \sum_{g=1}^{G} \sum_{h=1}^{G} \left(\sum_z p(z \,|\, y, \theta^{(m)}) \zeta_{gh}(z) \right) \log \mathbf{P}_{g,h} \,.
\end{aligned}
$$

Given that $\mathbf{P}$ must be a right stochastic matrix, that is, for $g = 1, \ldots, G$, $\sum_{h=1}^{G} \mathbf{P}_{g,h} = 1$ and $\mathbf{P}_{g,h} \geq 0$, $h = 1, \ldots, G$, we can solve the last line of

the above equation for each row of $\mathbf{P}$ independently:

$$\mathbf{P}_{g,\cdot}^{(m+1)} = \arg\max_{\mathbf{P}_{g,\cdot}} \sum_{h=1}^{G} \left(\sum_{z} p(z \,|\, y, \theta^{(m)}) \zeta_{gh}(z) \right) \log \mathbf{P}_{g,h}, \tag{4.4}$$

where $\mathbf{P}_{g,\cdot}$ denotes the gth row of $\mathbf{P}$. To solve (4.4), apply Gibbs' inequality as given in (3.4) with

$$\begin{aligned} q_h &= \mathbf{P}_{g,h}, \\ p_h &= \frac{\sum_z p(z \,|\, y, \theta^{(m)}) \zeta_{gh}(z)}{\sum_{l=1}^{G} \sum_z p(z \,|\, y, \theta^{(m)}) \zeta_{gl}(z)}, \end{aligned}$$

and conclude that the maximum of (4.4) must occur when Gibbs' inequality holds with equality, that is, when $q_h = p_h$ for all h, and thus

$$\mathbf{P}_{g,h}^{(m+1)} = \frac{\sum_z p(z \,|\, y, \theta^{(m)}) \zeta_{gh}(z)}{\sum_{l=1}^{G} \sum_z p(z \,|\, y, \theta^{(m)}) \zeta_{gl}(z)}. \tag{4.5}$$

At first glance, (4.5) looks like an awful state of affairs — it requires iterating through all possible G^T sequences of z! Luckily, there is another way to compute (4.5). Let $1_{\{A=a\}}$ be a random indicator that is 1 if the random variable $A = a$ and 0 otherwise; $1_{\{A=a\}}$ is a Bernoulli random variable, and thus its expectation is $E_A[1_{\{A=a\}}] = p(A = a)$. To simplify (4.5), rewrite its numerator as an expectation:

$$\begin{aligned} \sum_z p(z \,|\, y, \theta^{(m)}) \zeta_{gh}(z) &= E_{Z|y,\theta^{(m)}}[\zeta_{gh}(Z)] \\ &= E_{Z|y,\theta^{(m)}} \left[\sum_{t=2}^{T} 1_{\{Z_{t-1}=g, Z_t=h\}} \right] \\ &= \sum_{t=2}^{T} E_{Z|y,\theta^{(m)}}[1_{\{Z_{t-1}=g, Z_t=h\}}] \\ &= \sum_{t=2}^{T} p(Z_{t-1} = g, Z_t = h \,|\, y, \theta^{(m)}). \end{aligned} \tag{4.6}$$

Voilà! We have converted the numerator of (4.5) into merely a sum over the sequence! But alas, we still have to compute $p(Z_{t-1} = g, Z_t = h \,|\, y, \theta^{(m)})$. This will not hurt too much — just a little recursion. Using

Bayes' rule:

$$\begin{aligned} & p(Z_{t-1}=g, Z_t=h \,|\, y, \theta^{(m)}) \\ &= \frac{p(Z_{t-1}=g, Z_t=h, y \,|\, \theta^{(m)})}{p(y \,|\, \theta^{(m)}} \\ &= \frac{p(Z_{t-1}=g, Z_t=h, y \,|\, \theta^{(m)})}{\sum_{i=1}^{G}\sum_{j=1}^{G} p(Z_{t-1}=i, Z_t=j, y \,|\, \theta^{(m)})}, \end{aligned}$$

so really one only needs to be able to compute:

$$p(Z_{t-1}=g, Z_t=h, Y=\begin{bmatrix} y_1 & \dots & y_{t-1} & y_t & \dots & y_T \end{bmatrix} \,|\, \theta^{(m)}),$$

which, by the chain rule, can be expanded into the product of the following four terms:

(1) $p(Z_{t-1}=g, y_1, \dots, y_{t-1} \,|\, \theta^{(m)})$,
(2) $p(Z_t=h \,|\, Z_{t-1}=g, y_1, \dots, y_{t-1}, \theta^{(m)})$,
(3) $p(y_{t+1}, \dots, y_T \,|\, Z_t=h, Z_{t-1}=g, y_1, \dots, y_{t-1}, \theta^{(m)})$,
(4) $p(y_t \,|\, Z_t=h, Z_{t-1}=g, y_1, \dots, y_{t-1}, y_{t+1}, y_T, \theta^{(m)})$.

Using the Markov property and conditional independence property of the HMM model, the above four terms can be simplified respectively into the following terms:

(1) $p(Z_{t-1}=g, y_1, \dots, y_{t-1} \,|\, \theta^{(m)})$, called the *forward probability* and denoted by $\alpha_{t-1}^{(m)}(g)$ — we will explain how to compute it below;
(2) $p(Z_t=h \,|\, Z_{t-1}=g, \theta^{(m)})$, which is $\mathbf{P}_{g,h}^{(m)}$;
(3) $p(y_{t+1}, \dots, y_T \,|\, Z_t=h, \theta^{(m)})$, called the *backward probability* and denoted by $\beta_t^{(m)}(h)$ — we will explain how to compute it below;
(4) $p(y_t \,|\, Z_t=h, \theta^{(m)}) = p(y_t \,|\, b_h^{(m)})$, which we assume is computable.

The forward probability $\alpha_{t-1}^{(m)}(g)$ can be computed recursively:

$$\alpha_{t-1}^{(m)}(g) = p(y_{t-1} \,|\, b_g^{(m)}) \left(\sum_{l=1}^{G} \alpha_{t-2}^{(m)}(l) \mathbf{P}_{l,g}^{(m)} \right), \tag{4.7}$$

where

$$\alpha_1^{(m)}(l) = \pi_l^{(m)} p(y_1 \,|\, b_l^{(m)}), \quad l = 1, \ldots, G.$$

The backward probability $\beta_t^{(m)}(h)$ can also be computed recursively:

$$\beta_t^{(m)}(h) = \sum_{l=1}^{G} \beta_{t+1}^{(m)}(l) \mathbf{P}_{h,l}^{(m)} p(y_{t+1} \,|\, b_l^{(m)}), \tag{4.8}$$

where

$$\beta_T^{(m)}(l) = 1, \quad l = 1, \ldots, G.$$

The recursive computation of the α and β terms is referred to as the *forward–backward algorithm.*

To summarize this subsection: (i) we simplified the M-step for $\mathbf{P}$ to get $\mathbf{P}_{g,h}^{(m+1)}$ as given by (4.5); (ii) we showed that each sum over z in (4.5) could be expressed as a sum over t as given by (4.6); (iii) each term in the sum of (4.6) can be expressed as the product of the four terms listed above; (iv) two of those four terms are straightforward to compute from $\theta^{(m)}$, and the remaining two terms can be computed recursively as given by (4.7) and (4.8).

4.1.3 Estimating the Initial Distribution $\boldsymbol{\pi}$

The M-step for π is:

$$\begin{aligned}
\pi^{(m+1)} &= \arg\max_{\pi} E_{Z|y,\theta^{(m)}}[\log p(y, Z \,|\, \theta)] \\
&= \arg\max_{\pi} \sum_{z} p(z \,|\, y, \theta^{(m)}) \log p(y, z \,|\, \pi, \mathbf{P}, b) \\
&= \arg\max_{\pi} \sum_{z} p(z \,|\, y, \theta^{(m)}) \log(p(y \,|\, z, b) p(z \,|\, \pi, \mathbf{P})) \\
&= \arg\max_{\pi} \sum_{z} p(z \,|\, y, \theta^{(m)}) \log p(z \,|\, \pi, \mathbf{P}) \\
&\qquad \text{(drop } p(y \,|\, z, b) \text{ since it does not depend on } \pi) \\
&= \arg\max_{\pi} \sum_{z} p(z \,|\, y, \theta^{(m)}) \log \left(\pi_{z_1} \prod_{t=2}^{T} \mathbf{P}_{t-1,t} \right)
\end{aligned}$$

$$= \arg\max_{\pi} \sum_{z} p(z \,|\, y, \theta^{(m)}) \log \pi_{z_1}$$

$$= \arg\max_{\pi} \sum_{g=1}^{G} p(Z_1 = g \,|\, y, \theta^{(m)}) \log \pi_g. \tag{4.9}$$

Recall that $\sum_{g=1}^{G} \pi_g = 1$, so to solve (4.9), we can apply Gibbs' inequality as given in (3.4) with

$$q_g = \pi_g,$$
$$p_g = p(Z_1 = g \,|\, y, \theta^{(m)}),$$

and conclude that the maximum of (4.9) must occur when Gibbs' inequality holds with equality, that is, when $q_g = p_g$ for all g, and thus

$$\pi_g^{(m+1)} = p(Z_1 = g \,|\, y, \theta^{(m)}) \tag{4.10}$$

$$= \sum_{h=1}^{G} p(Z_1 = g, Z_2 = h \,|\, y, \theta^{(m)}), \tag{4.11}$$

where we expanded (4.10) into (4.11), because Section 4.1.2 has detailed how to compute the term $p(Z_1 = g, Z_2 = h \,|\, y, \theta^{(m)})$.

4.1.4 Estimating the State-Conditional Parameters *b*

The HMM specifies a probability model for an observation given a particular hidden state: $p(Y_t = y_t \,|\, z_t)$, which we have assumed is parameterized by state-conditional parameters b. The model $p(Y_t = y_t \,|\, z_t)$ could be anything: a Laplace distribution, a GMM, even another HMM![3] Here we illustrate how to use EM to learn an HMM with the common choice that y_t takes on one of a finite set of V values, and $p(Y_t = y_t \,|\, z_t)$ is simply a pmf over the V values such that the $G \times V$ parameter set b has components:

$$b_{v,g} = p(Y_t = v \,|\, Z_t = g), \quad v = 1, \ldots, V, \quad g = 1, \ldots, G.$$

[3] In fact, given certain initial conditions for our universe, Jorge Luis Borges writes a short story where everything is generated by an HMM whose state-conditional models are themselves each HMMs, whose state-conditional models are also HMMs, etc., up to the 11th HMM which has, as one of its state-conditional models, the very first HMM.

The M-step for b is:

$$\begin{aligned}
b^{(m+1)} &= \arg\max_b E_{Z|y,\theta^{(m)}}[\log p(y, Z \,|\, \theta)] \\
&= \arg\max_b \sum_z p(z \,|\, y, \theta^{(m)}) \log p(y, z \,|\, \pi, \mathbf{P}, b) \\
&= \arg\max_b \sum_z p(z \,|\, y, \theta^{(m)}) \log (p(y \,|\, z, b) p(z \,|\, \pi, \mathbf{P})) \\
&= \arg\max_b \sum_z p(z \,|\, y, \theta^{(m)}) \log p(y \,|\, z, b) \\
&\quad \text{(drop } p(z \,|\, \pi, \mathbf{P}) \text{ since it does not depend on } b) \\
&= \arg\max_b \sum_z p(z \,|\, y, \theta^{(m)}) \log \prod_{t=1}^{T} p(Y_t = y_t \,|\, Z_t = z_t, b) \\
&= \arg\max_b \sum_z p(z \,|\, y, \theta^{(m)}) \log \prod_{v=1}^{V} \prod_{g=1}^{G} b_{v,g}^{\eta_{vg}(y,z)},
\end{aligned}$$

where $\eta_{vg}(y, z)$ is defined as

$$\eta_{vg}(y, z) \triangleq \sum_{t=1}^{T} 1_{\{y_t = v, z_t = g\}}.$$

We continue the above equation and have

$$\begin{aligned}
b^{(m+1)} &= \arg\max_b \sum_z p(z \,|\, y, \theta^{(m)}) \sum_{v=1}^{V} \sum_{g=1}^{G} \eta_{vg}(y, z) \log b_{v,g} \\
&= \arg\max_b \sum_{g=1}^{G} \left(\sum_{v=1}^{V} \left(\sum_z p(z \,|\, y, \theta^{(m)}) \eta_{vg}(y, z) \right) \log b_{v,g} \right).
\end{aligned}$$

We can solve the last line of the above equation independently for the pmf corresponding to the gth state-conditional model. Again, due to the constraint $\sum_v b_{v,g} = 1$, we can apply Gibbs' inequality as given in (3.4) with

$$\begin{aligned}
q_v &= b_{v,g} \\
p_v &= \frac{\sum_z p(z \,|\, y, \theta^{(m)}) \eta_{vg}(y, z)}{\sum_{l=1}^{V} \sum_z p(z \,|\, y, \theta^{(m)}) \eta_{lg}(y, z)},
\end{aligned}$$

and conclude that the maximum must occur when Gibbs' inequality holds with equality, that is, when $q_v = p_v$ for all v, and thus

$$b_{v,g}^{(m+1)} = \frac{\sum_z p(z \,|\, y, \theta^{(m)}) \eta_{vg}(y,z)}{\sum_{l=1}^{V} \sum_z p(z \,|\, y, \theta^{(m)}) \eta_{lg}(y,z)}. \tag{4.12}$$

Next, we simplify the numerator of (4.12) (the terms in the denominator can be calculated similarly):

$$\begin{aligned}
\sum_z p(z \,|\, y, \theta^{(m)}) \eta_{vg}(y,z) &= E_{Z|y,\theta^{(m)}}[\eta_{vg}(y,Z)] \\
&= E_{Z|y,\theta^{(m)}} \left[\sum_{t=1}^{T} 1_{\{y_t = v, Z_t = g\}} \right] \\
&= \sum_{t=1}^{T} E_{Z|y,\theta^{(m)}}[1_{\{y_t = v, Z_t = g\}}] \\
&= \sum_{t=1}^{T} E_{Z_t|y,\theta^{(m)}}[1_{\{y_t = v, Z_t = g\}}] \\
&= \sum_{t=1}^{T} p(Z_t = g \,|\, y, \theta^{(m)}) 1_{\{y_t = v\}},
\end{aligned}$$

where the term $p(Z_t = g \,|\, y, \theta^{(m)})$ can be expressed as either

$$p(Z_t = g \,|\, y, \theta^{(m)}) = \sum_{h=1}^{G} p(Z_{t-1} = h, Z_t = g \,|\, y, \theta^{(m)}),$$

or

$$p(Z_t = g \,|\, y, \theta^{(m)}) = \sum_{h=1}^{G} p(Z_t = g, Z_{t+1} = h \,|\, y, \theta^{(m)}),$$

and the computation of the term $p(Z_t = g, Z_{t+1} = h \,|\, y, \theta^{(m)})$ is detailed in Section 4.1.2.

4.1.5 More on HMM and EM

For further details on HMM, we recommend Rabiner's tutorial [49] and the review article by Gales and Young [16], which considers the practical application of HMMs in depth. The HMM described above has at

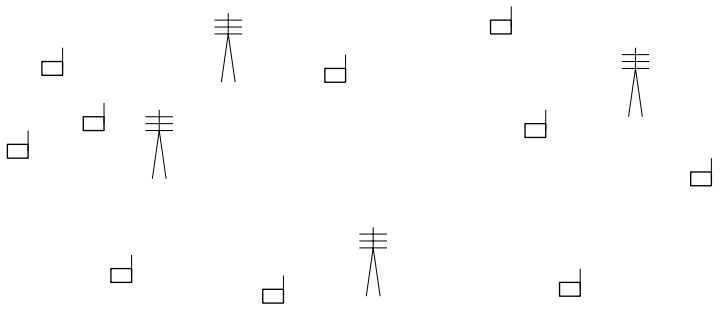

Fig. 4.1 An example set-up with 4 transmitters, and 10 receivers. The problem set-up assumes the locations of the 10 receivers are known, and that one is given a measurement of the received power at each receiver, and the goal is to estimate the location of each of the 4 transmitters.

its core a Markov chain. The two-dimensional (and higher-dimensional) analog of a Markov chain is a Markov random field. Applying EM to a hidden Markov random field model is significantly more troublesome, but is common in image processing; see [48] for details. See also: [19, 25, 26, 29, 31, 65].

4.2 Estimating Multiple Transmitter Locations

Consider the problem of estimating the most likely locations $\theta = \begin{bmatrix}\theta_1 & \theta_2 & \dots & \theta_M\end{bmatrix}$ of M transmitters with $\theta_i \in \mathbb{R}^2$, where we assume the transmitters are transmitting in the same band, and that we are given noisy power measurements $y = \begin{bmatrix}y_1 & y_2 & \dots & y_N\end{bmatrix}$ for N receivers located in the plane at known locations $r = \begin{bmatrix}r_1 & r_2 & \dots & r_N\end{bmatrix}$. This problem arises in cognitive radio [22], and is an illustrative example of a general class of problems that can be solved by EM where the goal is to estimate parameters given superimposed signals (see [12]). This example has been studied by Nelson and Gupta [42]; we present a simplified version.

The basic idea is that first we make a guess at where the transmitters are located: $\theta^{(0)}$. Then we use that guess and the measured total power y at each receiver to compute our best guess of the complete data, which is how much power each of the receivers picked up from each of the transmitters. Given the complete data estimate of how much power

each receiver got from each transmitter, we can independently estimate where each transmitter is located. That gives us a new guess $\theta^{(m+1)}$ of where the transmitters are, and then one iterates.

For simplicity, we assume all transmitters are transmitting one unit of power, and that the Gaussian measurement noise of the ith transmitter at the jth receiver W_{ij} is known to be zero-mean and have variance σ^2/M (a zero-mean Gaussian is not a good model for power noise; an issue we will return to in Section 5.2.3). Let X_{ij} denote the power sent by the ith transmitter and received by the jth receiver; X_{ij} is inversely proportional to the squared distance between the transmitter and receiver plus the Gaussian measurement noise:

$$X_{ij} = \frac{1}{\|\theta_i - r_j\|_2^2} + W_{ij}. \tag{4.13}$$

The observed power y_j at the jth receiver is the total power coming from all the transmitters:

$$y_j = \sum_{i=1}^{M} x_{ij}.$$

Conditioned on the transmitter locations θ and given the receiver locations r, the likelihood of the observed measurements y depends only on the Gaussian noise:

$$p(y\,|\,\theta) = \prod_{j=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \left(y_j - \sum_{i=1}^{M} \frac{1}{\|\theta_i - r_j\|_2^2} \right)^2 \right).$$

Thus the log-likelihood (ignoring the terms that do not depend on θ) is

$$\ell(\theta) = -\sum_{j=1}^{N} \left(y_j - \sum_{i=1}^{M} \frac{1}{\|\theta_i - r_j\|_2^2} \right)^2.$$

Like many problems, this log-likelihood $\ell(\theta)$ has multiple maxima, and while we can apply EM, we should keep in mind that EM will only find a local maximum.

To apply EM, we define the complete data to be the $M \times N$ powers between the ith transmitter and jth receiver, which we formulate as an $MN \times 1$ vector $X = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{MN} \end{bmatrix}^{\mathrm{T}}$ [42]. Next, consider

$\log p(X = x \,|\, \theta^{(m)})$. From (4.13), this probability is Gaussian because all the randomness comes from the Gaussian noise, and ignoring scale factors that will not change the M-step maximization over θ:

$$\log p(X = x \,|\, \theta) = -\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{ij} - \frac{1}{\|\theta_i - r_j\|_2^2}\right)^2. \tag{4.14}$$

Let $R(\theta)$ be the $MN \times 1$ vector with components $\frac{1}{\|\theta_i - r_j\|_2^2}$ ordered corresponding to the order in X. Then the log-likelihood in (4.14) can be expressed as

$$\log p(X = x \,|\, \theta) = -\|x - R(\theta)\|_2^2.$$

Rather than computing the E-step separately, we directly consider the M-step. We drop and add θ-independent terms to make things easier:

$$\begin{aligned}
\theta^{(m+1)} &= \arg\max_{\theta\in\Omega} E_{X|y,\theta^{(m)}}[\log p(X \,|\, \theta)] \\
&= \arg\max_{\theta\in\Omega} E_{X|y,\theta^{(m)}}[-\|X - R(\theta)\|_2^2] \\
&= \arg\min_{\theta\in\Omega} E_{X|y,\theta^{(m)}}[\|X - R(\theta)\|_2^2] \\
&= \arg\min_{\theta\in\Omega} E_{X|y,\theta^{(m)}}[(X - R(\theta))^T(X - R(\theta))] \\
&= \arg\min_{\theta\in\Omega} E_{X|y,\theta^{(m)}}[X^TX - 2R(\theta)^TX + R(\theta)^TR(\theta)] \\
&= \arg\min_{\theta\in\Omega} -2R(\theta)^TE_{X|y,\theta^{(m)}}[X] + R(\theta)^TR(\theta) \\
&= \arg\min_{\theta\in\Omega} E_{X|y,\theta^{(m)}}[X]^TE_{X|y,\theta^{(m)}}[X] - 2R(\theta)^TE_{X|y,\theta^{(m)}}[X] \\
&\quad + R(\theta)^TR(\theta) \\
&= \arg\min_{\theta\in\Omega} \|E_{X|y,\theta^{(m)}}[X] - R(\theta)\|_2^2.
\end{aligned} \tag{4.15}$$

Note that above we have massaged the need to compute the expected log-likelihood $E_{X|y,\theta^{(m)}}[\log p(X \,|\, \theta)]$ (that is, the E-step) to simply computing our current best guess of X, that is, $E_{X|y,\theta^{(m)}}[X]$. In order to compute $E_{X|y,\theta^{(m)}}[X]$, recall that it is the expectation of the MN received powers at each of the N receivers from each of the M transmitters, conditioned on knowing the total power at each receiver y and a guess of the transmitter locations $\theta^{(m)}$. If we only conditioned on

$\theta^{(m)}$, then because the noise W_{ij} on each component of the vector X is independent and zero-mean, $E_{X|\theta^{(m)}}[X]$ would be $R(\theta^{(m)})$ from (4.13). However, because of the additional conditioning on y, each component must be adjusted equally (since W_{ij} is additive and i.i.d.) so that the sum of our new guesses for the M individual powers for the jth receiver totals the observed power y_j for the jth receiver:

$$E_{X_{ij}|y,\theta^{(m)}}[X_{ij}] = R_{ij}(\theta^{(m)}) + \frac{1}{M}\left(y_j - \sum_{i=1}^{M} R_{ij}(\theta^{(m)})\right). \tag{4.16}$$

After calculating $E_{X|y,\theta^{(m)}}[X]$ using (4.16), each iteration's M-step given by (4.15) can be decomposed into estimating the ith transmitter's location independently for $i = 1, \ldots, M$:

$$\theta_i^{(m+1)} = \arg\min_{\theta_i \in \mathbb{R}^2} \sum_{j=1}^{N} (E_{X_{ij}|y,\theta^{(m)}}[X_{ij}] - R_{ij}(\theta_i))^2. \tag{4.17}$$

Note that solving (4.17) is not trivial as the objective function is not a convex function of θ_i. However, by using EM we have reduced the original non-convex likelihood maximization over $2M$ variables θ to iteratively solving M easier two-dimensional optimization problems specified by (4.17).

4.3 Estimating a Compound Dirichlet Distribution

In this section, we detail another popular example of applying EM with missing data: using EM to find the maximum likelihood estimate of the parameter of a *compound Dirichlet* distribution, which is also referred to as the *Pólya* distribution. First, we give a brief introduction to the compound Dirichlet distribution; for a more comprehensive introduction to the Dirichlet distribution, see [14].

The Dirichlet distribution is commonly used to model random probability mass functions (pmfs). For example, if someone hands you a coin, you would not know the coin's bias, and you could consider it a random coin, that is, one that has a random pmf over the sample space of *heads* and *tails*. You might have some idea of how likely the coin is to have different biases — for example, if you pick a 2010 penny off the

ground you might be fairly certain the coin will be close to a fair coin, with equal probability of being a little biased toward heads or tails depending on its wear. This knowledge could be modeled with a *beta* distribution, which specifies the distribution over possible biases of a given coin. The beta distribution is a model for the distribution of a random pmf if there are only two events, and the Dirichlet distribution is a generalization for modeling the distribution of a random pmf over *any* finite number of events.[4] For example, a six-sided die that you pick up at a casino can be modeled as a random pmf over six events using the Dirichlet distribution.

The Dirichlet distribution has one vector parameter: $\alpha \in \mathbb{R}_+^d$. If all the d components of α are greater than 1, the Dirichlet distribution is unimodal over the probability simplex. If all the components of α are less than 1, the Dirichlet distribution has peaks at the vertices of the probability simplex. Given a random pmf $V \sim \text{Dir}(\alpha)$, its expected pmf $E[V]$ is the normalization of the parameter α, that is, the jth component of the mean pmf is $(E[V])_j = \alpha_j / \alpha_0$ where $\alpha_0 = \sum_{k=1}^{d} \alpha_k$.

Given sample pmfs known to be drawn i.i.d. from a Dirichlet distribution, one could estimate α using maximum likelihood estimation for the underlying Dirichlet. More often in practice, and an interesting example of EM, is instead the case that the observed data are i.i.d. samples that have been drawn from pmfs that have been drawn i.i.d. from a Dirichlet distribution:

$$\text{Dir}(\alpha) \xrightarrow{\text{i.i.d.}} \begin{array}{ccc} \text{pmf } z_1 & \xrightarrow{\text{i.i.d.}} & \text{samples from } z_1 \\ \text{pmf } z_2 & \xrightarrow{\text{i.i.d.}} & \text{samples from } z_2 \\ \vdots & & \vdots \\ \text{pmf } z_n & \xrightarrow{\text{i.i.d.}} & \text{samples from } z_n \end{array}$$

For example, we could model the weather each day in April in Paris as an event from the sample space {rainy, cloudy, sunny}, and assume that the daily weather is a realization of a daily weather pmf z_i, and

[4] Here we only deal with the Dirichlet distribution which assumes the number of events is finite, but the Dirichlet distribution has a more general form called the Dirichlet process, which is a measure over measures over infinite sample spaces.

that each daily weather pmf z_i is drawn i.i.d. from some Dirichlet distribution $\mathrm{Dir}(\alpha)$ over possible weather pmfs. Then at the end of April we would have observed 30 samples of the weather, and we could attempt to find the maximum likelihood estimate of the parameter α for the Dirichlet that generated the weather pmfs that generated the 30 days of observed weather. In this example, we only generated one sample from each pmf, but in general we may have many samples known to be drawn from the ith pmf, and then the corresponding observed data y_i is taken to be the empirical histogram over the sample space:

$$\mathrm{Dir}(\alpha) \xrightarrow{\text{i.i.d.}} \begin{array}{ccccc} \text{pmf } z_1 & \xrightarrow{\text{i.i.d.}} & \text{samples from } z_1 & \xrightarrow{\text{count}} & \text{histogram } y_1 \\ \text{pmf } z_2 & \xrightarrow{\text{i.i.d.}} & \text{samples from } z_2 & \xrightarrow{\text{count}} & \text{histogram } y_2 \\ \vdots & & \vdots & & \vdots \\ \text{pmf } z_n & \xrightarrow{\text{i.i.d.}} & \text{samples from } z_n & \xrightarrow{\text{count}} & \text{histogram } y_n \end{array}$$

The distribution of i.i.d. samples y drawn from pmfs drawn i.i.d. from a Dirichlet distribution is the *compound Dirichlet* distribution, also called the *multivariate Pólya* distribution. Let the given data y be an $n \times d$ matrix of n sample histograms, each over d possible events, such that the ith row vector y_i is the ith histogram for $i = 1, \ldots, n$, and y_{ij} is the number of times we have observed the jth event from samples drawn from the ith pmf z_i, where z_{ij} denotes the probability of observing the jth event given the ith pmf. Namely, y_i has a multinomial distribution with parameter z_i such that

$$p(y_i \,|\, z_i) = \frac{\left(\sum_{j=1}^d y_{ij}\right)!}{\prod_{j=1}^d y_{ij}!} \prod_{j=1}^d z_{ij}^{y_{ij}}.$$

Let z_i be a realization of a random pmf $Z_i \in \mathcal{S}$, where $\mathcal{S}$ is the $(d-1)$-dimensional probability simplex such that $\sum_{j=1}^d Z_{ij} = 1$ and $Z_{ij} > 0$, $j = 1, \ldots, d$. The random pmf Z_i is assumed to have a Dirichlet distribution with parameter α such that

$$p(Z_i = z_i \,|\, \alpha) = \frac{\Gamma\left(\sum_{j=1}^d \alpha_j\right)}{\prod_{j=1}^d \Gamma(\alpha_j)} \prod_{j=1}^d z_{ij}^{\alpha_j - 1}. \tag{4.18}$$

Then, if $z_1, \ldots, z_n$ were drawn i.i.d. from a Dirichlet with parameter α, the probability of seeing all the n corresponding histograms $y_1, \ldots, y_n$

is the following *compound Dirichlet* distribution (also called the *multivariate Pólya* distribution):

$$\begin{aligned} p(y\,|\,\alpha) &= \prod_{i=1}^{n} \int_{\mathcal{S}} p(y_i\,|\,z_i) p(z_i\,|\,\alpha) dz_i \\ &= \prod_{i=1}^{n} \frac{\left(\sum_{j=1}^{d} y_{ij}\right)! \Gamma\left(\sum_{j=1}^{d} \alpha_j\right)}{\left(\prod_{j=1}^{d} y_{ij}!\right)\left(\prod_{j=1}^{d} \Gamma(\alpha_j)\right)} \int_{\mathcal{S}} \prod_{j=1}^{d} z_{ij}^{y_{ij}+\alpha_j-1} dz_i \\ &= \prod_{i=1}^{n} \frac{\left(\sum_{j=1}^{d} y_{ij}\right)! \Gamma\left(\sum_{j=1}^{d} \alpha_j\right) \prod_{j=1}^{d} \Gamma(\alpha_j + y_{ij})}{\left(\prod_{j=1}^{d} y_{ij}!\right)\left(\prod_{j=1}^{d} \Gamma(\alpha_j)\right) \Gamma\left(\sum_{j=1}^{d} (\alpha_j + y_{ij})\right)}. \end{aligned}$$

Given y, we describe how to use EM to find the maximum likelihood of the parameter α. However, neither the likelihood $p(y\,|\,\alpha)$ nor its log-likelihood is concave, and EM is not guaranteed to find the global maximum likelihood solution.

To apply the EM method here, consider the missing data to be the pmfs $\{z_i\}_{i=1}^{n}$ that generated the observed data y such that the complete data is $x = (y, z)$, $z = \{z_i\}_{i=1}^{n}$. We will search for the parameter α that maximizes that the expected log-likelihood of x. This use of EM fits the missing data paradigm described in Section 1.4.1, and we can use (1.5) to express the Q-function. We also use the assumption that Z_i are independent. Then, the M-step is:

$$\begin{aligned} \alpha^{(m+1)} &= \arg\max_{\alpha \in \mathbb{R}_+^d} \sum_{i=1}^{n} E_{Z_i|y_i,\alpha^{(m)}}[\log p(y_i, Z_i\,|\,\alpha)] \\ &= \arg\max_{\alpha \in \mathbb{R}_+^d} \sum_{i=1}^{n} E_{Z_i|y_i,\alpha^{(m)}}[\log(p(y_i\,|\,Z_i, \alpha) p(Z_i\,|\,\alpha))] \\ &= \arg\max_{\alpha \in \mathbb{R}_+^d} \sum_{i=1}^{n} E_{Z_i|y_i,\alpha^{(m)}}[\log(p(y_i\,|\,Z_i) p(Z_i\,|\,\alpha))] \\ &= \arg\max_{\alpha \in \mathbb{R}_+^d} \sum_{i=1}^{n} E_{Z_i|y_i,\alpha^{(m)}}[\log p(Z_i\,|\,\alpha)]. \end{aligned} \tag{4.19}$$

Note that the Q-function given in (4.19) is concave because $\log p(z_i\,|\,\alpha)$ is concave by a theorem of Ronning [54], and (4.19) is a finite integration of such concave functions and hence also concave.

Applying (4.18) to (4.19) and noting that the first two terms can be pulled out of the expectation, the M-step becomes:

$$\alpha^{(m+1)} = \arg\max_{\alpha \in \mathbb{R}_+^d} n \log \left(\frac{\Gamma(\alpha_0)}{\prod_{j=1}^d \Gamma(\alpha_j)} \right) + \sum_{i=1}^n E_{Z_i | y_i, \alpha^{(m)}} \left[\sum_{j=1}^d (\alpha_j - 1) \log Z_{ij} \right]. \tag{4.20}$$

Since expectation is linear, the second term on the right-hand side of (4.20) can be written as

$$\sum_{i=1}^n \sum_{j=1}^d (\alpha_j - 1) \int_{\mathcal{S}} \log(z_{ij}) p(z_i \,|\, y_i, \alpha^{(m)}) dz_i, \tag{4.21}$$

where the probability $p(z_i \,|\, y_i, \alpha^{(m)})$ is in fact itself a Dirichlet distribution because the Dirichlet distribution is a conjugate prior for the multinomial distribution. To be explicit,

$$\begin{aligned} p(z_i \,|\, y_i, \alpha^{(m)}) &= \frac{p(y_i, z_i \,|\, \alpha^{(m)})}{p(y_i \,|\, \alpha^{(m)})} \\ &= \frac{p(y_i \,|\, z_i) p(z_i \,|\, \alpha^{(m)})}{p(y_i \,|\, \alpha^{(m)})} \\ &= \gamma(y_i, \alpha^{(m)}) \prod_{j=1}^d z_{ij}^{y_{ij}} \prod_{j=1}^d z_{ij}^{\alpha_j^{(m)} - 1} \\ &\qquad \text{(where } \gamma(y_i, \alpha^{(m)}) \text{ is a normalizer independent of } z_i\text{)} \\ &= \gamma(y_i, \alpha^{(m)}) \prod_{j=1}^d z_{ij}^{y_{ij} + \alpha_j^{(m)} - 1}, \end{aligned}$$

which is a Dirichlet distribution with parameter $y_i + \alpha^{(m)}$. Thus, the integral in (4.21) is the expected log of the jth component of a pmf drawn from a Dirichlet with parameter $y_i + \alpha^{(m)}$.

To compute this expected log, we consider the general case where $V \sim \text{Dir}(\alpha)$, and derive $E[\log V_j]$, $j = 1, \ldots, d$. To that end, it is useful to recall that the general form of a distribution in the exponential family

with parameter α is

$$p(v\,|\,\alpha) = h(v)\exp(\eta(\alpha)\cdot T(v) - A(\alpha)),$$

where $\cdot$ denotes the standard inner product, $v \in \mathbb{R}^r$, $\alpha \in \mathbb{R}^s$, $h\colon \mathbb{R}^r \to \mathbb{R}$, $\eta\colon \mathbb{R}^s \to \mathbb{R}^k$ (called the *natural parameter*), $T\colon \mathbb{R}^r \to \mathbb{R}^k$ (called the *sufficient statistic* of the distribution), $A\colon \mathbb{R}^s \to \mathbb{R}$ (called the *normalization factor*), and $r, s, k \in \mathbb{N}$. The Dirichlet distribution is a member of the exponential family with $r = s = k = d$, $h(v) \equiv 1$, $\eta(\alpha) = \alpha - \mathbf{1}$ (where $\mathbf{1}$ is a vector of ones), $T(v) = \log v$, and $A(\alpha) = \sum_{j=1}^{d} \log\Gamma(\alpha_j) - \log\Gamma(\alpha_0)$, that is, the Dirichlet density can be written as

$$\begin{aligned} p(v\,|\,\alpha) &= \frac{\Gamma(\alpha_0)}{\prod_{j=1}^{d}\Gamma(\alpha_j)}\prod_{j=1}^{d} v_j^{\alpha_j - 1} \\ &= \exp\left(\sum_{j=1}^{d}(\alpha_j - 1)\log v_j - \left(\sum_{j=1}^{d}\log\Gamma(\alpha_j) - \log\Gamma(\alpha_0)\right)\right). \end{aligned}$$

We will need the following identity for the Dirichlet:

$$1 = \int_{\mathcal{S}} p(v\,|\,\alpha)dv = \int_{\mathcal{S}} e^{(\alpha-\mathbf{1})\cdot\log v - A(\alpha)}dv = e^{-A(\alpha)}\int_{\mathcal{S}} e^{(\alpha-\mathbf{1})\cdot\log v}dv,$$

and therefore,

$$e^{A(\alpha)} = \int_{\mathcal{S}} e^{(\alpha-\mathbf{1})\cdot\log v}dv. \tag{4.22}$$

We can produce $E[\log V_j]$ by starting with the moment-generating function $M\colon \mathbb{R}^d \to \mathbb{R}$ for the sufficient statistics $T(v) = \log v$ given α:

$$\begin{aligned} M(u) &= E_V[e^{u\cdot T(V)}] \\ &= \int_{\mathcal{S}} e^{u\cdot T(v)} p(v\,|\,\alpha)dv \\ &= \int_{\mathcal{S}} e^{u\cdot\log v} e^{(\alpha-\mathbf{1})\cdot\log v - A(\alpha)}dv \\ &= e^{-A(\alpha)}\int_{\mathcal{S}} e^{(u+\alpha-\mathbf{1})\cdot\log v}dv \\ &= e^{A(u+\alpha)-A(\alpha)}, \end{aligned}$$

where the last line follows from (4.22). Then the partial derivative of $M(u)$ with regard to u_j is

$$\frac{\partial}{\partial u_j} M(u) = \frac{\partial}{\partial u_j} E_V[e^{u \cdot T(V)}] \tag{4.23}$$

$$= \frac{\partial}{\partial u_j} e^{A(u+\alpha)-A(\alpha)}$$

$$= (e^{A(u+\alpha)-A(\alpha)}) \frac{\partial A(u+\alpha)}{\partial u_j}. \tag{4.24}$$

But we can interchange the expectation and the differentiation of (4.23) (by Theorem 2.27 of [13]), so we also have

$$\frac{\partial}{\partial u_j} M(u) = E_V \left[\frac{\partial}{\partial u_j} e^{u \cdot T(V)} \right]$$

$$= E_V[(\log V_j) e^{u \cdot \log V}]. \tag{4.25}$$

Setting $u = 0$ in the equivalent (4.25) and (4.24) produces the expected log of V_j that we need:

$$E_V[\log V_j] = \left. \frac{\partial A(u+\alpha)}{\partial u_j} \right|_{u=0} = \psi(\alpha_j) - \psi(\alpha_0),$$

where ψ is the digamma function:

$$\psi(x) \triangleq \frac{d}{dx} \log \Gamma(x).$$

Finally, we see that the integral in (4.21), which is the expected log of the jth component of a pmf drawn from a Dirichlet with parameter $y_i + \alpha^{(m)}$, is

$$E_{Z_i|y_i,\alpha^{(m)}}[\log Z_{ij}] = \psi(y_{ij} + \alpha_j^{(m)}) - \psi\left(\sum_{l=1}^{d} y_{il} + \alpha_0^{(m)} \right).$$

In summary, EM repeatedly solves:

$$\alpha^{(m+1)} = \arg\max_{\alpha \in \mathbb{R}_+^d} g(\alpha), \tag{4.26}$$

where

$$g(\alpha) = n \log \Gamma(\alpha_0) - n \sum_{j=1}^{d} \log \Gamma(\alpha_j) + \sum_{i=1}^{n} \sum_{j=1}^{d} (\alpha_j - 1) \left(\psi(y_{ij} + \alpha_j^{(m)}) - \psi \left(\sum_{l=1}^{d} y_{il} + \alpha_0^{(m)} \right) \right).$$

In order to execute the M-step, one can solve (4.26) anyway one likes, but a standard approach is to apply Newton's method [5], for which we need the gradient and Hessian matrix of $g(\alpha)$. By definition of the digamma function ψ, the gradient of $g(\alpha)$ is

$$\nabla g(\alpha) = \begin{bmatrix} \rho_1(\alpha) & \dots & \rho_d(\alpha) \end{bmatrix}^{\mathrm{T}},$$

where for $j = 1, \dots, d$,

$$\rho_j(\alpha) = n\psi(\alpha_0) - n\psi(\alpha_j) + \sum_{i=1}^{n} \left(\psi(y_{ij} + \alpha_j^{(m)}) - \psi \left(\sum_{l=1}^{d} y_{il} + \alpha_0^{(m)} \right) \right).$$

Then using the definition of the trigamma function,

$$\psi_1(x) \triangleq \frac{d}{dx} \psi(x),$$

the Hessian matrix of $g(\alpha)$ is

$$H(\alpha) = n\psi_1(\alpha_0) \mathbf{1}\mathbf{1}^T - n \operatorname{diag}(\psi_1(\alpha_1), \dots, \psi_1(\alpha_d)),$$

where $\mathbf{1}$ is a vector of ones so that $\mathbf{1}\mathbf{1}^{\mathrm{T}}$ is a $d \times d$ matrix of ones, and $\operatorname{diag}(\cdot)$ is a matrix with its argument on the diagonal and zeros elsewhere.

Newton's method is an iterative algorithm, and here for solving (4.26), each iteration takes the following update step:

$$\alpha \leftarrow \alpha - tH^{-1}(\alpha) \nabla g(\alpha),$$

where $t > 0$ is the step size. The geometric interpretation of the above update step can be found in [5]. Note that here inverting $H(\alpha)$ is not as problematic as it might appear because this Hessian matrix has a

very nice structure that simplifies the inversion using the Woodbury identity [47]. Let $\xi \in \mathbb{R}^d$ have jth component:

$$\xi_j = \frac{1}{\psi_1(\alpha_j)},$$

for $j = 1, \ldots, d$, and let

$$\xi_0 = \frac{1}{\psi_1(\alpha_0)},$$

then

$$H^{-1}(\alpha) = -\frac{1}{n}\operatorname{diag}(\xi) - \frac{1}{n(\xi_0 - \mathbf{1}^T\xi)}\xi\xi^{\mathrm{T}}.$$

5

EM Variants

EM produces convenient solutions for many simple problems, but,

(1) EM only finds stationary points of the likelihood function;
(2) the computations required may not be computationally tractable;
(3) the convergence may be too slow;
(4) the maximum likelihood estimate may not be the desired output.

Many variants of EM exist to address subsets of these issues. We have already touched on two variants of EM: MAP EM in Section 1.3, and the point-estimate EM described in Section 1.2. In this section we describe other variants that may be useful, categorized by which of the four above problems the variant best addresses.

5.1 EM May Not Find the Global Optimum

EM is a handy tool, but if the log-likelihood is not concave, one run of EM cannot be trusted to find the optimal solution. Non-concavity is very common in practical problems; for example, the log-likelihoods for the GMM and HMM are usually *not* concave.

The simplest approach to dealing with non-concavity is to run EM with 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 efficiently: 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 [1, 23, 24, 28, 38, 46].

5.2 EM May Not Simplify the Computation

We have seen that instead of solving the potentially difficult problem of directly maximizing $\ell(\theta)$, the EM algorithm chooses to repeatedly maximize $Q(\theta\,|\,\theta^{(m)})$, but sometimes this maximization problem is still difficult. When EM does not provide simple solutions, the variants in this section may be useful.

5.2.1 Generalized EM (GEM)

GEM is a popular variant of EM in which the Q-function is only improved at each iteration but not necessarily maximized [30]. That is, at the $(m+1)$th iteration, one finds a $\theta^{(m+1)} \in \Omega$ that satisfies

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

By Theorem 2.1, the GEM algorithm retains the monotonicity property.

5.2.2 Monte Carlo Alternatives to EM

EM is best when the distributions are nice and give rise to a simple form for the Q-function. However, when that is not the case, Monte Carlo sampling methods may be needed to approximate the E-step, or it might be better to toss aside the EM algorithm and use Monte Carlo sampling to approximate the posterior mode (or posterior mean) of θ directly. For further reading on Monte Carlo sampling and particularly Markov Chain Monte Carlo (MCMC), we recommend the introductory material in [34], which is available online, and the more comprehensive book on MCMC (which specifically discusses EM) by Robert and Casella [52].

5.2.3 Quasi-EM

Quasi-EM is a variant of EM that simplifies the problem, finds the EM solution for the simpler problem, and applies the same idea to the original complicated problem [43]. For example, as derived in Section 3, fitting a GMM alternates between two tasks: (i) estimating the parameters of the component models, and (ii) estimating the relative likelihood that each sample was generated by each model. If the component models are not Gaussian, then alternating between these two tasks may not actually be the EM solution, but may still be a practical approach to finding a useful solution.

As a second example, consider the transmitter-localization example given in Section 4.2, a more realistic noise model than the additive white Gaussian noise model given in (4.13) is a lognormal shadowing model [18], a simplified illustrative version of which is:

$$Z_{ij} = \frac{1}{\|\theta_i - r_j\|_2^2} 10^{W_{ij}},$$

where $W_{ij} \sim \mathcal{N}(0, \sigma^2)$ models the random shadowing. Then the likelihood function of Z is a product of lognormal densities of the form in (4.13), and the log-likelihood needed for EM is a sum of lognormal densities. However, there is no analytic form for a sum of lognormal densities. One could use a Monte Carlo approach to generate random samples to compute an approximation of the log-likelihood, but generating random samples is computationally intensive (and removes the guarantee that EM will converge).

However, consider the intuition behind the simpler Gaussian noise model for the transmitter-localization problem as covered in Section 4.2. The EM algorithm alternated between (i) estimating the transmitter locations based on the current guess of how much of the received power came from each transmitter, and (ii) using the current estimate of the transmitter locations to guess how much of the received power came from each transmitter. Nelson et al. [43] showed that using the same alternation with the more complicated lognormal shadowing model was 10 times more accurate at estimating the transmitter locations than making the same number of guesses with a state-of-the-art

global optimizer (particle swarm optimization [38]), and 50 times more accurate for the same number of guesses than random guessing.

5.3 Speed

As we touched on in Section 2.1, the EM algorithm has relatively slow convergence compared to numerical optimization approaches like Newton–Raphson updates. Many variants have been proposed to attempt to speed up EM convergence, though these tend to lose the simplicity of EM without achieving the theoretical convergence speed-up of Newton–Raphson. Further, as noted in Section 2.1, it is often difficult before you run an algorithm for a specific problem to know whether the convergence speed-up gain iteration-by-iteration of a variant is worth the increased computation for each iteration.

Surveys of variants for speeding up convergence can be found in the book by McLachlan and Krishnan [36] and in the tutorial by Roche [53].

5.4 When Maximizing the Likelihood Is Not the Goal

EM is designed to find an estimate of θ that maximizes the likelihood $p(y\,|\,\theta)$. However, the maximum likelihood estimate may not be the best estimate. For example, another popular estimate for θ is the posterior mean $E_{\Theta|y}[\Theta]$. The posterior mean is the best estimate in the sense that it minimizes the expected posterior squared-error loss, and in fact minimizes the expectation of any of the Bregman divergences [3, 15].

In this section, we describe some stochastic variants of EM, leading up to the *data augmentation method*, which provides an estimate of the full posterior distribution, which can be used to find the posterior mean.

5.4.1 Randomizing the E-step

In Section 1.2, we discussed the point-estimate variant of EM where in an E-like step the hidden data is estimated, for example taking the maximum likelihood estimate of x. A stochastic variant [8] is that in the E-like step a random sample $x^{(m)}$ is drawn from $p(x\,|\,y,\theta^{(m)})$, which

is then used in the M-step:

$$\begin{aligned} &\text{Stochastic E-step:} && X^{(m)} \sim p(x \,|\, y, \theta^{(m)}) \\ &\text{Deterministic M-step:} && \theta^{(m+1)} = \arg\max_{\theta} p(x^{(m)} \,|\, \theta). \end{aligned}$$

The sequence of estimates $\{\theta^{(m)}\}$ will generally not converge to a specific value, but rather to a stationary pdf [8]. One can use this method to generate candidate θ's and choose the most likely.

5.4.2 Monte Carlo EM

In Monte Carlo EM [60], one maximizes in the M-step an estimated Q-function $\hat{Q}$, created with random draws:

$$\hat{Q}(\theta \,|\, \theta^{(m)}) = \frac{1}{J} \sum_{j=1}^{J} \log p(x^{(m,j)} \,|\, \theta),$$

where $x^{(m,j)}$ is the jth random i.i.d. draw of X with distribution $p(x \,|\, y, \theta^{(m)})$. For $J = 1$, this degenerates to the stochastic EM method described in Section 5.4.1. As $J \to \infty$, this converges almost surely to the M-step of the standard EM. By increasing J as the iteration index m increases, the greater randomness in the early iterations means that this method does not necessarily lock into the initial guess's local maxima, but as long as $J \to \infty$, eventually local convergence will hold.

After reading the next subsection on data augmentation, the reader may understand why the original Monte Carlo EM paper [60] was subtitled "poor man's data augmentation."

5.4.3 Data Augmentation

In the stochastic EM method described in Section 5.4.1 and the above Monte Carlo EM, one *only* randomly draws the complete data x. What happens if one also makes a random draw of θ in the M-step? That is, one alternates between (i) drawing J i.i.d. random samples $\{x^{(m,j)}\}_{j=1}^{J}$

of the complete data, and (ii) drawing one random sample of θ:

$$\text{Stochastic Imputation step:} \quad X^{(m,j)} \sim p(x \,|\, y, \theta^{(m)}), \quad j = 1, \ldots, J$$

$$\text{Stochastic Posterior step:} \quad \Theta^{(m+1)} \sim \frac{1}{J} \sum_{j=1}^{J} p(\theta \,|\, x^{(m,j)}).$$

The above was proposed as the *data augmentation algorithm* [57].[1]

Iterating the stochastic imputation and posterior steps does not explicitly produce the maximum likelihood estimate of θ, but instead produces an estimate of the entire distribution of θ given y:

$$\hat{p}^{(m)}(\theta \,|\, y) = \frac{1}{J} \sum_{j=1}^{J} p(\theta \,|\, x^{(m,j)}).$$

As any Bayesian might tell you, it is much better to have a good guess for the whole distribution than just a good guess at a local peak. In particular, having a guess for the whole distribution makes it easy to estimate the posterior mean.

Data augmentation is useful for problems where it is not easier to work with $p(\theta \,|\, x)$ and $p(x \,|\, \theta, y)$ than $p(\theta \,|\, y)$. Data augmentation was designed to be a random approximation to carrying out successive iterations of a Markov chain that has the true $p(\theta \,|\, y)$ as its stationary distribution [57] (this is rather beautifully explained by the originators [57], and we recommend reading this source to enjoy the full details). This clever design makes $\hat{p}^{(m)}(\theta \,|\, y)$ converge linearly under rather broad conditions to the true $p(\theta \,|\, y)$ [57]. For more on data augmentation, see also [58].

[1] The term *data augmentation* is also used to mean any estimation method that specifies augmented data x [58], including the EM algorithm.

6

Conclusions and Some Historical Notes

We have focused this work on the aspects and applications of EM that we think best illustrate its power, usefulness, and weaknesses. A treatment this short is necessarily incomplete, but we hope this text gives readers a solid foundation from which to further explore the theory, applications, and practical implementation of EM.

EM was formalized as an approach to solving arbitrary maximum likelihood problems and named *EM* in a seminal 1977 paper by Dempster et al. [11]. However, the history of EM is much messier than this. Part of the confusion is that for various specific problems, researchers independently arrived at the same solution that one obtains using EM before 1977. For example, in 1958 Hartley presented the main ideas of EM, rooted in the special case of count data [20]. Similarly, Baum et al. and Welch developed an algorithm for fitting hidden Markov models (HMMs) that is often called the Baum–Welch algorithm, which is equivalent to applying the EM algorithm, and in this context the ideas of EM date back to the 1970 paper by Baum et al. [4, 61]. Another notable instance of a special case of the EM algorithm is the Richardson–Lucy image deconvolution of the early 1970s [51, 33]. Meng and Van Dyk [40] have traced back the ideas of

EM to 1886 [44], and we refer the reader to their paper and MacLachlan's book [36] for more complete historical discussions.

Today, EM and its variants are regularly used to solve a broad range of today's estimation problems, from the *multiple EM for motif elicitation* (MEME) algorithm for motif-finding in DNA squences [2], to fitting mixture models to disambiguate targets from clutter in radar [59]. We hope that you, too, will find EM useful.

Acknowledgments

This work was supported by the United States Office of Naval Research and the United States PECASE award. We thank the following people for their suggestions and proofreading: Raman Arora, Ji Cao, Sergey Feldman, Bela Frigyik, Eric Garcia, Robert M. Gray, Adam Gustafson, Evan Hanusa, Kevin Jamieson, Amol Kapila, Jill Nelson, Nicole Nichols, Mikyoung Park, Nathan Parrish, Tien Re, Peter Sadowski, Eric Swanson, and Kristi Tsukida.

References

[1] 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, no. 4, pp. 635–672, April 2005.

[2] T. L. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Machine Learning*, vol. 21, pp. 51–80, 1995.

[3] A. Banerjee, X. Guo, and H. Wang, "On the optimality of conditional expectation as a Bregman predictor," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2664–2669, July 2005.

[4] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, February 1970.

[5] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[6] 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.

[7] P. Bryant and J. A. Williamson, "Asymptotic behavior of classification maximum likelihood estimates," *Biometrika*, vol. 65, no. 2, pp. 273–281, 1978.

[8] G. Celeux and J. Diebolt, "The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem," *Computational Statistics Quaterly*, vol. 2, pp. 73–82, 1985.

[9] G. Celeux and G. Govaert, "A classification EM algorithm for clustering and two stochastic versions," *Computational Statistics Data Analysis*, vol. 14, pp. 315–332, 1992.

[10] Y. Chen and J. Krumm, "Probabilistic modeling of traffic lanes from GPS traces," in *Proceedings of 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010.

[11] 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.

[12] M. Feder and E. Weinstein, "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 4, pp. 477–489, April 1988.

[13] G. B. Folland, *Real Analysis: Modern Techniques and Their Applications*. New York, NY: John Wiley & Sons, 2nd Edition, 1999.

[14] B. A. Frigyik, A. Kapila, and M. R. Gupta, "Introduction to the Dirichlet distribution and related processes," Department of Electrical Engineering, University of Washignton, UWEETR-2010-0006, 2010.

[15] B. A. Frigyik, S. Srivastava, and M. R. Gupta, "Functional Bregman divergence and Bayesian estimation of distributions," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 5130–5139, November 2008.

[16] M. Gales and S. Young, "The application of hidden Markov models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

[17] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1991.

[18] A. Goldsmith, *Wireless Communications*. Cambridge, UK: Cambridge University Press, 2005.

[19] M. I. Gurelli and L. Onural, "On a parameter estimation method for Gibbs-Markov random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 4, pp. 424–430, April 1994.

[20] H. O. Hartley, "Maximum likelihood estimation from incomplete data," *Biometrics*, vol. 14, no. 2, pp. 174–194, June 1958.

[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer, 2nd Edition, 2009.

[22] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, February 2005.

[23] 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.

[24] 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.

[25] T. Hebert and R. Leahy, "A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors," *IEEE Transactions on Medical Imaging*, vol. 8, no. 2, pp. 194–202, June 1989.

[26] T. J. Hebert and K. Lu, "Expectation–maximization algorithms, null spaces, and MAP image restoration," *IEEE Transactions on Image Processing*, vol. 4, no. 8, pp. 1084–1095, August 1995.

[27] M. Jamshidian and R. I. Jennrich, "Acceleration of the EM algorithm by using quasi-Newton methods," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 59, no. 3, pp. 569–587, 1997.

[28] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, December 2001.

[29] K. Lange, "Convergence of EM image reconstruction algorithms with Gibbs smoothing," *IEEE Transactions on Medical Imaging*, vol. 9, no. 4, pp. 439–446, December 1990.

[30] K. Lange, "A gradient algorithm locally equivalent to the EM algorithm," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 57, no. 2, pp. 425–437, 1995.

[31] J. Li and R. M. Gray, *Image Segmentation and Compression Using Hidden Markov Models*. New York, NY: Springer, 2000.

[32] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, First published in 1957 as a Bell Labs technical note, 1982.

[33] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astronomical Journal*, vol. 79, no. 6, pp. 745–754, June 1974.

[34] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, UK: Cambridge University Press, 2003.

[35] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[36] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York, NY: John Wiley & Sons, 2nd Edition, 2008.

[37] G. J. McLachlan and D. Peel, *Finite Mixture Models*. New York, NY: John Wiley & Sons, 2000.

[38] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, June 2004.

[39] X.-L. Meng and D. B. Rubin, "On the global and componentwise rates of convergence of the EM algorithm," *Linear Algebra and its Applications*, vol. 199, pp. 413–425, March 1994.

[40] X.-L. Meng and D. A. van Dyk, "The EM algorithm — an old folk-song sung to a fast new tune," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 59, no. 3, pp. 511–567, 1997.

[41] 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, November 1998.

[42] J. K. Nelson and M. R. Gupta, "An EM technique for multiple transmitter localization," in *Proceedings of the 41st Annual Conference on Information Sciences and Systems*, pp. 610–615, 2007.

[43] J. K. Nelson, M. R. Gupta, J. Almodovar, and W. H. Mortensen, "A quasi EM method for estimating multiple transmitter locations," *IEEE Signal Processing Letters*, vol. 16, no. 5, pp. 354–357, May 2009.

[44] S. Newcomb, "A generalized theory of the combination of observations so as to obtain the best result," *American Journal of Mathematics*, vol. 8, no. 4, pp. 343–366, August 1886.

[45] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY: Springer, 2nd Edition, 2006.

[46] P. M. Pardalos and H. E. Romeijn, eds., *Handbook of Global Optimization*. Vol. 2, Norwell, MA: Kluwer, 2002.

[47] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*. November 2008. `http://matrixcookbook.com/`.

[48] W. Qian and D. M. Titterington, "Stochastic relaxations and EM algorithms for Markov random fields," *Journal of Statistical Computation and Simulation*, vol. 40, pp. 55–69, 1992.

[49] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.

[50] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review*, vol. 26, no. 2, pp. 195–239, April 1984.

[51] W. H. Richardson, "Bayesian-based iterative method of image restoration," *Journal of Optical Society of America*, vol. 62, no. 1, pp. 55–59, 1972.

[52] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York, NY: Springer, 2nd Edition, 2004.

[53] 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.

[54] G. Ronning, "Maximum Likelihood estimation of Dirichlet distributions," *Journal of Statistical Computation and Simulation*, vol. 32, no. 4, pp. 215–221, 1989.

[55] H. Stark and Y. Yang, *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics*. New York, NY: John Wiley & Sons, 1998.

[56] C. A. Sugar and G. M. James, "Finding the number of clusters in a dataset: An information-theoretic approach," *Journal of the American Statistical Association*, vol. 98, no. 463, pp. 750–763, September 2003.

[57] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 528–540, June 1987.

[58] D. A. van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, March 2001.

[59] J. Wang, A. Dogandzic, and A. Nehorai, "Maximum likelihood estimation of compound-Gaussian clutter and target parameters," *IEEE Transactions on Signal Processing*, vol. 54, no. 10, pp. 3884–3898, October 2006.

[60] G. C. G. Wei and M. A. Tanner, "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 699–704, September 1990.

[61] L. R. Welch, "Hidden Markov Models and the Baum-Welch Algorithm," *IEEE Information Theory Society Newsletter*, vol. 53, no. 4, pp. 1–13, December 2003.

[62] C. F. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95–103, March 1983.

[63] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for Gaussian mixtures," *Neural Computation*, vol. 8, no. 1, pp. 129–151, January 1996.

[64] R. W. Yeung, *A First Course in Information Theory*. New York, NY: Springer, 2002.

[65] J. Zhang, "The mean field theory in EM procedures for Markov random fields," *IEEE Transactions on Signal Processing*, vol. 40, no. 10, pp. 2570–2583, October 1992.