Chapter 3

EXPECTATION-MAXIMIZATION THEORY

3.1 Introduction

Learning networks are commonly categorized in terms of supervised and unsupervised networks. In unsupervised learning, the training set consists of input training patterns only. In contrast, in supervised learning networks, the training data consist of many pairs of input/output patterns. Therefore, the learning process can benefit greatly from the teacher's assistance. In fact, the amount of adjustment of the updating coefficients often depends on the difference between the desired teacher value and the actual response. As demonstrated in Chapter 5, many supervised learning models have been found to be promising for biometric authentication; their implementation often hinges on an effective data-clustering scheme, which is perhaps the most critical component in unsupervised learning methods. This chapter addresses a data-clustering algorithm, called the expectation-maximization (EM) algorithm, when complete or partial information of observed data is made available.

3.1.1 K-Means and VQ algorithms

An effective data-clustering algorithm is known as K-means [85], which is very similar to another clustering scheme known as the vector quantization (VQ) algorithm [118]. Both methods classify data patterns based on the *nearest-neighbor* criterion.

Verbally, the problem is to cluster a given data set $X = \{x_t; t = 1, \ldots, T\}$ into K groups, each represented by its centroid denoted by $\mu^{(j)}, j = 1, \ldots, K$. The task is (1) to determine the K centroids $\{\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(K)}\}$ and (2) to assign each pattern x_t to one of the centroids. The nearest-neighbor rule assigns a pattern x to the class associated with its nearest centroid, say $\mu^{(i)}$.

Mathematically speaking, one denotes the centroid associated with x_t as μ_t, where $\mu_t \in \{\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(K)}\}$. Then the objective of the K-means algorithm is

to minimize the following sum of squared errors:

$$E(X) = \sum_t \|x_t - \mu_t\|^2, \tag{3.1.1}$$

where $\|\cdot\|$ is the Euclidean norm.

Let X_k denote the set of data patterns associated with the k-th cluster with the centroid $\mu^{(k)}$ and N_k denotes the number of patterns in the cluster X_k, where $k = 1, \ldots, K$. The learning rule of the K-means algorithm consists of the following two basic steps.

1. *Determine the membership of a data pattern:*

$$\mathbf{x} \in X_k \quad \text{if} \quad \|\mathbf{x} - \mu_k\| < \|\mathbf{x} - \mu_j\| \quad \forall j \neq k. \tag{3.1.2}$$

2. *Updating the representation of the cluster:* In a clustering process, the inclusion (or removal) of a new pattern in a cluster (or from a cluster) affects the representation (e.g., the centroid or variance) of the cluster. Therefore, the centroid should be updated based on the new membership:

$$\mu_j \;=\; \frac{1}{N_j} \sum_{\mathbf{x} \in X_j} \mathbf{x}. \tag{3.1.3}$$

Sometimes, the variance of the data cluster is also of great interest (e.g., in Gaussian mixture models). In this case, the variance can be computed as

$$\Sigma_j \;=\; \frac{1}{N_j} \sum_{\mathbf{x} \in X_j} (\mathbf{x} - \mu_j)(\mathbf{x} - \mu_j)^T. \tag{3.1.4}$$

3.1.2 Gaussian Mixture Model

The EM scheme can be seen as a generalized version of K-means clustering. The main difference hinges on the notion of a hard-versus-soft membership. A hard membership is adopted in the K-means algorithm, (i.e., a data pattern is assigned to one cluster only). This is not the case with the EM algorithm, where a soft membership is adopted, (i.e., the membership of each data pattern can be distributed over multiple clusters).

The necessity of using a distributed (i.e., soft) membership is the most conspicuous for a Gaussian mixture model (GMM). Given a set of N-independent and identically distributed patterns $X^{(i)} = \{\mathbf{x}_t; t = 1, 2, \ldots, N\}$ associated with class ω_i, the likelihood function $p(\mathbf{x}_t|\omega_i)$ for class ω_i is a mixture of Gaussian distributions; that is,

$$p(\mathbf{x}_t|\omega_i) = \sum_{r=1}^{R} P(\Theta_{r|i}|\omega_i) p(\mathbf{x}_t|\omega_i, \Theta_{r|i}), \tag{3.1.5}$$

where $\Theta_{r|i}$ represents the parameters of the r-th mixture component; R is the total number of mixture components; $p(\mathbf{x}_t|\omega_i, \Theta_{r|i}) \equiv \mathcal{N}(\mathbf{x}; \mu_{r|i}, \Sigma_{r|i})$ is the probability density function of the r-th component; and $P(\Theta_{r|i}|\omega_i)$ is the prior probability of the r-th component. Typically, $\mathcal{N}(\mathbf{x}; \mu_{r|i}, \Sigma_{r|i})$ is a Gaussian distribution with mean $\mu_{r|i}$ and covariance $\Sigma_{r|i}$.

In short, the output of a GMM is the weighted sum of R-component densities. The training of GMMs can be formulated as a maximum likelihood problem, where the mean vectors $\{\mu_{r|i}\}$, covariance matrices $\{\Sigma_{r|i}\}$, and mixture coefficients $\{P(\Theta_{r|i}|\omega_i)\}$ are often estimated by the iterative EM algorithm—the main topic of the current chapter.

3.1.3 Expectation-Maximization Algorithm

The expectation-maximization (EM) algorithm is an ideal candidate for solving parameter estimation problems for the GMM or other neural networks. In particular, EM is applicable to problems, where the observable data provide only partial information or where some data are "missing"—see Figure 3.1(a). Another important class of parameter estimation that can be addressed by EM involves a mixture-of-experts—see Figure 3.1(b). In this class of problems, there are two categories of unknown parameters: one pertaining to the membership function of an expert (or cluster) and the other consisting of the unknown parameters defining individual experts. Let's use a Gaussian mixture model shown in Figure 3.1(b) as an example, where $\pi^{(j)}$ denotes the prior probability of expert j and where $\phi^{(j)} = \{\mu^{(j)}, \Sigma^{(j)}\}$ denotes the parameters (mean and variance) of the expert. This chapter explains why the EM method can serve as a powerful tool for estimating these parameters. It also demonstrates how the EM algorithm can be applied to data clustering.

The EM algorithm is a very general parameter estimation method in that it is applicable to many statistical models, for example, mixture-of-experts (MOE), Gaussian mixture models (GMMs), and vector quantization (VQ). Figure 3.2 depicts the relationship among EM, MOE, GMM, and VQ. In particular, the figure highlights the fact that VQ is a special case of GMM, which in turn is a special case of the more general mixture-of-experts. More important, EM is applicable to all of these models.

The classic EM algorithm can be dated back to Dempster, Laird, and Rubin's paper in 1977 [74]. It is a special kind of quasi-Newton algorithm with a searching direction having a positive projection on the gradient of log-likelihood. Each EM iteration consists of two steps—Estimation (E) and Maximization (M). The M-step maximizes a likelihood function that is refined in each iteration by the E-step. Interested readers can refer to the references [74, 168, 297, 350]

One important feature of the EM algorithm is that it can be applied to problems in which observed data provide "partial" information only or when artificially introducing some information (referred to as "hidden"-state information hereafter) can greatly simplify the parameter estimation process. Figure 3.3 illustrates the concept of hidden and partial data. In Figure 3.3(a), all data (x_1 to x_7) are known.

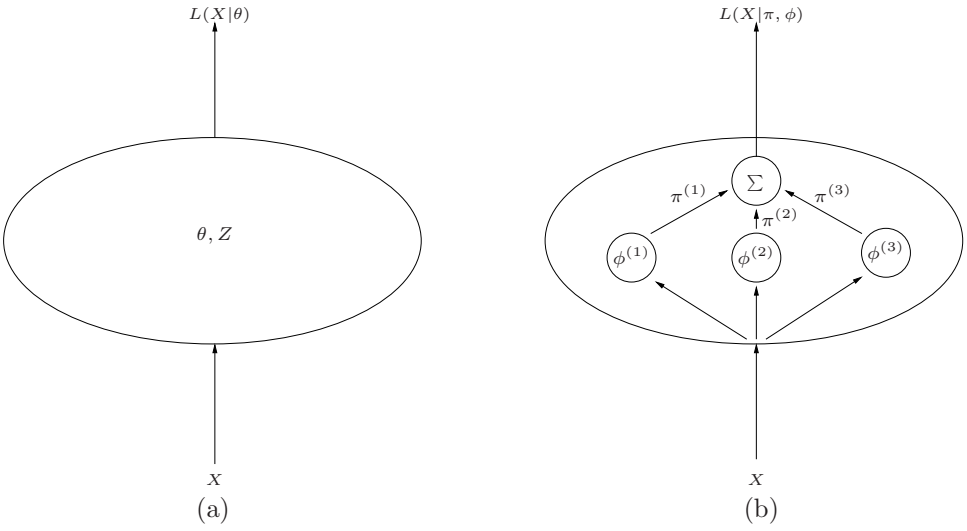


Figure 3.1. Parameter estimation by EM. (a) EM for general missing data problems, where θ is the nonstructural parameters to be estimated and Z is the set of missing data. (b) EM for hidden-state problems in which the parameter θ can be divided into two groups: $\{\pi^{(j)}\}_{j=1}^{3}$ and $\{\phi^{(j)}\}_{j=1}^{3}$, where $\pi^{(j)}$ represents the prior probability of the j-th expert and $\phi^{(j)}$ defines the density function associated with the j-th expert.

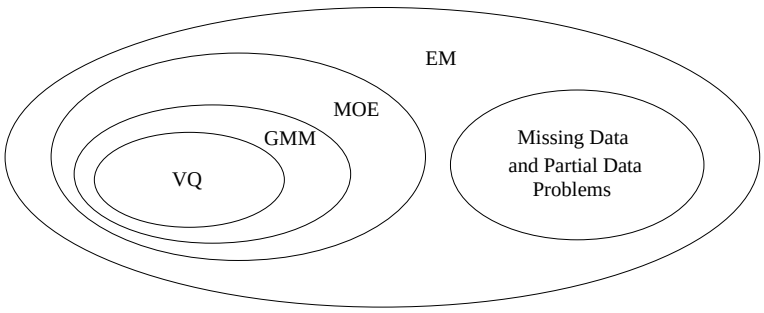


Figure 3.2. Diagram depicting the relationship among EM, MOE, GMM, VQ and the class of problems known as missing- and partial-data problems.

Let's assume that there are two clusters in the observed data. Although all data constituting the two clusters are observable, one does not know exactly to which cluster each of these data belongs. Lacking this hidden membership information results in a complicated parameter estimation procedure. The estimation procedure, however, can be greatly simplified if this membership information is assumed to be

known. For example, if the cluster identities of x_1 to x_7 in Figure 3.3(a) were known, finding the cluster means is reduced to computing the mean of individual clusters separately. Figure 3.3(b) illustrates the idea of partial data. Unlike Figure 3.3(a), the partial-data problem in Figure 3.3(b) contains uncertain data y because y can be equal to 5.0 or 6.0. As a result, the true value of y is unobservable whereas those of x_1 to x_4 are observable. The EM algorithm can solve this partial-data problem effectively by computing the expected value of y. Figure 3.3(c) illustrates the case in which cluster membership information is hidden and only partial information is available. The problem can be viewed as a generalization of the problems in Figure 3.3(a) and Figure 3.3(b). A new type of EM called doubly-stochastic EM is derived in Section 3.4 to address this kind of general problem. Numerical solutions for the problems in Figure 3.3 are provided in later sections.

The concepts of hidden and partial data have been applied to many scientific and engineering applications. For instance, in digital communication, the receiver receives a sequence consisting of $+1$'s and -1's without knowing which bit in the sequence is a $+1$ and which bit is a -1. In such cases, the state of each bit constitutes the missing information. In biometric applications, a MOE is typically applied to model the features of an individual. Each expert is designed to model some of the user-specific features. In such cases, the contribution of individual experts constitutes the hidden information.

EM has been shown to have favorable convergence properties, automatical satisfaction of constraints, and fast convergence. The next section explains the traditional approach to deriving the EM algorithm and proving its convergence property. Section 3.3 covers the interpretion the EM algorithm as the maximization of two quantities: the entropy and the expectation of complete-data likelihood. Then, the K-means algorithm and the EM algorithm are compared. The conditions under which the EM algorithm is reduced to the K-means are also explained. The discussion in Section 3.4 generalizes the EM algorithm described in Sections 3.2 and 3.3 to problems with partial-data and hidden-state. We refer to this new type of EM as the doubly stochastic EM. Finally, the chapter is concluded in Section 3.5.

3.2 Traditional Derivation of EM

Each EM iteration is composed of two steps—Estimation (E) and Maximization (M). The M-step maximizes a likelihood function that is further refined in each iteration by the E-step. This section derives the traditional EM and establishes its convergence property.

3.2.1 General Analysis

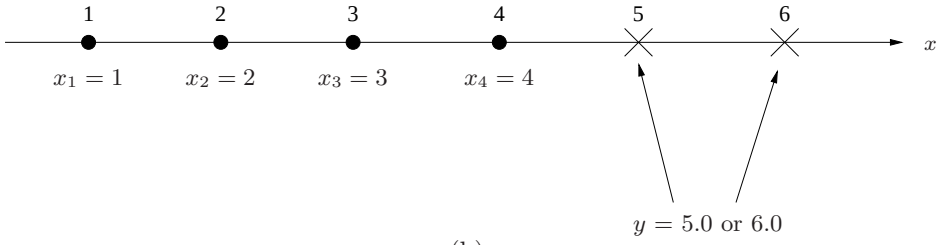
The following notations are adopted.

- $X = \{x_t \in \Re^D; t = 1, \ldots, T\}$ is the observation sequence, where T is the number of observations and D is the dimensionality of x_t.

Example 1: Hidden-State Problem

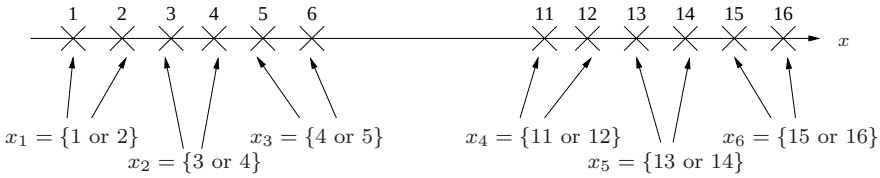
(a)

Example 2: Partial-Data Problem



(b)

Example 3: Doubly-Stochastic (Partial-Data and Hidden-State)



(c)

Figure 3.3. One-dimensional example illustrating the concept of (a) hidden-state, (b) partial-data, and (c) combined partial-data and hidden-state. In (a) the information regarding the cluster membership of x_t is hidden; in (b) y is partial in that its exact value is unknown; and in (c) data x_t provide partial information only because none of their exact values are known. The cluster membership information is also hidden.

- $\mathcal{C} = \{\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(J)}\}$ is the set of cluster mixture labels, where J is the number of mixture components.
- $Z = \{z_t \in \mathcal{C}; t = 1, \ldots, T\}$ is the set of missing data (specifying the hidden-state information).
- $\theta = \{\theta^{(j)}; j = 1, \ldots, J\}$ is the set of unknown parameters that define the density function for approximating the true probability density of X.
- $\theta^{(j)} = \{\pi^{(j)}, \phi^{(j)}\}$, where $\pi^{(j)}$ denotes the prior probability of the j-th component density and $\phi^{(j)}$ defines the j-th component density.

Note that the combination of observations X and the "hidden-states" Z constitute the complete-data. The likelihood of the complete-data is instrumental in accordance with the EM formulation.

To facilitate the derivation, define

$$L(X|\theta_n) \equiv \log p(X|\theta_n) \tag{3.2.1}$$

as the log-likelihood of the incomplete-data given the current estimate θ_n, where n represents the iteration index; also, define $p(Z, X|\theta_n)$ as the completed data likelihood. According to probability theory,[1] $p(X|\theta_n)$ can be expressed as

$$p(X|\theta_n) = \frac{p(Z, X|\theta_n)}{P(Z|X, \theta_n)}. \tag{3.2.2}$$

Using Eq. 3.2.1 and Eq. 3.2.2, one can write the incomplete-data log-likelihood as follows:

$$\begin{aligned}
L(X|\theta_n) &\equiv \log p(X|\theta_n) \\
&= [\log p(X|\theta_n)] \sum_Z P(Z|X, \theta_n) && \text{(since } \textstyle\sum_Z P(Z|X, \theta_n) = 1\text{)} \\
&= \sum_Z P(Z|X, \theta_n) \log p(X|\theta_n) \\
&= \sum_Z P(Z|X, \theta_n) \log \frac{p(Z, X|\theta_n)}{P(Z|X, \theta_n)} && \text{(as a result of Eq. 3.2.2)} \\
&= \sum_Z P(Z|X, \theta_n) \log p(Z, X|\theta_n) - \sum_Z P(Z|X, \theta_n) \log P(Z|X, \theta_n) \\
&= E_Z\{\log p(Z, X|\theta_n)|X, \theta_n\} \\
&\qquad - E_Z\{\log P(Z|X, \theta_n)|X, \theta_n\} && \text{(by definition of expectation)} \\
&= Q(\theta_n|\theta_n) + R(\theta_n|\theta_n), && (3.2.3)
\end{aligned}$$

where $E_Z\{\}$ denotes expectation with respect to Z. Thus, denote

$$Q(\theta|\theta_n) \equiv E_Z\{\log p(Z, X|\theta)|X, \theta_n\} \tag{3.2.4}$$

[1] Hereafter, a capital P denotes probabilities and a lowercase p denotes density functions.

and

$$R(\theta|\theta_n) \equiv -E_Z\{\log P(Z|X,\theta)|X,\theta_n\}, \tag{3.2.5}$$

where $R(\theta|\theta_n)$ is an entropy term representing the difference between the incomplete-data likelihood and the expectation of the completed-data likelihood. Interpretation of $R(\theta|\theta_n)$ and its role in the EM algorithm is discussed further in Section 3.3.

3.2.2 Convergence Property of EM

The following demonstrates why the EM algorithm has a general convergence property. The basic idea is via Jensen's inequality. More precisely, it can be shown that if the Q-function in Eq. 3.2.4 is improved in each iteration (in the M-step), then so will be the likelihood function L.

The proof of convergence begins with the observation of the following relationship:

$$L(X|\theta) = \log p(X|\theta) = \log\left\{\sum_Z p(Z,X|\theta)\right\} = \log\left\{\sum_Z P(Z|X,\theta_n)\frac{p(Z,X|\theta)}{P(Z|X,\theta_n)}\right\}. \tag{3.2.6}$$

Using Eq. 3.2.6 and Jensen's inequality, this is obtained:

$$\begin{aligned}
L(X|\theta) &= \log p(X|\theta) \\
&= \log\left\{\sum_Z P(Z|X,\theta_n)\frac{p(Z,X|\theta)}{P(Z|X,\theta_n)}\right\} \\
&= \log\left\{E_Z\left[\frac{p(Z,X|\theta)}{P(Z|X,\theta_n)}\middle|X,\theta_n\right]\right\} && \text{(by definition of expectation)} \\
&\geq E_Z\left\{\log\left[\frac{p(Z,X|\theta)}{P(Z|X,\theta_n)}\right]\middle|X,\theta_n\right\} && \text{(by Jensen's inequality)} \\
&= \sum_Z P(Z|X,\theta_n)\log\left[\frac{p(Z,X|\theta)}{P(Z|X,\theta_n)}\right] && \text{(by definition of expectation)} \\
&= \sum_Z P(Z|X,\theta_n)\log p(Z,X|\theta) - \sum_Z P(Z|X,\theta_n)\log P(Z|X,\theta_n) \\
&= E_Z\{\log p(Z,X|\theta)|X,\theta_n\} - E_Z\{\log P(Z|X,\theta_n)|X,\theta_n\} \\
&= Q(\theta|\theta_n) + R(\theta_n|\theta_n).
\end{aligned} \tag{3.2.7}$$

In the M-step of the n-th iteration, θ^* is selected according to

$$\theta^* = \arg\max_\theta Q(\theta|\theta_n). \tag{3.2.8}$$

This means one can always choose a θ^* at iteration n such that

$$Q(\theta^*|\theta_n) \geq Q(\theta_n|\theta_n). \tag{3.2.9}$$

Note that this equation constitutes a sufficient condition to ensure the convergence property of the EM algorithm because, according to Eqs. 3.2.3, 3.2.7, and 3.2.9

$$\begin{aligned} L(X|\theta^*) &\geq Q(\theta^*|\theta_n) + R(\theta_n|\theta_n) \\ &\geq Q(\theta_n|\theta_n) + R(\theta_n|\theta_n) \\ &= L(X|\theta_n). \end{aligned}$$

Instead of directly maximizing $L(X|\theta)$, the EM algorithm divides the optimization problem into two subproblems: **E**xpectation and **M**aximization.

In each EM iteration, the E-step computes $Q(\theta|\theta_n)$ using a set of presumed model parameters θ_n. The M-step determines the value of θ (say θ^*) that maximizes $Q(\theta|\theta_n)$; that is,

$$\theta^* = \max_{\theta} \sum_{Z} P(Z|X, \theta_n) \log p(Z, X|\theta). \tag{3.2.10}$$

This results in (see Problem 8)

$$p(Z, X|\theta^*) = \frac{P(Z|X, \theta_n)}{\sum\limits_{Z} P(Z|X, \theta_n)}. \tag{3.2.11}$$

Dividing the optimization into two interdependent steps is most useful if optimizing $Q(\theta|\theta_n)$ is simpler than that of $L(X|\theta_n)$. Figure 3.4 illustrates how the E- and M-steps interplay to obtain a maximum-likelihood solution. The next section explains how to compute $Q(\theta|\theta_n)$ in the E-step and how to maximize $Q(\theta|\theta_n)$ in the M-step.

Generalized EM

In case θ^* in Eq. 3.2.8 is difficult to attain, the EM approach is still applicable if one can *improve* $Q(\theta|\theta_n)$ in each M-step (e.g., by gradient ascent). The algorithm is known as generalized EM. Although convergence of generalized EM is slower than that of the standard EM, it offers a more general and flexible framework for dividing the optimization process into the EM steps.

3.2.3 Complete-Data Likelihood

EM begins with an optimization of a likelihood function, which may be considerably simplified if a set of "missing" or "hidden" data is assumed to be known. The following demonstrates that computing the expectation of the complete-data likelihood in the E-step can be accomplished by finding the expectation of the missing or hidden data.

If $X = \{x_t; t = 1, \ldots, T\}$ contains T statistically independent vectors and $Z = \{z_t \in \mathcal{C}; t = 1, \ldots, T\}$, where $z_t = \mathcal{C}^{(j)}$ means that the j-th mixture generates x_t, then one can write $p(Z, X|\theta)$ as

$$p(Z, X|\theta) = \prod_{t=1}^{T} p(z_t, x_t|\theta).$$

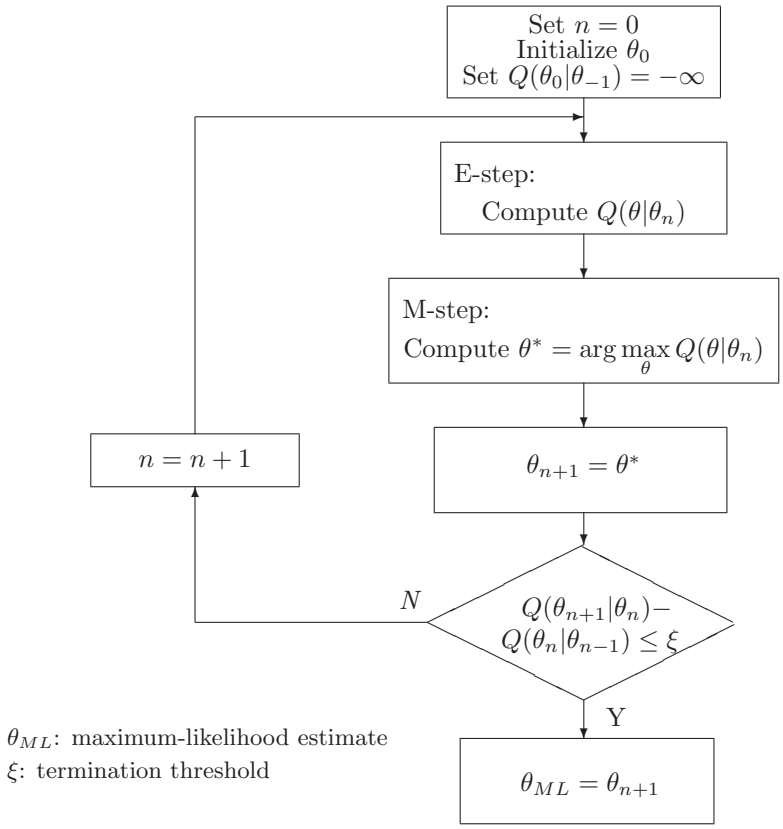


Figure 3.4. The flow of the EM algorithm.

Now, a set of indicator variables is introduced to indicate the status of the hidden-states:[2]

$$\Delta = \{\delta_t^{(j)}; j = 1, \ldots, J \text{ and } t = 1, \ldots, T\}$$

where

$$\delta_t^{(j)} \equiv \delta(z_t, \mathcal{C}^{(j)}) = \begin{cases} 1 & \text{if } x_t \text{ is generated by mixture } \mathcal{C}^{(j)}, \\ 0 & \text{otherwise.} \end{cases}$$

Since for each t only one of the terms in $\{\delta_t^{(j)}; j = 1, \ldots, J\}$ is equal to one and all of the others are equal to 0, one can express $p(Z, X|\theta)$ as follows:

$$p(Z, X|\theta) = \prod_{t=1}^{T} \sum_{j=1}^{J} \delta_t^{(j)} p(x_t, z_t|\theta)$$

[2]For illustration simplicity, assume that the missing data is in discrete form or the hidden data is the cluster membership.

$$= \prod_{t=1}^{T} \sum_{j=1}^{J} \delta_t^{(j)} p(x_t, z_t = \mathcal{C}^{(j)}|\theta)$$
$$= \prod_{t=1}^{T} \sum_{j=1}^{J} \delta_t^{(j)} p(x_t, \delta_t^{(j)} = 1|\theta).$$

Hence, the completed-data likelihood is given by

$$\begin{aligned}
\log p(Z, X|\theta) &= \sum_{t=1}^{T} \log \left\{ \sum_{j=1}^{J} \delta_t^{(j)} p(x_t, \delta_t^{(j)} = 1|\theta) \right\} \\
&= \sum_{t=1}^{T} \log \left\{ \sum_{j=1}^{J} \delta_t^{(j)} p(x_t|\delta_t^{(j)} = 1, \theta) P(\delta_t^{(j)} = 1|\theta) \right\} \\
&= \sum_{t=1}^{T} \log \left\{ \sum_{j=1}^{J} \delta_t^{(j)} p(x_t|\delta_t^{(j)} = 1, \phi^{(j)}) P(\delta_t^{(j)} = 1) \right\} \\
&= \sum_{t=1}^{T} \sum_{j=1}^{J} \delta_t^{(j)} \log \left[p(x_t|\delta_t^{(j)} = 1, \phi^{(j)}) \pi^{(j)} \right] \\
&= \sum_{t=1}^{T} \sum_{j=1}^{J} \delta_t^{(j)} \log \left[p(x_t|z_t = \mathcal{C}^{(j)}, \phi^{(j)}) \pi^{(j)} \right], \qquad (3.2.12)
\end{aligned}$$

where $\pi^{(j)}$ is the mixing coefficient of the j-th mixture. Eq. 3.2.12 uses the fact that $p(x_t|\delta_t^{(j)} = 1, \theta) = p(x_t|\delta_t^{(j)} = 1, \phi^{(j)})$ and $P(\delta_t^{(j)} = 1|\theta) = \pi^{(j)}$. Moreover, because there is only one non-zero term inside the summation $\sum_{j=1}^{J}$, one can extract $\delta_t^{(j)}$ from the log function without affecting the result.

E-Step. Taking the expectations of Eq. 3.2.12 and using the defintions in Eq. 3.2.4, one obtains

$$\begin{aligned}
Q(\theta|\theta_n) &= E_Z\{\log p(Z, X|\theta)|X, \theta_n\} \\
&= \sum_{t=1}^{T} \sum_{j=1}^{J} E\{\delta_t^{(j)}|x_t, \theta_n\} \log \left[p(x_t|\delta_t^{(j)} = 1, \phi^{(j)}) \pi^{(j)} \right]. \qquad (3.2.13)
\end{aligned}$$

Then, define

$$h_n^{(j)}(x_t) \equiv E\{\delta_t^{(j)}|x_t, \theta_n\} = P(\delta_t^{(j)} = 1|x_t, \theta_n)$$

and denote $\pi_n^{(j)}$ as the j-th mixture coefficient at iteration n. Using the Bayes theorem, one can express $h_n^{(j)}(x_t)$ as

$$h_n^{(j)}(x_t) = P(\delta_t^{(j)} = 1|x_t, \theta_n)$$

$$
\begin{aligned}
&= \frac{p(x_t|\delta_t^{(j)}=1,\theta_n)P(\delta_t^{(j)}=1|\theta_n)}{p(x_t|\theta_n)} \\
&= \frac{p(x_t|\delta_t^{(j)}=1,\phi_n^{(j)})P(\delta_t^{(j)}=1|\theta_n)}{p(x_t|\theta_n)} \\
&= \frac{p(x_t|\delta_t^{(j)}=1,\phi_n^{(j)})\pi_n^{(j)}}{\sum_{k=1}^{J} p(x_t|\delta_t^{(k)}=1,\phi_n^{(k)})\pi_n^{(k)}}.
\end{aligned} \tag{3.2.14}
$$

The E-step determines the best guess of the membership function $h_n^{(j)}(x_t)$. Once the probability $h_n^{(j)}(x_t)$ are computed for each t and j, $Q(\theta|\theta_n)$ can be considered as a function of θ. In the M-step of each iteration, this function is maximized to obtain the best value of θ (denoted as θ^*). In most cases, the M-step is substantially simplified if $h_n^{(j)}(x_t)$ are known. Therefore, the E-step can be viewed as a preparation step for the M-step.

3.2.4 EM for GMMs

To better illustrate the EM steps, a simple example applying EM to Gaussian mixture models (GMMs) is presented next. The most common forms for the mixture density are the radial basis functions (RBFs) or the more general elliptical basis functions (EBFs). In the latter case, the component density $p(x_t|\delta_t^{(j)}=1,\phi^{(j)})$ is a Gaussian distribution, with the model parameter of the j-th cluster $\phi^{(j)} = \{\mu^{(j)}, \Sigma^{(j)}\}$ consisting of a mean vector and a full-rank covariance matrix.

Assume a Gaussian mixture model:

$$\theta = \{\pi^{(j)}, \mu^{(j)}, \Sigma^{(j)}; j = 1, \ldots, J\},$$

where $\pi^{(j)}$, $\mu^{(j)}$, and $\Sigma^{(j)}$ denote, respectively, the mixture coefficient, mean vector, covariance matrix of the j-th component density. The GMM's output is given by

$$p(x_t|\theta) = \sum_{j=1}^{J} \pi^{(j)} p(x_t|\delta_t^{(j)}=1,\phi^{(j)}), \tag{3.2.15}$$

where

$$p(x_t|\delta_t^{(j)}=1,\phi^{(j)}) = (2\pi)^{-\frac{D}{2}} |\Sigma^{(j)}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_t-\mu^{(j)})^T(\Sigma^{(j)})^{-1}(x_t-\mu^{(j)})\right\} \tag{3.2.16}$$

is the j-th Gaussian density of the GMM. A closer look at Eqs. 3.2.15 and 3.2.16 reveals that the GMM parameters θ can be divided into two groups: one containing $\pi^{(j)}$s and another containing $\mu^{(j)}$s and $\sum^{(j)}$s. The former indicates the importance of individual mixture densities via the prior probabilities $\pi^{(j)}$s, whereas the latter is commonly regarded as the kernel parameter defining the form of the mixture density. Unlike other optimization techniques (e.g., gradient descent) in which

unknown parameters can be arranged in any order, the EM approach effectively makes use of the structural relationship among the unknown parameters to simplify the optimization process.

After the initialization of θ_0, the EM iteration is as follows:

- *E-step.* In the n-th iteration, compute $h_n^{(j)}(x_t)$ for each j and t using Eqs. 3.2.14 and 3.2.16. This is followed by the M-step described next.
- *M-step.* Maximize $Q(\theta|\theta_n)$ with respect to θ to find θ^*. Replace θ_n by θ^*. Then, increment n by 1 and repeat the E-step until convergence.

To determine $\mu^{(k)*}$, set $\frac{\partial Q(\theta|\theta_n)}{\partial \mu^{(k)}} = 0$, which gives

$$\mu^{(k)*} = \frac{\sum_{t=1}^{T} h_n^{(k)}(x_t) x_t}{\sum_{t=1}^{T} h_n^{(k)}(x_t)}. \tag{3.2.17}$$

To determine $\Sigma^{(k)*}$, set $\frac{\partial Q(\theta|\theta_n)}{\partial \Sigma^{(k)}} = 0$, which gives

$$\Sigma^{(k)*} = \frac{\sum_{t=1}^{T} h_n^{(k)}(x_t)(x_t - \mu^{(k)*})(x_t - \mu^{(k)*})^T}{\sum_{t=1}^{T} h_n^{(k)}(x_t)}. \tag{3.2.18}$$

To determine $\pi^{(k)*}$, maximize $Q(\theta|\theta_n)$ with respect to $\pi^{(k)}$ subject to the constraint $\sum_{j=1}^{J} \pi^{(j)} = 1$, which gives

$$\pi^{(k)*} = \frac{1}{T} \sum_{t=1}^{T} h_n^{(k)}(x_t). \tag{3.2.19}$$

The detailed derivations of Eq. 3.2.17 to Eq. 3.2.19 are as follows:

$$\begin{aligned}
\frac{\partial Q(\theta|\theta_n)}{\partial \mu^{(k)}} &= \sum_{t=1}^{T} \sum_{j=1}^{J} h_n^{(j)}(x_t) \frac{\partial}{\partial \mu^{(k)}} \log \left\{ p(x_t|\delta_t^{(j)} = 1, \phi^{(j)}) \right\} \\
&= \sum_{t=1}^{T} \sum_{j=1}^{J} h_n^{(j)}(x_t) \frac{1}{p(x_t|\delta_t^{(j)} = 1, \phi^{(j)})} \frac{\partial}{\partial \mu^{(k)}} p(x_t|\delta_t^{(j)} = 1, \phi^{(j)}) \\
&= -\frac{1}{2} \sum_{t=1}^{T} \sum_{j=1}^{J} h_n^{(j)}(x_t) \cdot \frac{\partial}{\partial \mu^{(k)}} \Big\{ x_t^T (\Sigma^{(j)})^{-1} x_t - (\mu^{(j)})^T (\Sigma^{(j)})^{-1} x_t \\
&\qquad + (\mu^{(j)})^T (\Sigma^{(j)})^{-1} \mu^{(j)} - x_t^T (\Sigma^{(j)})^{-1} \mu^{(j)} \Big\} \\
&= -\frac{1}{2} \sum_{t=1}^{T} h_n^{(k)}(x_t) \Big\{ (0 - (\Sigma^{(k)})^{-1} x_t + (\Sigma^{(k)})^{-1} \mu^{(k)} + ((\Sigma^{(k)})^{-1})^T \mu^{(k)} \\
&\qquad - ((\Sigma^{(k)})^{-1})^T x_t) \Big\} \\
&= \sum_{t=1}^{T} h_n^{(k)}(x_t) (\Sigma^{(k)})^{-1} (\mu^{(k)} - x_t) = 0
\end{aligned}$$

$$\Longrightarrow \mu^{(k)*} = \frac{\sum_{t=1}^{T} h_n^{(k)}(x_t) x_t}{\sum_{t=1}^{T} h_n^{(k)}(x_t)}. \tag{3.2.20}$$

To determine $\Sigma^{(k)*}$, $k = 1, \ldots, J$, let $\Lambda^{(k)} = (\Sigma^{(k)})^{-1}$ and set $\frac{\partial Q(\theta|\theta_n)}{\partial \Lambda^{(k)}} = 0$, that is,

$$\begin{aligned}
\frac{\partial Q(\theta|\theta_n)}{\partial \Lambda^{(k)}} &= \sum_{t=1}^{T}\sum_{j=1}^{J} h_n^{(j)}(x_t) \frac{\partial}{\partial \Lambda^{(k)}} \log\left\{p(x_t|\delta_t^{(j)} = 1, \phi^{(j)})\right\} \\
&= \sum_{t=1}^{T}\sum_{j=1}^{J} h_n^{(j)}(x_t) \frac{\partial}{\partial \Lambda^{(k)}} \\
&\qquad \left\{-\frac{D}{2}\log(2\pi) + \frac{1}{2}\log|\Lambda^{(j)}| - \frac{1}{2}(x_t - \mu^{(j)})^T \Lambda^{(j)} (x_t - \mu^{(j)})\right\} \\
&= \sum_{t=1}^{T}\sum_{j=1}^{J} h_n^{(j)}(x_t) \\
&\qquad \left\{\frac{1}{2}\frac{\partial}{\partial \Lambda^{(k)}}\log|\Lambda^{(j)}| - \frac{1}{2}\frac{\partial}{\partial \Lambda^{(k)}}(x_t - \mu^{(j)})^T \Lambda^{(j)} (x_t - \mu^{(j)})\right\} \\
&= \frac{1}{2}\sum_{t=1}^{T} h_n^{(k)}(x_t) \left\{\frac{\partial}{\partial \Lambda^{(k)}}\log|\Lambda^{(k)}| - \frac{\partial}{\partial \Lambda^{(k)}}(x_t - \mu^{(k)})^T \Lambda^{(k)} (x_t - \mu^{(k)})\right\} \\
&= \frac{1}{2}\sum_{t=1}^{T} h_n^{(k)}(x_t) \left\{\frac{1}{|\Lambda^{(k)}|}\frac{\partial|\Lambda^{(k)}|}{\partial \Lambda^{(k)}} - (x_t - \mu^{(k)})(x_t - \mu^{(k)})^T\right\} \\
&= \frac{1}{2}\sum_{t=1}^{T} h_n^{(k)}(x_t) \left\{\frac{1}{|\Lambda^{(k)}|}|\Lambda^{(k)}|(\Lambda^{(k)})^{-1} - (x_t - \mu^{(k)})(x_t - \mu^{(k)})^T\right\} \\
&= \frac{1}{2}\sum_{t=1}^{T} h_n^{(k)}(x_t) \left\{(\Lambda^{(k)})^{-1} - (x_t - \mu^{(k)})(x_t - \mu^{(k)})^T\right\} = 0
\end{aligned} \tag{3.2.21}$$

$$\Longrightarrow (\Lambda^{(k)*})^{-1} = \Sigma^{(k)*} = \frac{\sum_{t=1}^{T} h_n^{(k)}(x_t)(x_t - \mu^{(k)})(x_t - \mu^{(k)})^T}{\sum_{t=1}^{T} h_n^{(k)}(x_t)} \tag{3.2.22}$$

Note that Eq. 3.2.21 makes use of the identity $\frac{\partial|A|}{\partial A} = |A|A^{-1}$, where A is a symmetric matrix. Note also that one can replace $\mu^{(k)}$ by $\mu^{(k)*}$ in Eq. 3.2.20 to obtain Eq. 3.2.18.

To determine $\pi^{(r)}, r = 1, \ldots, J$, maximize $Q(\theta|\theta_n)$ with respect to $\pi^{(r)}$ subject to the constraint $\sum_{j=1}^{J} \pi^{(j)} = 1$. More specifically, maximize the function $f(\lambda, \pi^{(j)}) = Q(\theta|\theta_n) + \lambda(\sum_{j=1}^{J} \pi^{(j)} - 1)$ where λ is the Lagrange multiplier. Setting $\frac{\partial f(\lambda, \pi^{(j)})}{\partial \pi^{(r)}} = 0$ results in

$$\frac{\partial Q(\theta|\theta_n)}{\partial \pi^{(r)}} + \lambda = 0 \tag{3.2.23}$$

$$\Longrightarrow \lambda = -\sum_{t=1}^{T} h_n^{(r)}(x_t) \frac{\partial}{\partial \pi^{(r)}} \log \pi^{(r)}$$

$$\Longrightarrow \lambda \pi^{(r)*} = -\sum_{t=1}^{T} h_n^{(r)}(x_t). \tag{3.2.24}$$

Summing both size of Eq. 3.2.24 from $r = 1$ to J, one has

$$\lambda \sum_{r=1}^{J} \pi^{(r)*} = -\sum_{t=1}^{T} \sum_{r=1}^{J} h_n^{(r)}(x_t) \tag{3.2.25}$$

$$\Longrightarrow \lambda = -\sum_{t=1}^{T} \sum_{r=1}^{J} h_n^{(r)}(x_t) = -\sum_{t=1}^{T} 1 = -T. \tag{3.2.26}$$

Substituting Eq. 3.2.26 into Eq. 3.2.24 results in

$$\pi^{(k)*} = \frac{1}{T} \sum_{t=1}^{T} h_n^{(k)}(x_t). \tag{3.2.27}$$

Complexity of EM. Let T denote the number of patterns, J the number of mixtures, and D the feature dimension, then the following is a rough estimation of the computation complexity of using EM to train a GMM:

- *E-step.* $\mathcal{O}(TJD + TJ)$ for each epoch.
- *M-step.* $\mathcal{O}(2TJD)$ for each epoch.

Numerical Example 1. This example uses the data in Figure 3.3(a) as the observed data. Assume that when EM begins, $n = 0$ and

$$\begin{aligned} \theta_0 &= \left\{ \pi_0^{(1)}, \{\mu_0^{(1)}, \sigma_0^{(1)}\}, \pi_0^{(2)}, \{\mu_0^{(2)}, \sigma_0^{(2)}\} \right\} \\ &= \{0.5, \{0, 1\}, 0.5, \{9, 1\}\}. \end{aligned}$$

Therefore, one has

$$\begin{aligned} h_0^{(1)}(x_t) &= \frac{\frac{\pi_0^{(1)}}{\sigma_0^{(1)}} e^{-\frac{1}{2}(x_t - \mu^{(1)})^2/(\sigma_0^{(1)})^2}}{\sum_{k=1}^{2} \frac{\pi_0^{(k)}}{\sigma_0^{(k)}} e^{-\frac{1}{2}(x_t - \mu^{(k)})^2/(\sigma_0^{(k)})^2}} \\ &= \frac{e^{-\frac{1}{2}{x_t}^2}}{e^{-\frac{1}{2}{x_t}^2} + e^{-\frac{1}{2}(x_t - 9)^2}} \end{aligned} \tag{3.2.28}$$

and

$$h_0^{(2)}(x_t) = \frac{e^{-\frac{1}{2}(x_t - 9)^2}}{e^{-\frac{1}{2}{x_t}^2} + e^{-\frac{1}{2}(x_t - 9)^2}}. \tag{3.2.29}$$

Table 3.1. Values of $h_0^{(j)}(x_t)$ in Example 1

Pattern Index (t)	Pattern (x_t)	$h_0^{(1)}(x_t)$	$h_0^{(2)}(x_t)$
1	1	1	0
2	2	1	0
3	3	1	0
4	4	1	0
5	6	0	1
6	7	0	1
7	8	0	1

Table 3.2. Values of $Q(\theta|\theta_n)$, $\mu^{(j)}$ and $(\sigma^{(j)})^2$ in the course of EM iterations. Data shown in Figure 3.3(a) were used as the observed data.

Iteration (n)	$Q(\theta\|\theta_n)$	$\mu_n^{(1)}$	$(\sigma_n^{(1)})^2$	$\mu_n^{(2)}$	$(\sigma_n^{(2)})^2$
0	-∞	0	1	9	1
1	-43.71	2.50	1.25	6.99	0.70
2	-25.11	2.51	1.29	7.00	0.68
3	-25.11	2.51	1.30	7.00	0.67
4	-25.10	2.52	1.30	7.00	0.67
5	-25.10	2.52	1.30	7.00	0.67

Substituting $X = \{1, 2, 3, 4, 6, 7, 8\}$ into Eqs. 3.2.28 and 3.2.29, Table 3.1 is obtained. Substituting $h_0^{(j)}(x_t)$ in Table 3.1 into Eqs. 3.2.17 through 3.2.19 results in

$$\theta_1 = \{0.57, \{2.50, 1.12\}, 0.43, \{6.99, 0.83\}\}.$$

Then, continue the algorithm by computing $Q(\theta|\theta_1)$—that is, $h_1^{(j)}(x_t)$—which are then substituted into Eqs. 3.2.17 through 3.2.19 to obtain θ_2. Figure 3.5 depicts the movement of the component density functions specified by $\mu^{(j)}$ and $\sigma^{(j)}$ during the EM iterations, and Table 3.2 lists the numerical values of $Q(\theta|\theta_n)$ and θ_n for the first five iterations. It is obvious that the algorithm converges quickly in this example.

3.3 An Entropy Interpretation

The previous section has shown that the EM algorithm is a powerful tool in estimating the parameters of finite-mixture models. This is achieved by iteratively maximizing the expectation of the model's completed-data likelihood function. The model's parameters, however, can also be obtained by maximizing an incomplete-data likelihood function, leading to an entropy interpretation of the EM algorithm.

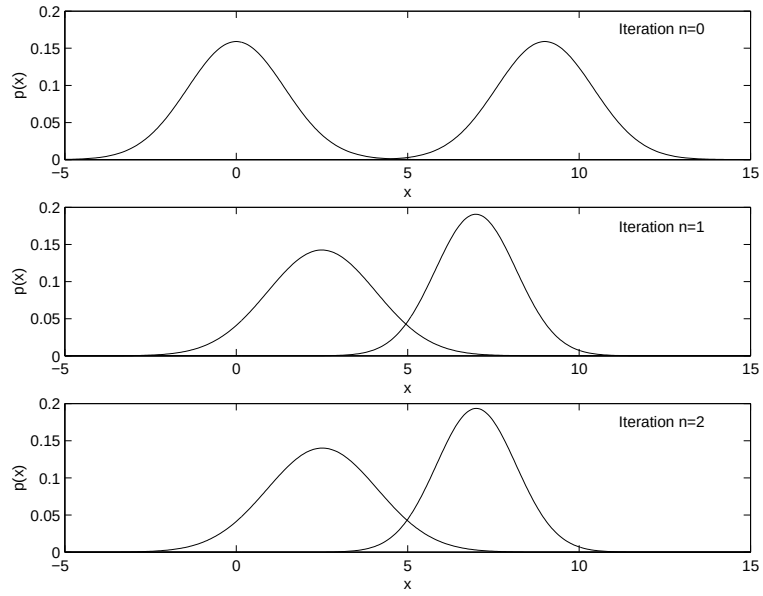


Figure 3.5. Movement of the component density function specified by $\mu^{(j)}$ and $(\sigma^{(j)})^2$ for the first two EM iterations.

3.3.1 Incomplete-Data Likelihood

The optimal estimates are obtained by maximizing

$$\begin{aligned} L(X|\theta) &= \sum_{t=1}^{T} \log p(x_t|\theta) \\ &= \sum_{t=1}^{T} \frac{\sum_{j=1}^{J} \pi^{(j)} p(x_t|\delta_t^{(j)} = 1, \phi^{(j)})}{p(x_t|\theta)} \log p(x_t|\theta). \end{aligned} \tag{3.3.1}$$

Define

$$h^{(j)}(x_t) \equiv \frac{\pi^{(j)} p(x_t|\delta_t^{(j)} = 1, \phi^{(j)})}{p(x_t|\theta)}$$

such that $\sum_j \pi^{(j)} = 1$ and $p(x_t|\theta) = \sum_j \pi^{(j)} p(x_t|\delta_t^{(j)} = 1, \phi^{(j)})$.[3] Eq. 3.3.1 becomes

$$L(X|\theta) = \sum_{t=1}^{T} \sum_{j=1}^{J} h^{(j)}(x_t) \log p(x_t|\theta)$$

[3]Note that $h^{(j)}(x_t)$ equals the probability of x_t belonging to the j-th cluster, given x_t and the model—that is, $h^{(j)}(x_t) = Pr(x_t \in j\text{-th cluster}|x_t, \theta)$; it can be considered a "fuzzy" membership function.

$$
\begin{aligned}
&= \sum_{t=1}^{T}\sum_{j=1}^{J} h^{(j)}(x_t)\left\{\log p(x_t|\theta) - \log\left[\pi^{(j)}p(x_t|\delta_t^{(j)}=1,\phi^{(j)})\right]\right. \\
&\qquad\qquad \left. + \log\left[\pi^{(j)}p(x_t|\delta_t^{(j)}=1,\phi^{(j)})\right]\right\} \\
&= \sum_{t=1}^{T}\sum_{j=1}^{J} h^{(j)}(x_t)\left\{-\log h^{(j)}(x_t) + \log\left[\pi^{(j)}p(x_t|\delta_t^{(j)}=1,\phi^{(j)})\right]\right\} \\
&= \sum_{t,j} h^{(j)}(x_t)\log\pi^{(j)} + \sum_{t,j} h^{(j)}(x_t)\log p(x_t|\delta_t^{(j)}=1,\phi^{(j)}) \\
&\qquad\qquad - \sum_{t,j} h^{(j)}(x_t)\log h^{(j)}(x_t) \\
&\equiv Q + R,
\end{aligned}
$$

where the first two terms correspond to the Q-term in Eq. 3.2.4 and the second terms corresponds to the R-term in Eq. 3.2.5. This means the maximization of L can be accomplished by maximizing the completed-data likelihood Q, as well as maximizing an entropy term R.

Now, define $s(x_t,\phi^{(j)}) \equiv \log p(x_t|\delta_t^{(j)}=1,\phi^{(j)})$ so that the likelihood $L(X|\theta)$ can be expressed as:

$$
L(X|\theta) = -\sum_{t,j} h^{(j)}(x_t)\log h^{(j)}(x_t) + \sum_{t,j} h^{(j)}(x_t)\log\pi^{(j)} + \sum_{t,j} h^{(j)}(x_t)s(x_t,\phi^{(j)}). \tag{3.3.2}
$$

In Eq. 3.3.2, the following three terms have different interpretations:

- The first term can be interpreted as the *entropy* term, which helps induce the membership's fuzziness.
- The second term represents the *prior information*. For each sample x_t, this term grasps the influence (prior probability) of its neighboring clusters; the larger the prior probability, the larger the influence.
- The third term is the *observable-data term*, where $s(x_t,\phi^{(j)})$ represents the influence of the observable data x_t on the total likelihood L.

3.3.2 Simulated Annealing

To control the inference of the entropy terms and the prior information on the total likelihood, one can introduce a temperature parameter σ_T similar to simulated annealing; that is,

$$
\begin{aligned}
L(X|\theta) \;=\; & -\sigma_T\sum_{t,j} h^{(j)}(x_t)\log h^{(j)}(x_t)\sigma_T\sum_{t,j} h^{(j)}(x_t)\log\pi^{(j)} \\
& + \sum_{t,j} h^{(j)}(x_t)s^{(j)}(x_t,\phi^{(j)}).
\end{aligned} \tag{3.3.3}
$$

Maximization of

$$L(X|\theta) = \sum_t L_t \;=\; \sum_t \Bigg[-\sigma_T \sum_j h^{(j)}(x_t) \log h^{(j)}(x_t) + \sigma_T \sum_j h^{(j)}(x_t) \log \pi^{(j)} + \sum_j h^{(j)}(x_t) s(x_t, \phi^{(j)}) \Bigg] \tag{3.3.4}$$

can be reformulated as the maximization of L_t under the constraint that

$$\sum_j h^{(j)}(x_t) - 1 = 0.$$

This is achieved by introducing a Lagrange multiplier λ such that

$$\begin{aligned} \mathcal{L} \;&=\; L_t + \lambda \left(\sum_j h^{(j)}(x_t) - 1 \right) \\ &=\; -\sigma_T \sum_j h^{(j)}(x_t) \log h^{(j)}(x_t) + \sigma_T \sum_j h^{(j)}(x_t) \log \pi^{(j)} + \sum_j h^{(j)}(x_t) s(x_t, \phi^{(j)}) \\ &\quad + \lambda \left(\sum_j h^{(j)}(x_t) - 1 \right) \end{aligned} \tag{3.3.5}$$

is to be maximized. To solve this constrained optimization problem, one needs to apply two different kinds of derivatives, as shown here:

1. $\frac{\partial \mathcal{L}}{\partial h^{(j)}(x_t)} = 0$, which means that

$$-\sigma_T \log h^{(j)}(x_t) - \frac{\sigma_T h^{(j)}(x_t)}{h^{(j)}(x_t)} + \sigma_T \log \pi^{(j)} + s(x_t, \phi^{(j)}) + \lambda = 0$$

that is,

$$h^{(j)}(x_t) = \alpha \pi^{(j)} e^{s(x_t, \phi^{(j)})/\sigma_T}, \tag{3.3.6}$$

where $\alpha = e^{\frac{\lambda}{\sigma_T} - 1}$.

2. $\frac{\partial \mathcal{L}}{\partial \lambda} = 0$, which means that

$$\sum_j h^{(j)}(x_t) - 1 = 0. \tag{3.3.7}$$

Plugging Eq. 3.3.6 into Eq. 3.3.7 results in

$$\alpha = \left(\sum_j \pi^{(j)} e^{s(x_t, \phi^{(j)})/\sigma_T} \right)^{-1}. \tag{3.3.8}$$

Hence, the optimal membership (Eq. 3.3.6) for each data is

$$h^{(j)}(x_t) = \frac{\pi^{(j)} e^{s(x_t,\phi^{(j)})/\sigma_T}}{\sum_k \pi^{(k)} e^{s(x_t,\phi^{(k)})/\sigma_T}}. \tag{3.3.9}$$

It is interesting to note that both Eqs. 3.3.9 and 3.2.14 have the same "marginalized" form. They can be connected by observing that $p(x_t|\delta_t^{(j)} = 1, \phi^{(j)}) \propto e^{s(x_t,\phi^{(j)})/\sigma_T}$ in the case of mixture-of-experts. As an additional bonus, such a connection leads to a claim that the expectation of hidden-states (Eq. 3.2.14) provides an optimal membership estimation.

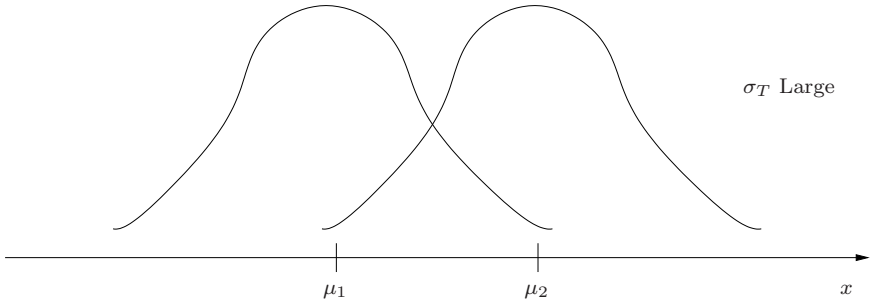
The role of σ_T can be illustrated by Figure 3.6. For simplicity, only two clusters are considered and both $\pi^{(1)}$ and $\pi^{(2)}$ are initialized to 0.5 before the EM iterations begin. Refer to Figure 3.6(a), where the temperature σ_T is extremely high, there exists a major ambiguity between clusters 1 and 2 (i.e., they have almost equivalent probability). This is because according to Eq. 3.3.9, $h^{(j)}(x_t) \simeq 0.5$ at the first few EM iterations when $\sigma_T \to \infty$. When σ_T decreases during the course of EM iterations, such ambiguity becomes more resolved—cf. Figure 3.6(b). Finally, when σ_T approaches zero, a total "certainty" is reached: the probability that either cluster 1 or 2 will approach 100%—cf. Figure 3.6(c). This can be explained by rewriting Eq. 3.3.9 in the following form (for the case $J = 2$ and $j = 2$):

$$\begin{aligned} h^{(2)}(x_t) &= \frac{\pi^{(2)} e^{s(x_t,\phi^{(2)})/\sigma_T}}{\pi^{(1)} e^{s(x_t,\phi^{(1)}/\sigma_T)} + \pi^{(2)} e^{s(x_t,\phi^{(2)})/\sigma_T)}} \\ &= \frac{\frac{\pi^{(2)}}{\pi^{(1)}} e^{s(x_t,\phi^{(2)})/\sigma_T - s(x_t,\phi^{(1)})/\sigma_T}}{1 + \frac{\pi^{(2)}}{\pi^{(1)}} e^{s(x_t,\phi^{(2)})/\sigma_T - s(x_t,\phi^{(1)})/\sigma_T}}. \end{aligned} \tag{3.3.10}$$

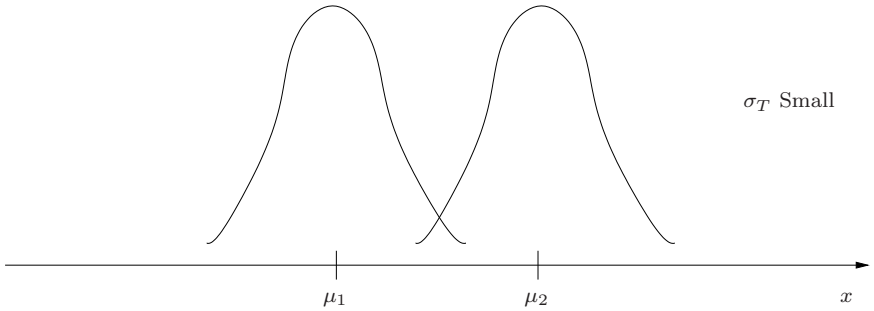
In Eq. 3.3.10, when $\sigma_T \to 0$ and $s(x_t, \phi^{(2)}) > s(x_t, \phi^{(1)})$, $h^{(2)}(x_t) \simeq 1.0$, and $h^{(1)}(x_t) \simeq 0.0$. This means that x_t is closer to cluster 2 than to cluster 1. Similarly, $h^{(2)}(x_t) \simeq 0.0$ and $h^{(1)}(x_t) \simeq 1.0$ when $s(x_t, \phi^{(2)}) < s(x_t, \phi^{(1)})$. Therefore, Eq. 3.3.10 suggests that when $\sigma_T \to 0$, there is a hard-decision clustering (i.e., with cluster probabilities equal to either 1 or 0). This demonstrates that σ_T plays the same role as the temperature parameter in the simulated annealing method. It is a common practice to use annealing temperature schedules to force a more certain classification (i.e., starting with a higher σ_T and then gradually decreasing σ_T to a lower value as iterations progress).

3.3.3 EM Iterations

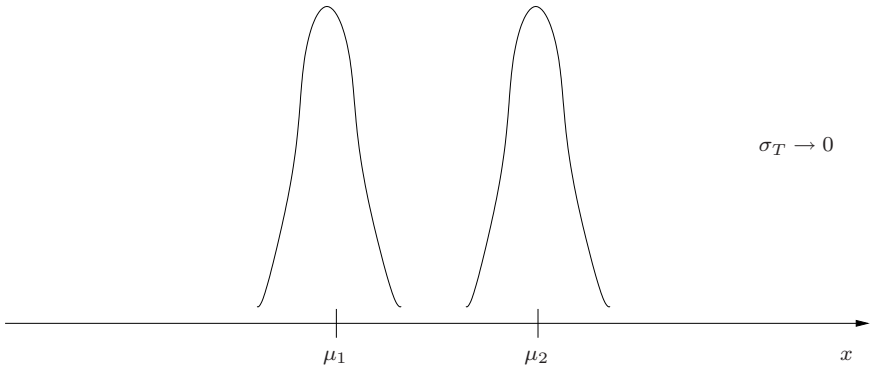
Next, the optimization formulation described in Section 3.2 is slightly modified (but causes no net effect). The EM problem can be expressed as one that maximizes L with respect to both (1) the model parameters $\theta = \{\theta^{(j)}, \forall j\}$ and (2) the membership function $\{h^{(j)}(x_t), \forall t \text{ and } j\}$. The interplay of these two sets of variables can hopefully induce a bootstrapping effect facilitating the convergence process. The list that follows further elaborates on this.



At the beginning of the EM algorithm, σ_T is set to a large value.



After several iterations, σ_T is decreased.



After many iterations, σ_T is set very close to zero.

Figure 3.6. This figure demonstrates how the temperature σ_T can be applied to control the convergence of the EM algorithm.

- In the E-step, while fixing the model parameter $\theta = \{\theta^{(j)}, \forall j\}$, one can find the best cluster probability $h^{(j)}(x_t)$ to optimize L with the constraint $\sum_{j=1}^{J} h^{(j)}(x_t) = 1$, which gave Eq. 3.3.9.

- In the M-step, one searches for the best model parameter $\theta = \{\theta^{(j)}, \forall j\}$ that optimizes L, while fixing the cluster probability $h^{(j)}(x_t), \forall t$ and j.

3.3.4 Special Case: GMM

When θ defines a GMM, $s(x_t, \phi^{(j)})$ becomes

$$s(x_t, \phi^{(j)}) = -\frac{D}{2}\log 2\pi - \frac{1}{2}\log|\Sigma^{(j)}| - \frac{1}{2}(x_t - \mu^{(j)})^T(\Sigma^{(j)})^{-1}(x_t - \mu^{(j)}). \quad (3.3.11)$$

Ignoring terms independent of $h^{(j)}(x_t)$, $\mu^{(j)}$, $\Sigma^{(j)}$, and $\pi^{(j)}$, the likelihood function in Eq. 3.3.2 can be rewritten as:

$$\begin{aligned} L(X|\theta) \;=\; & -\sum_{t,j} h^{(j)}(x_t)\log h^{(j)}(x_t) + \sum_{t,j} h^{(j)}(x_t)\log \pi^{(j)} - \\ & \sum_{t,j} h^{(j)}(x_t)\left\{\frac{1}{2}\log|\Sigma^{(j)}| + \frac{1}{2}(x_t - \mu^{(j)})^T(\Sigma^{(j)})^{-1}(x_t - \mu^{(j)})\right\}. \end{aligned} \quad (3.3.12)$$

Note that the maximization of Eq. 3.3.12 with respect to θ leads to the same maximum likelihood estimtates as shown in Section 3.2.4.

For RBF- and EBF-type likelihood functions, the parameters that maximize $s(x_t, \phi^{(j)})$ can be obtained analytically (see Section 3.2.4), which simplifies the optimization process. On the other hand, if a linear model (e.g. LBF) is chosen to parameterize the likelihood, an iterative method is needed to achieve the optimal solutions in the M-step. In other words, the EM algorithm becomes a double-loop optimization known as the generalized EM. For example, Jordan and Jacobs [168] applied a Fisher scoring method called *iteratively reweighted least squares* (IRLS) to train the LBF mixture-of-experts network.

3.3.5 K-Means versus EM

K-means [85] and VQ [118] are often used interchangeably: They classify input patterns based on the *nearest-neighbor rule*. The task is to cluster a given data set $X = \{x_t; t = 1, \ldots, T\}$ into K groups, each represented by its centroid denoted by $\mu^{(j)}, j = 1, \ldots, K$. The nearest-neighbor rule assigns a pattern x to the class associated with its nearest centroid, say $\mu^{(i)}$. K-means and VQ have simple learning rules and the classification scheme is straightforward. In Eq. 3.3.12, when $h^{(j)}(x_t)$ implements a hard-decision scheme—that is, $h^{(j)}(x_t) = 1$ for the members only, otherwise $h^{(j)}(x_t) = 0$—and $\Sigma^{(j)} = c^2 I \ \forall j$, where c is a constant and I is an

Table 3.3. Learning algorithms as a result of optimizing Eq. 3.3.12 using different kernel types and decision types. RBF and EBF stand for radial basis functions and elliptical basis functions, respectively. Note that EM types of learning occur whenever the decisions in $h^{(j)}(x_t)$ are soft.

Kernel Type	$\Sigma^{(j)}$	$h^{(j)}(x_t)$	**Learning Algorithm**
RBF	Diagonal	Hard	K-means with Euclidean distance
		Soft	EM with Euclidean distance
EBF	Nondiagonal, symmetric	Hard	K-means with Mahalanobis distance
		Soft	EM with Mahalanobis distance

identity matrix, the maximization of Eq. 3.3.12 reduces to the minimization of

$$E(h, X) = \sum_{t} \sum_{j=1}^{K} h^{(j)}(x_t) \|x_t - \mu^{(j)}\|^2. \tag{3.3.13}$$

Therefore, the K-means algorithm aims to minimize the sum of squared error with K clusters.

The EM scheme can be seen as a generalized version of K-means clustering. In other words, K-means clustering is a special case of the EM scheme (cf. Figure 3.2). Table 3.3 summarizes the kinds of learning algorithms that the EM formulation Eq. 3.3.12 can produce.

3.4 Doubly-Stochastic EM

This section presents an EM-based algorithm for problems that possesses partial data with multiple clusters. The algorithm is referred to as as a doubly-stochastic EM. To facilitate the derivation, adopt the following notations:

- $X = \{x_t \in \Re^D; t = 1, \ldots, T\}$ is a sequence of partial-data.
- $Z = \{z_t \in \mathcal{C}; t = 1, \ldots, T\}$ is the set of hidden-states.
- $\mathcal{C} = \{\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(J)}\}$, where J is the number of hidden-states.
- $\Gamma = \{\gamma^{(1)}, \ldots, \gamma^{(K)}\}$ is the set of values that x_t can attain, where K is the number of possible values for x_t.

Also define two sets of indicator variables as:

$$\beta_t^{(k)} = \begin{cases} 1 \text{ if } x_t = \gamma^{(k)} \\ 0 \text{ otherwise} \end{cases}$$

and

$$\delta_t^{(j)} = \begin{cases} 1 \text{ if } z_t = \mathcal{C}^{(j)} \\ 0 \text{ otherwise.} \end{cases}$$

Using these notations and those defined in Section 3.2, $Q(\theta|\theta_n)$ can be written as

$$\begin{aligned}
Q(\theta|\theta_n) &= E\left\{\log p(X, Z|\theta, \Gamma)|X, \Gamma, \theta_n\right\} \\
&= E\left\{\log \prod_{t=1}^{T} p(x_t, z_t|\theta, x_t \in \Gamma)\middle| X, \Gamma, \theta_n\right\} \\
&= E\left\{\sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} \beta_t^{(k)}\delta_t^{(j)} \log p(x_t, z_t|\theta, x_t \in \Gamma)\middle| X, \Gamma, \theta_n\right\} \\
&= \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} E\left\{\beta_t^{(k)}\delta_t^{(j)}\middle| x_t \in \Gamma, \theta_n\right\} \log p(x_t = \gamma^{(k)}, z_t = \mathcal{C}^{(j)}|x_t \in \Gamma, \theta) \\
&= \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} P(x_t = \gamma^{(k)}, z_t = \mathcal{C}^{(j)}|x_t \in \Gamma, \theta_n) \cdot \\
&\qquad \log p(x_t = \gamma^{(k)}, z_t = \mathcal{C}^{(j)}|x_t \in \Gamma, \theta) \\
&= \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} P(x_t = \gamma^{(k)}|z_t = \mathcal{C}^{(j)}, x_t \in \Gamma, \theta_n) P(z_t = \mathcal{C}^{(j)}|x_t \in \Gamma, \theta_n) \cdot \\
&\qquad \log\left[p(x_t = \gamma^{(k)}|x_t \in \Gamma, z_t = \mathcal{C}^{(j)}, \theta) P(z_t = \mathcal{C}^{(j)}|x_t \in \Gamma, \theta)\right] \\
&= \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} g_n^{(k,j)}(x_t) h_n^{(j)}(x_t) \cdot \\
&\qquad \log\left[p(x_t = \gamma^{(k)}|x_t \in \Gamma, z_t = \mathcal{C}^{(j)}, \phi^{(j)})\pi^{(j)}\right], \qquad (3.4.1)
\end{aligned}$$

where

$$g_n^{(k,j)}(x_t) = P(x_t = \gamma^{(k)}|z_t = \mathcal{C}^{(j)}, x_t \in \Gamma, \theta_n) \text{ and}$$
$$h_n^{(j)}(x_t) = P(z_t = \mathcal{C}^{(j)}|x_t \in \Gamma, \theta_n).$$

If θ defines a GMM—that is, $\theta = \{\pi^{(j)}, \mu^{(j)}, \sum^{(j)}\}_{j=1}^{J}$—then

$$\begin{aligned}
Q(\theta|\theta_n) &= \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{k=1}^{K} (2\pi)^{-\frac{D}{2}} |\Sigma_n^{(j)}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\gamma^{(k)} - \mu_n^{(j)})^T (\Sigma_n^{(j)})^{-1}(\gamma^{(k)} - \mu_n^{(j)})\right\} \cdot \\
&\qquad h_n^{(j)}(x_t)\left\{-\frac{D}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma^{(j)}|\right. \\
&\qquad\qquad \left. -\frac{1}{2}(\gamma^{(k)} - \mu^{(j)})^T (\Sigma^{(j)})^{-1}(\gamma^{(k)} - \mu^{(j)}) + \log\pi^{(j)}\right\}.
\end{aligned}$$

3.4.1 Singly-Stochastic Single-Cluster with Partial Data

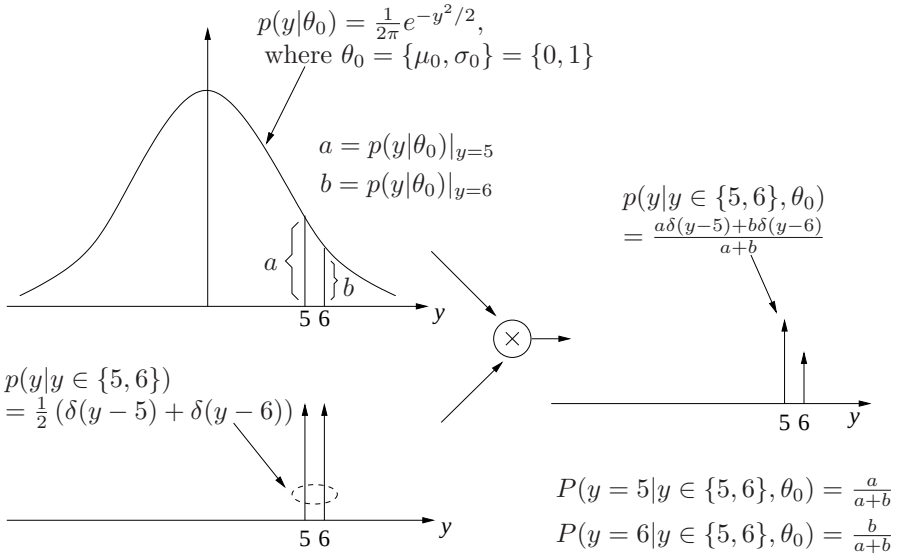
This section demonstrates how the general formulation in Eq. 3.4.1 can be applied to problems with a single cluster and partially observable data. Referring to Example 2 shown in Figure 3.3(b), let $X = \{x_1, x_2, x_3, x_4, y\} = \{1, 2, 3, 4, \{5 \text{ or } 6\}\}$ be the observed data, where $y = \{5 \text{ or } 6\}$ is the observation with missing information. The information is missing because the exact value of y is unknown. Also let $z \in \Gamma$, where $\Gamma = \{\gamma^{(1)}, \gamma^{(2)}\} = \{5, 6\}$, be the missing information. Since there is one cluster only and x_1 to x_4 are certain, define $\theta \equiv \{\mu, \sigma^2\}$, set $\pi^{(1)} = 1.0$ and write Eq. 3.4.1 as

$$\begin{aligned} Q(\theta|\theta_n) &= \sum_{t=1}^{4}\sum_{j=1}^{1}\sum_{k=1}^{1} g_n^{(k,j)}(x_t) h_n^{(j)}(x_t) \log p(x_t|\theta) \\ &\qquad + \sum_{j=1}^{1}\sum_{k=1}^{2} g_n^{(k,j)}(y) h_n^{(j)}(y) \log p(y = \gamma^{(k)}|y \in \Gamma, \theta) \\ &= \sum_{t=1}^{4} \log p(x_t|\theta) + \sum_{k=1}^{2} P(y = \gamma^{(k)}|y \in \Gamma, \theta_n) \log p(y = \gamma^{(k)}|y \in \Gamma, \theta). \end{aligned} \tag{3.4.2}$$

Note that the discrete density $p(y = \gamma^{(k)}|y \in \Gamma, \theta)$ can be interpreted as the product of density $p(y = \gamma^{(k)}|y \in \Gamma)$ and the functional value of $p(y|\theta)$ at $y = \gamma^{(k)}$, as shown in Figure 3.7.

Assume that at the start of the iterations, $n = 0$ and $\theta_0 = \{\mu_0, \sigma_0^2\} = \{0, 1\}$. Then, Eq. 3.4.2 becomes

$$\begin{aligned} Q(\theta|\theta_0) &= \sum_{t=1}^{4} \log p(x_t|\theta) + P(y = 5|y \in \Gamma, \theta_0) \log p(y = 5|y \in \Gamma, \theta) \\ &\qquad + P(y = 6|y \in \Gamma, \theta_0) \log p(y = 6|y \in \Gamma, \theta) \\ &= \text{Const.} - 4\log\sigma - \sum_{t=1}^{4} \frac{(t-\mu)^2}{2\sigma^2} - \\ &\qquad \frac{\frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(5-\mu_0)^2}{2{\sigma_0}^2}}}{\frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(5-\mu_0)^2}{2{\sigma_0}^2}} + \frac{1}{\sqrt{2\pi}\sigma_0} e^{\frac{(6-\mu_0)^2}{2{\sigma_0}^2}}} \left[\frac{(5-\mu)^2}{2\sigma^2} + \log\sigma\right] - \\ &\qquad \frac{\frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(6-\mu_0)^2}{2{\sigma_0}^2}}}{\frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(5-\mu_0)^2}{2{\sigma_0}^2}} + \frac{1}{\sqrt{2\pi}\sigma_0} e^{\frac{(6-\mu_0)^2}{2{\sigma_0}^2}}} \left[\frac{(6-\mu)^2}{2\sigma^2} + \log\sigma\right] - \\ &= \text{Const.} - 4\log\sigma - \sum_{t=1}^{4} \frac{(t-\mu)^2}{2\sigma^2} - \left(\frac{e^{-25/2}}{e^{-25/2} + e^{-36/2}}\right) \left[\frac{(5-\mu)^2}{2\sigma^2} + \log\sigma\right] \end{aligned}$$



Note: Given $y \in \{5, 6\}$, y has non-zero probability only when $y = 5$ or $y = 6$.

Figure 3.7. The relationship between $p(y|\theta_0)$, $p(y|y \in \Gamma)$, $p(y|y \in \Gamma, \theta_0)$, and $P(y = \gamma^{(k)}|y \in \Gamma, \theta_0)$, where $\Gamma = \{5, 6\}$.

$$- \left(\frac{e^{-36/2}}{e^{-25/2} + e^{-36/2}} \right) \left[\frac{(6-\mu)^2}{2\sigma^2} + \log \sigma \right]. \tag{3.4.3}$$

In the M-step, compute θ_1 according to

$$\theta_1 = \arg\max_{\theta} Q(\theta|\theta_0).$$

The next iteration replaces θ_0 in Eq. 3.4.3 with θ_1 to compute $Q(\theta|\theta_1)$. The procedure continues until convergence. Table 3.4 shows the value of μ and σ^2 in the course of EM iterations when their initial values are $\mu_0 = 0$ and $\sigma_0^2 = 1$. Figure 3.8 depicts the movement of the Gaussian density function specified by μ and σ^2 during the EM iterations.

3.4.2 Doubly-Stochastic (Partial-Data and Hidden-State) Problem

Here, the single-dimension example shown in Figure 3.9 is used to illustrate the application of Eq. 3.4.1 to problems with partial-data and hidden-states. Review the following definitions:

- $X = \{x_1, x_2, \ldots, x_6, y_1, y_2\}$ is the available data with certain $\{x_1, \ldots, x_6\}$ and uncertain $\{y_1, y_2\}$ observations.

Table 3.4. Values of μ and σ^2 in the course of EM iterations. Data shown in Figure 3.3(b) were used for the EM iterations.

Iteration (n)	$Q(\theta\|\theta_n)$	μ	σ^2
0	$-\infty$	0.00	1.00
1	-29.12	3.00	7.02
2	-4.57	3.08	8.62
3	-4.64	3.09	8.69
4	-4.64	3.09	8.69
5	-4.64	3.09	8.69

- $Z = \{z_1, z_2, \ldots, z_6, z_1', z_2'\}$, where z_t and $z_t' \in \mathcal{C}$ is the set of hidden-states.
- $\Gamma_1 = \{\gamma_1^{(1)}, \gamma_1^{(2)}\} = \{5, 6\}$ and $\Gamma_2 = \{\gamma_2^{(1)}, \gamma_2^{(2)}\} = \{8.9, 9.1\}$ such that $y_1 \in \Gamma_1$ and $y_2 \in \Gamma_2$ are the values attainable by y_1 and y_2, respectively.
- $J = 2$ and $K = 2$.

Using the preceding notations results in

$$
\begin{aligned}
Q(\theta|\theta_n) &= E\left\{\log p(Z, X|\theta, \Gamma_1, \Gamma_2)|X, \theta_n\right\} \\
&= E\left\{\log \prod_{t=1}^{6} p(z_t, x_t|\theta) \prod_{t'=1}^{2} p(z_{t'}, y_{t'}|y_{t'} \in \Gamma_{t'}, \theta)\Big| X, \Gamma_1, \Gamma_2, \theta_n\right\} \\
&= E\left\{\sum_{t=1}^{6} \log p(z_t, x_t|\theta)\Big| X, \Gamma_1, \Gamma_2, \theta_n\right\} \\
&\qquad + E\left\{\sum_{t'=1}^{2} \log p(z_{t'}, y_{t'}|y_{t'} \in \Gamma_{t'}, \theta)\Big| X, \Gamma_1, \Gamma_2, \theta_n\right\} \\
&= \sum_{t=1}^{6}\sum_{j=1}^{2} E\left\{\delta_t^{(j)}|x_t, \theta_n\right\} \log p(z_t, x_t|\theta) \\
&\qquad + \sum_{t'=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} E\left\{\beta_{t'}^{(k)}\delta_{t'}^{(j)}|y_{t'} \in \Gamma_{t'}, \theta_n\right\} \log p(z_{t'}, y_{t'}|y_{t'} \in \Gamma_{t'}, \theta) \\
&= \sum_{t=1}^{6}\sum_{j=1}^{2} h_n^{(j)}(x_t) \log\left[p(x_t|z_t = \mathcal{C}^{(j)}, \phi^{(j)})\pi^{(j)}\right] \\
&\qquad + \sum_{t'=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} P(y_{t'} = \gamma_{t'}^{(k)}|z_{t'} = \mathcal{C}^{(j)}, y_{t'} \in \Gamma_{t'}, \theta_n)
\end{aligned}
$$

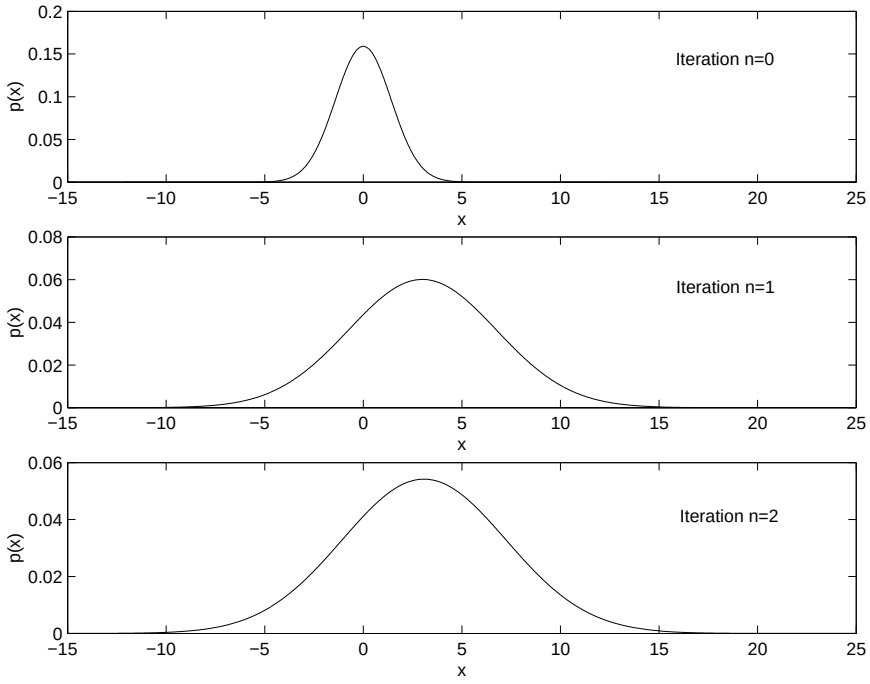


Figure 3.8. Movement of a Gaussian density function during the EM iterations. The density function is to fit the data containing a single cluster with partially observable data.

$$
\begin{aligned}
&\cdot P(z_{t'} = \mathcal{C}^{(j)} | y_{t'} \in \Gamma_{t'}, \theta_n) \\
&\cdot \log \left[p(y_{t'} = \gamma_{t'}^{(k)} | z_{t'} = \mathcal{C}^{(j)}, y_{t'} \in \Gamma_{t'}, \theta) P(z_{t'} = \mathcal{C}^{(j)} | \theta) \right] \\
= & \sum_{t=1}^{6} \sum_{j=1}^{2} h_n^{(j)}(x_t) \log \left[p(x_t | z_t = \mathcal{C}^{(j)}, \phi^{(j)}) \pi^{(j)} \right] \\
&+ \sum_{t'=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} P(y_{t'} = \gamma_{t'}^{(k)} | z_{t'} = \mathcal{C}^{(j)}, y_{t'} \in \Gamma_{t'}, \theta_n) h_n^{(j)}(y_{t'}) \\
&\cdot \log \left[p(y_{t'} = \gamma_{t'}^{(k)} | z_{t'} = \mathcal{C}^{(j)}, y_{t'} \in \Gamma_{t'}, \phi^{(j)}) \pi^{(j)} \right] \\
= & \sum_{t=1}^{6} \sum_{j=1}^{2} h_n^{(j)}(x_t) \log \left[p(x_t | z_t = \mathcal{C}^{(j)}, \phi^{(j)}) \pi^{(j)} \right] \\
&+ \sum_{t'=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} g_n^{(k,j)}(y_{t'}) h_n^{(j)}(y_{t'})
\end{aligned}
$$

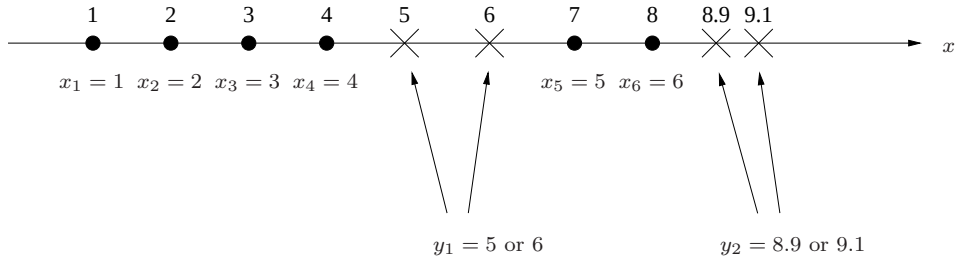


Figure 3.9. Single-dimension example illustrating the idea of hidden-states and partial-data.

$$\cdot \log \left[p(y_{t'} = \gamma_{t'}^{(k)} | z_{t'} = \mathcal{C}^{(j)}, y_{t'} \in \Gamma_{t'}, \phi^{(j)}) \pi^{(j)} \right], \tag{3.4.4}$$

where $g_n^{(k,j)} = P(y_{t'} = \gamma_{t'}^{(k)} | z_{t'} = \mathcal{C}^{(j)}, y_{t'} \in \Gamma_{t'}, \theta_n)$ is the posterior probability that $y_{t'}$ is equal to $\gamma_{t'}^{(k)}$ given that $y_{t'}$ is generated by cluster $\mathcal{C}^{(j)}$. Note that when the values of y_1 and y_2 are certain (e.g., it is known that $y_1 = 5$, and $\gamma_2^{(1)}$ and $\gamma_2^{(2)}$ become so close that we can consider $y_2 = 9$), then $K = 1$ and $\Gamma_1 = \{\gamma_1\} = \{5\}$ and $\Gamma_2 = \{\gamma_2\} = \{9\}$. In such cases, the second term of Eq. 3.4.4 becomes

$$\begin{aligned} &\sum_{t'=1}^{2} \sum_{j=1}^{2} P(y_{t'} = \gamma_{t'} | z_{t'} = \mathcal{C}^{(j)}, y_{t'} \in \Gamma_{t'}, \theta_n) h_n^{(j)}(y_{t'}) \\ &\quad \log \left[p(y_{t'} = \gamma_{t'} | z_{t'} = \mathcal{C}^{(j)}, y_{t'} \in \Gamma_{t'}, \phi^{(j)}) \pi^{(j)} \right] \\ &\quad = \sum_{t'=1}^{2} \sum_{j=1}^{2} h_n^{(j)}(y_{t'}) \log \left[p(y_{t'} | z_{t'} = \mathcal{C}^{(j)}, \phi^{(j)}) \pi^{(j)} \right]. \end{aligned} \tag{3.4.5}$$

Replacing the second term of Eq. 3.4.4 by Eq. 3.4.5 and seting $x_7 = y_1$ and $x_8 = y_2$ results in

$$Q(\theta | \theta_n) = \sum_{t=1}^{8} \sum_{j=1}^{2} h_n^{(j)}(x_t) \log \left[p(x_t | z_t = \mathcal{C}^{(j)}, \phi^{(j)}) \pi^{(j)} \right],$$

which is the Q-function of a GMM without partially unknown data with all observable data being certain.

3.5 Concluding Remarks

This chapter has detailed the algorithmic and convergence property of the EM algorithm. The standard EM has also been extended to a more general form called doubly-stochastic EM. A number of numerical examples were given to explain the algorithm's operation. The following summarizes the EM algorithm:

- EM offers an option of "soft" classification.
- EM offers a "soft pruning" mechanism. It is important because features with low probability should not be allowed to unduly influence the training of class parameters.
- EM naturally accommodates model-based clustering formulation.
- EM allows incorporation of prior information.
- EM training algorithm yields probabilistic parameters that are instrumental for media fusion. For linear-media fusion, EM plays a role in training the weights on the fusion layer. This will be elaborated on in subsequent chapters.

Problems

1. Assume that you are given a set of one-dimensional data $X = \{0, 1, 2, 3, 4, 3, 4, 5\}$. Find two cluster centers using

   (a) the K-means algorithm

   (b) the EM algorithm

   You may assume that the initial cluster centers are 0 and 5.

2. Compute the solutions of the single-cluster partial-data problem in Example 2 of Figure 3.3 with the following initial conditions:

$$\theta_0 = \{\mu_0, \sigma_0^2\} = \{-1, 0.5\}$$

3. In each iteration of the EM algorithm, the maximum-likelihood estimates of an M-center GMM are given by

$$\pi_j^{\text{new}} = \frac{\sum_{\mathbf{x}\in X} h_j(\mathbf{x})}{\sum_{\mathbf{x}\in X} 1}, \quad \mu_j^{\text{new}} = \frac{\sum_{\mathbf{x}\in X} h_j(\mathbf{x})\mathbf{x}}{\sum_{\mathbf{x}\in X} h_j(\mathbf{x})}, \text{ and}$$
$$\Sigma_j^{\text{new}} = \frac{\sum_{\mathbf{x}\in X} h_j(\mathbf{x})(\mathbf{x}-\mu_j^{\text{new}})(\mathbf{x}-\mu_j^{\text{new}})^T}{\sum_{\mathbf{x}\in X} h_j(\mathbf{x})},$$

   where

$$h_j(\mathbf{x}) = \frac{\pi_j^{\text{old}} p_j(\mathbf{x}|\mu_j^{\text{old}}, \Sigma_j^{\text{old}})}{\sum_{k=1}^{M} \pi_j^{\text{old}} p_k(\mathbf{x}|\mu_j^{\text{old}}, \Sigma_j^{\text{old}})}.$$

   X is the set of observed samples. $\{\pi_j^{\text{old}}, \mu_j^{\text{old}}, \Sigma_j^{\text{old}}\}_{j=1}^{M}$ are the maximum likelihood estimates of the last EM iteration, and

$$p_j(\mathbf{x}|\mu_j^{\text{old}}, \Sigma_j^{\text{old}}) = \frac{1}{(2\pi)^{\frac{D}{2}} \left|\Sigma_j^{\text{old}}\right|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_j^{\text{old}})^T(\Sigma_j^{\text{old}})^{-1}(\mathbf{x}-\mu_j^{\text{old}})\right\}.$$

   State the condition in which the EM algorithm reduces to the K-means algorithm.

4. You are given a set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ of T unlabeled samples drawn independently from a population whose density function is approximated by a GMM of the form

$$p(\mathbf{x}) \approx p(\mathbf{x}|\theta) = \sum_{j=1}^{M} \pi_j p(\mathbf{x}|\theta_j),$$

where $\theta = \{\theta_j\}_{j=1}^M = \{\mu_j, \Sigma_j\}_{j=1}^M$ are the means and covariance matrices of the component densities $\{p(\mathbf{x}|\theta_j)\}_{j=1}^M$. Assume that π_j are known, θ_j and $\theta_i (i \neq j)$ are functionally independent, and $\mathbf{x}_t$ are statistically independent.

   (a) Show that the log-likelihood function is $L(X|\theta) = \sum_{t=1}^{T} \log p(\mathbf{x}_t|\theta)$.

   (b) Show that the maximum-likelihood estimate $\hat{\theta}_i$ that maximizes L satisfies the conditions

$$\sum_{t=1}^{T} P(\pi_i|\mathbf{x}_t, \hat{\theta}) \frac{\partial}{\partial \theta_i} \log p(\mathbf{x}_t|\hat{\theta}_i) = 0 \quad i = 1, \ldots, M,$$

   where $P(\pi_i|\mathbf{x}_t, \hat{\theta}) = p(\mathbf{x}_t|\hat{\theta}_i)\pi_i / p(\mathbf{x}_t|\hat{\theta})$ is the posterior probability that the i-th cluster generates $\mathbf{x}_t$.

   (c) Hence, show that if $\{\Sigma_j\}_{j=1}^M$ are known, the maximum-likelihood estimate $\hat{\mu}_i$, $i = 1, \ldots, M$ are given by

$$\hat{\mu}_i = \frac{\sum_{t=1}^{T} P(\pi_i|\mathbf{x}_t, \hat{\theta})\mathbf{x}_t}{\sum_{t=1}^{T} P(\pi_i|\mathbf{x}_t, \hat{\theta})}.$$

   (d) Hence, state the conditions where the equation in Problem 4c reduces to the K-means algorithm. State also the condition where the K-means algorithm and the equation in Problem 4c give similar solutions.

5. Based on the normal distribution

$$p(\vec{x}; \vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})\right\}$$

in D-dimensions, show that the mean vector and covariance matrix that maximize the log-likelihood function

$$L(X; \vec{\mu}, \Sigma) = \log p(X; \vec{\mu}, \Sigma) = \log \prod_{\vec{x} \in X} p(\vec{x}; \vec{\mu}, \Sigma)$$

are, respectively, given by

$$\hat{\vec{\mu}} = \frac{1}{N} \sum_{\vec{x} \in X} \vec{x} \text{ and } \hat{\Sigma} = \frac{1}{N} \sum_{\vec{x} \in X} (\vec{x} - \hat{\vec{\mu}})(\vec{x} - \hat{\vec{\mu}})^T,$$

where X is a set of samples drawn from the population with distribution $p(\vec{x}; \vec{\mu}, \Sigma)$, N is the number of samples in X, and T denotes matrix transpose.

Hint: Use the derivatives

$$\frac{\partial}{\partial \vec{x}}\left(\vec{x}^T\vec{y}\right) = \vec{y},\ \frac{\partial}{\partial \vec{x}}\left(\vec{x}^T A\vec{y}\right) = A\vec{y},\ \frac{\partial}{\partial \vec{x}}\left(\vec{x}^T A\vec{x}\right) = A\vec{x} + A^T\vec{x},$$
$$\frac{\partial}{\partial A}\left(\vec{x}^T A\vec{x}\right) = \vec{x}\vec{x}^T, \text{ and } \frac{\partial |A|}{\partial A} = |A|\, A^{-1},$$

where A is a symmetric matrix.

6. Let $p(\mathbf{x}|\Sigma) \equiv \mathcal{N}(\mu, \Sigma)$ be a D-dimensional Gaussian density function with mean vector μ and covariance matrix Σ. Show that if μ is known and Σ is unknown, the maximum-likelihood estimate for Σ is given by

$$\hat{\Sigma} = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{x}_k - \mu)(\mathbf{x}_k - \mu)^T.$$

7. *LBF-Type EM Methods:* The fitness criterion for an RBF-type EM is determined by the closeness of a cluster member to a designated centroid of the cluster. As to its LBF-type counterpart, the fitness criterion hinges on the closeness of a subset of data to a linear plane (more exactly, hyperplane). In an exact fit, an ideal hyperplane is prescribed by a system of linear equations: $A\mathbf{x}_t + \mathbf{b} = \mathbf{0}$, where $\mathbf{x}_t$ is a data point on the plane. When the data are approximated by the hyperplane, then the following fitness function

$$\|A\mathbf{x}_t + \mathbf{b}\|^2$$

should approach zero. Sometimes, the data distribution can be better represented by more than one hyperplane. In a multiplane model, the data can be effectively represented by say N hyperplanes, which may be derived by minimizing the following LBF fitness function:

$$E(h, X) = \sum_t \sum_{j=1}^{K} h^{(j)}(\mathbf{x}_t)\|A\mathbf{x}_t + \mathbf{b}\|^2, \tag{3.5.1}$$

where $h^{(j)}(\mathbf{x}_t)$ is the membership probability satisfying $\sum_j h^{(j)}(\mathbf{x}_t) = 1$. If $h^{(j)}(\mathbf{x}_t)$ implements a hard-decision scheme, then $h^{(j)}(\mathbf{x}_t) = 1$ for the cluster members only, otherwise $h^{(j)}(\mathbf{x}_t) = 0$.

   (a) Compare the LBF-based formulation in Eq. 3.5.1 with the RBF-based formulation in Eq. 3.3.13.

   (b) Modify an RBF-based EM Matlab code so that it may be applicable to either RBF-based or LBF-based representation.

8. The following is a useful optimization formulation for the derivation of many EM algorithms. Given N known positive values u_n, where $n = 1, \ldots, N$. The problem is to determine the unknown parameters w_n to maximize the criterion function

$$\sum_{n=1}^{N} u_n \log w_n \tag{3.5.2}$$

under the constraints $w_n > 0$ and $\sum_{n=1}^{N} w_n = 1$.

   (a) Provide a mathematical proof that the optimal parameters have a closed-form solution:

$$w_n = \frac{u_n}{\sum_{n=1}^{N} u_n} \quad \text{for} \quad n = 1, \ldots, N.$$

   Hints: refer to Eqs. 3.2.23 through 3.2.26.

   (b) As an application example of the EM formulation in Eq. 3.2.10, what are the parameters corresponding to the known positive values u_n and the unknown positive values w_n. Hence, provide a physical meaning of the criterion function in Eq. 3.5.2.

9. As a numerical example of the preceding problem, given $u_1 = 3$, $u_2 = 4$, and $u_3 = 5$, and denote $x = u_1$, $y = u_2$, and $1 - x - y = u_3$, the criterion function can then be expressed as

$$3\log(x) \; + \; 4\log(y) + \; 5\log(1 - x - y).$$

   (a) Write a simple Matlab program to plot the criterion function over the admissible space $0 < x$, $0 < y$, and $x + y < 1$ (verify this range!).

   (b) Show numerically that the maximum value occurs at $x = \frac{1}{4}$ and $x = \frac{1}{3}$.

10. Suppose that someone is going to train a GMM by the EM algorithm. Let T denote the number of patterns, M the number of mixtures, and D the feature dimension. Show that the orders of computational complexity (in terms of multiplications) for each epoch in the E-step is $\mathcal{O}(TMD + TM)$ and that in the M-step is $\mathcal{O}(2TMD)$.

11. Assume that you are given the following observed data distribution:

$$x_1 = 1, x_2 = 2, x_3 = 4, x_4 = 8, \text{ and } x_5 = 9.$$

Assume also that when EM begins, $n = 0$ and

$$\begin{aligned} \theta_0 \; &= \; \left\{ \pi_0^{(1)}, \{\mu_0^{(1)}, \sigma_0^{(1)}\}, \pi_0^{(2)}, \{\mu_0^{(2)}, \sigma_0^{(2)}\} \right\} \\ &= \; \{0.5, \{1, 1\}, 0.5, \{9, 1\}\}\,. \end{aligned}$$

(a) Derive

$$h_0^{(1)}(x_t) = \frac{\frac{\pi_0^{(1)}}{\sigma_0^{(1)}} e^{-\frac{1}{2}(x_t-\mu^{(1)})^2/(\sigma_0^{(1)})^2}}{\sum_{k=1}^{2} \frac{\pi_0^{(k)}}{\sigma_0^{(k)}} e^{-\frac{1}{2}(x_t-\mu^{(k)})^2/(\sigma_0^{(k)})^2}}.$$

In a similar manner, derive $h_0^{(2)}(x_t)$.

(b) Substituting $X = \{1, 2, 4, 8, 9\}$ into your derivation to obtain the corresponding membership values.

(c) Substitute the derived membership values into Eqs. 3.2.17 through 3.2.19, and compute the values of the new parameters θ_1.

(d) To do more iterations, one can continue the algorithm by computing $Q(\theta|\theta_1)$, which can again be substituted into Eqs. 3.2.17 through 3.2.19 to obtain θ_2. Perform the iterative process until it converges.

12. It is difficult to provide any definitive assurance on the convergence of the EM algorithm to a global optimal solution. This is especially true when the data vector space has a very high dimension. Fortunately, for many inherently offline applications, there is no pressure to produce results in realtime speed. For such applications, the adoption of a user interface to pinpoint a reasonable initial estimate could prove helpful. To facilitate a visualization-based user interface, it is important to project from the original t-space (via a discriminant axis) to a two-dimensional (or three-dimensional) x-space [367, 370] (see Figure 2.7). Create a Matlab program to execute the following steps:

(a) Project the data set onto a reduced-dimensional x-space, via say PCA.

(b) Select initial cluster centers in the x-space by user's pinpoint. Based on the user-pinpointed membership, perform the EM algorithm in the x-space.

(c) Calculate the values of AIC and MDL to select the number of clusters (see the next problem).

(d) Trace the membership information back to the t-space, and use the membership function as the initial condition and further fine-tune the GMM clustering by the EM algorithm in the t-space.

13. One of the most important factors in data clustering is to select the proper number of clusters. Two prominent criteria for such selections are AIC and MDL [5, 314]. From the literature, find out the differences between AIC and MDL criteria. Do you have a preference and why?

14. Given a set of observed data X, develop Matlab code so that the estimated probability density $p(x)$ can be represented in terms of a set of means and variances.

15. Use Matlab to create a three-component Gaussian mixture distribution with different means and variances for each Gaussian component. Ensure that there is some overlap among the distributions.

    (a) Use 2-mean, 3-mean, and K-mean algorithms to cluster the data.

    (b) Compute the likelihood of the true parameters (means and variances that define the three Gaussian components) and the likelihood of your estimates. Which of your estimates is closest to the true distribution in the maximum-likelihood sense?

    (c) Compute the symmetric divergence between the true Gaussian distributions and your estimates. *Hint*: Given two Gaussian distributions Λ_j and Λ_k with mean vectors μ_j and μ_k and covariance matrices Σ_j and Σ_k, their symmetric divergence is

$$D(\Lambda_j||\Lambda_k) = \frac{1}{2}tr\left\{(\Sigma_j)^{-1}\Sigma_k + (\Sigma_k)^{-1}\Sigma_j - 2I\right\} + \frac{1}{2}(\mu_j - \mu_k)^T\left[(\Sigma_k)^{-1} + (\Sigma_j)^{-1}\right](\mu_j - \mu_k),$$

    where I is an identity matrix.

    (d) Repeat (a), (b), and (c) with the EM clustering algorithm.

16. Use Matlab to create a mixture density function with three Gaussian component densities. The prior probabilities should be as follows:

    (a) $P(\omega_1) = 0.2$, $P(\omega_2) = 0.3$, and $P(\omega_3) = 0.5$.

    (b) $P(\omega_1) = 0.1$, $P(\omega_2) = 0.1$, and $P(\omega_3) = 0.8$.

    Use 2-mean, 3-mean, and 4-mean VQ algorithms. Compute the likelihood between the data distribution and your estimate. Repeat the problem with the EM clustering algorithm.