The basic idea behind Expectation-Maximization

Jianxin Wu

LAMDA Group
National Key Lab for Novel Software Technology
Nanjing University, China
wujx2001@gmail.com

February 20, 2019

Contents

Statistical learning models are very important in many areas inside computer science, including but not confined to machine learning, computer vision, pattern recognition and data mining. It is also important in some deep learning models, such as the Restricted Boltzmann machine (RBM).

Statistical learning models have parameters, and estimating such parameters from data is one of the key problems in the study of such models. Expectation-

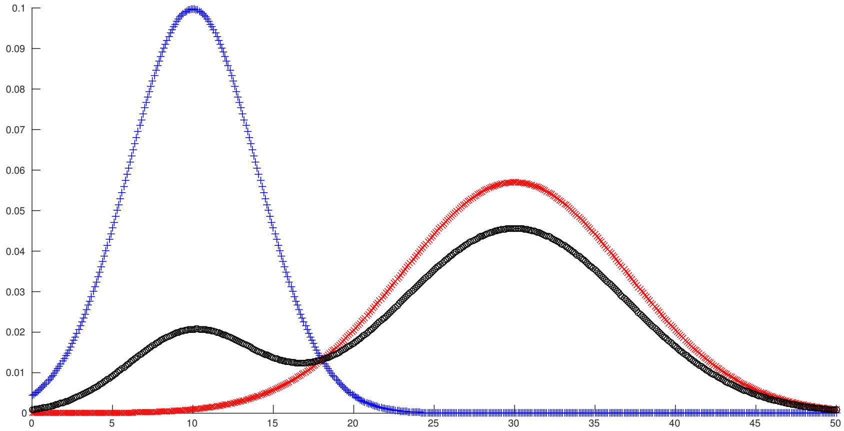

Figure 1: A simple GMM illustration.

Maximization (EM) is arguably the most widely used parameter estimation technique. Hence, it is worthwhile to know some basics of EM.

However, although EM is a must-have knowledge in studying statistical learning models, it is not easy for beginners. This chapter introduces the basic idea behind EM. We want to emphasize that the main purpose of this chapter is to introduce the *basic idea* (or, emphasizing the *intuition*) behind EM, not for covering all details of EM or presenting completely rigorous mathematical derivations.

1 GMM: A working example

Let us start from a simple working example: the Gaussian Mixture Model (GMM).

1.1 Gaussian mixture model

In Figure 1, we show three curves corresponding to three different probability density functions (p.d.f.) The blue curve is the p.d.f. of a normal distribution $N(10, 16)$, i.e., a Gaussian distribution with the mean $\mu = 10$ and the standard deviation $\sigma = 4$ (and $\sigma^2 = 16$). We denote this p.d.f. as

$$p_1(x) = N(x; 10, 16)\,.$$

The red curve is another normal distribution $N(30, 49)$ with $\mu = 30$ and $\sigma = 7$. Similarly, we denote it as

$$p_2(x) = N(x; 30, 49)\,.$$

We are interested in the black curve, whose first half is similar to the blue one, while the second half is similar to the red one. This curve is also the p.d.f. of a distribution, denoted by p_3. Since the black curve is similar to parts of the blue and red curves, it is reasonable to conjecture that p_3 is related to both p_1 and p_2.

Indeed, p_3 is a *weighted* combination of p_1 and p_2. In this example,

$$p_3(x) = 0.2p_1(x) + 0.8p_2(x)\,. \tag{1}$$

Because $0.2 + 0.8 = 1$, it is easy to verify that $p_3(x) \geq 0$ always holds and $\int_{-\infty}^{\infty} p_3(x)\,\mathrm{d}x = 1$. Hence, p_3 is a valid p.d.f.

p_3 is a *mixture* of two Gaussians (p_1 and p_2), hence a *Gaussian mixture model*. The definition of a GMM is in fact more general: it can have more than two components, and the Gaussians can be multivariate.

A GMM is a distribution whose p.d.f. has the following form:

$$\begin{aligned} p(\boldsymbol{x}) &= \sum_{i=1}^{N} \alpha_i N(\boldsymbol{x}; \boldsymbol{\mu}_i, \Sigma_i) & (2)\\ &= \sum_{i=1}^{N} \frac{\alpha_i}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right)\,, & (3) \end{aligned}$$

in which $\boldsymbol{x}$ is a d-dimensional random vector.

In this GMM, there are N Gaussian components, with the i-th Gaussian having the mean vector $\boldsymbol{\mu}_i \in \mathbb{R}^d$ and the covariance matrix $\Sigma_i \in \mathbb{R}^{d\times d}$. These Gaussian components are combined together using a linear combination, where the weight for the i-th component is α_i (called the *mixing coefficient*). The mixing coefficients must satisfy the following conditions

$$\sum_{i=1}^{N} \alpha_i = 1\,, \tag{4}$$

$$\alpha_i \geq 0, \forall\, i\,. \tag{5}$$

It is easy to verify that under these conditions, $p(\boldsymbol{x})$ is a valid multivariate probability density function.

1.2 The hidden variable interpretation

We can have a different interpretation of the Gaussian mixture model, using the hidden variable concept, as illustrated in Figure 2.

In Figure 2, the random variable X follows a Gaussian mixture model (cf. Equation 3). Its parameters are

$$\boldsymbol{\theta} = \{\alpha_i, \boldsymbol{\mu}_i, \Sigma_i\}_{i=1}^{N}\,. \tag{6}$$

If we want to sample an instance from this GMM, we could directly sample from the p.d.f. in Equation 3. However, there is another two-step way to perform the sampling.

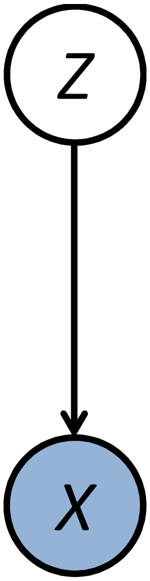

Figure 2: GMM as a graphical model.

Let us define a random variable Z. Z is a multinomial discrete distribution, taking values from the set $\{1, 2, \dots, N\}$. The probability that Z takes the value $Z = i$ is α_i, i.e., $\Pr(Z = i) = \alpha_i$, for $1 \le i \le N$. Then, the two-step sampling procedure is:

Step 1 Sample from Z, and get a value i ($1 \le i \le N$);

Step 2 Sample $\boldsymbol{x}$ from the i-th Gaussian component $N(\boldsymbol{\mu}_i, \Sigma_i)$.

It is easy to verify that the sample $\boldsymbol{x}$ achieved from this two-step sampling procedure follows the underlying GMM distribution in Equation 3.

In learning GMM parameters, we are given a sample set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_M\}$, where $\boldsymbol{x}_i$ are i.i.d. (independent and identically distributed) instances sampled from the p.d.f. in Equation 3. From this set of samples, we want to estimate or learn the GMM parameters $\boldsymbol{\theta} = \{\alpha_i, \boldsymbol{\mu}_i, \Sigma_i\}_{i=1}^N$.

Because we are given the samples $\boldsymbol{x}_i$, the random variable X (cf. Figure 2) is called an *observed* (or observable) random variable. As shown in Figure 2, observed random variables are usually shown as filled circles.

The random variable Z, however, is not observable, and is called a *hidden variable* (or a latent variable). Hidden variables are shown as empty circles, as the Z node in Figure 2.

1.3 What if we can observe the hidden variable?

In real applications, we do not know the value (or instantiation) of Z, because it is hidden (not observable). This fact makes estimating GMM parameters rather difficult, and techniques such as EM (the focus of this chapter) have to be employed.

However, for the sample set

$$\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\} ,$$

let us consider the scenario in which we can further suppose that some oracle has given us the value of Z:

$$\mathcal{Z} = \{z_1, z_2, \ldots, z_M\} .$$

In other words, we know that $\boldsymbol{x}_i$ is sampled from the z_i-th Gaussian component.

In this case, it is easy to estimate the parameters $\boldsymbol{\theta}$. First, we can find all those samples that are generated from the i-th component, and use $\mathcal{X}_i$ to denote this subset of samples. In precise mathematical languages,

$$\mathcal{X}_i = \{\boldsymbol{x}_j | z_j = i, 1 \le j \le M\} . \tag{7}$$

The mixing coefficient estimation is a simple counting. We can count the number of examples which are generated from the i-th Gaussian component as

$$m_i = |\mathcal{X}_i| ,$$

where $|\cdot|$ is the size (number of elements) of a set. Then, the maximum likelihood estimation for α_i is

$$\hat{\alpha}_i = \frac{m_i}{\sum_{j=1}^{N} m_j} = \frac{m_i}{M} . \tag{8}$$

Second, it is also easy to estimate the $\boldsymbol{\mu}_i$ and Σ_i parameters for any $1 \le i \le N$. The maximum likelihood estimation solutions are the same as those single Gaussian equations:

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{m_i} \sum_{\boldsymbol{x} \in \mathcal{X}_i} \boldsymbol{x} , \tag{9}$$

$$\hat{\Sigma}_i = \frac{1}{|m_i|} \sum_{\boldsymbol{x} \in \mathcal{X}_i} (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_i) (\boldsymbol{x} - \hat{\boldsymbol{\mu}}_i)^T . \tag{10}$$

In short, if we know the hidden variable's instantiations, the estimation is straightforward. Unfortunately, we are only given the observed sample set $\mathcal{X}$. The hidden variables $\mathcal{Z}$ are unknown to us. This fact complicates the entire parameter estimation process.

1.4 Can we imitate an oracle?

A natural question to ask ourselves is: if we do not have an oracle to teach us, can we imitate the oracle's teaching? In other words, can we *guess* the value of z_j for $\boldsymbol{x}_j$?

As we will see, EM is an iterative process, in which we use a variable t to denote the iteration index. We will update the parameters $\boldsymbol{\theta}$ in every iteration, and use $\boldsymbol{\theta}^{(t)}$ to denote its value in the t-th iteration. Then, a natural choice for our guess is to use the posterior $p(z_j|\boldsymbol{x}_j, \boldsymbol{\theta}^{(t)})$ as a replacement for z_j. This term is the probability of z_j given the sample $\boldsymbol{x}_j$ and the current parameter value $\boldsymbol{\theta}^{(t)}$, which seems to be the best educated guess we can have given the information that is handy.

In this guessing game, we have at least two issues in our way. First, an oracle is supposed to know everything, and will be able to tell us that $\boldsymbol{x}_7$ comes from the third Gaussian component with 100% confidence. If an oracle exists, we can simply say $z_7 = 3$ for this example $\boldsymbol{x}_7$. However, our guess will never be deterministic—it can at best be a probability distribution about the *random variable* z_j.

Hence, we will assume that for every observed sample $\boldsymbol{x}$, there is a corresponding hidden vector $\boldsymbol{z}$, whose values can be guessed but cannot be observed. We still use Z to denote the underlying random variable, and use

$$\mathcal{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_M\}$$

to denote the set of hidden vectors for the observable training examples

$$\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}\,.$$

In the GMM example, a vector $\boldsymbol{z}_j$ will have N dimensions, corresponding to the N Gaussian components. One and only one of these dimensions will be 1, and all others shall be 0.

Second, the guess we have about $\boldsymbol{z}_j$ is a distribution determined by the posterior $p(\boldsymbol{z}_j|\boldsymbol{x}_j, \boldsymbol{\theta}^{(t)})$. However, what we really want are values instead of a distribution. How are we going to use this guess? A common trick in statistical learning is to use its expectation. We will leave the details about how the expectation is computed and used to later sections.

2 An informal description of the EM algorithm

Now we are ready to give an informal description of the EM algorithm.

- We first initialize the values of $\boldsymbol{\theta}$ in any reasonable way;
- Then, we can estimate the best possible $\mathcal{Z}$ (or the expectation of its posterior distribution) using $\mathcal{X}$ and the current $\boldsymbol{\theta}$ estimation;
- With this $\mathcal{Z}$ estimation, we can find a better estimate of $\boldsymbol{\theta}$ using $\mathcal{X}$;
- A better $\boldsymbol{\theta}$ (combined with $\mathcal{X}$) will lead to a better guess of $\mathcal{Z}$;
- This process (estimating $\boldsymbol{\theta}$ and $\mathcal{Z}$ in alternating order) can proceed until the change in $\boldsymbol{\theta}$ is small (i.e., when the procedure converges).

In still more informal languages, after proper initialization of the parameters, we can perform

E-Step Find a better guess of the non-observable hidden variables, by using the data and current parameter values;

M-Step Find a better parameter estimation, by using the current guess for the hidden variables and the data;

Repeat Repeat the above two steps until convergence.

In the EM algorithm, the first step is usually called the Expectation step, abbreviated as the E-step; and the second step is usually called the Maximization step, abbreviated as the M-step. The EM algorithm repeats E- and M-steps in alternating order. When the algorithm converges, we get the desired parameter estimations.

3 The Expectation-Maximization algorithm

Now we will show more details of the EM algorithm. Suppose we are dealing with two sets of random variables: the observed variables X and the hidden variables Z. The joint p.d.f. is $p(X, Z; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the parameters. We are given a set of instances of X to learn the parameters, as

$$\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\} \,.$$

The task is to estimate $\boldsymbol{\theta}$ from $\mathcal{X}$.

For every $\boldsymbol{x}_j$, there is a corresponding $\boldsymbol{z}_j$, which we denote collectively as

$$\mathcal{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_M\} \,.$$

And, we want to clarify that $\boldsymbol{\theta}$ now include the parameters which are associated with Z. In the GMM example, $\{\boldsymbol{z}_i\}_{i=1}^{M}$ are estimates for Z, $\{\alpha_i, \boldsymbol{\mu}_i, \Sigma_i\}_{i=1}^{N}$ are parameters specifying X, and $\boldsymbol{\theta}$ include both sets of parameters.

3.1 Jointly-non-concave incomplete log-likelihood

If we use the maximum likelihood (ML) estimation technique, the ML estimate for $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{X}|\boldsymbol{\theta}) \,. \tag{11}$$

Or equivalently, we can maximize the log-likelihood

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ln p(\mathcal{X}|\boldsymbol{\theta}) \,, \tag{12}$$

because $\ln(\cdot)$ is a monotonically increasing function.

Then, parameter estimation becomes an optimization problem. We will use the notation $\ell\ell(\boldsymbol{\theta})$ to denote the log-likelihood, that is,

$$\ell\ell(\boldsymbol{\theta}) = \ln p(\mathcal{X}|\boldsymbol{\theta})\,. \tag{13}$$

Recent developments in optimization theory tell that we can generally consider a minimization problem as "easy" if it is *convex*, but non-convex problems are usually difficult to solve. Equivalently, a *concave* maximization problem is generally considered easy, while non-concave maximization is often difficult, because the negation of a convex function is a concave one, and vice versa.

Unfortunately, the log-likelihood is non-concave in most cases. Let us take the Gaussian mixture model as an example. The likelihood $p(\mathcal{X}|\boldsymbol{\theta})$ is

$$p(\mathcal{X}|\boldsymbol{\theta}) = \prod_{j=1}^{M}\left(\sum_{i=1}^{N}\frac{\alpha_i}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)\right)\right)\,. \tag{14}$$

The log-likelihood has the following form:

$$\sum_{j=1}^{M}\ln\left(\sum_{i=1}^{N}\frac{\alpha_i}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)\right)\right)\,. \tag{15}$$

This equation is non-concave with respect to $\{\alpha_i, \boldsymbol{\mu}_i, \Sigma_i\}_{i=1}^{n}$, the joint optimization variables. In other words, this is a difficult maximization problem.

We have two sets of random variables X and Z. The log-likelihood in Equation 15 is called the *incomplete data* log-likelihood because Z is not in that equation.

3.2 (Possibly) Concave complete data log-likelihood

The complete data log-likelihood is

$$\ln p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta})\,, \tag{16}$$

in which the hidden variable Z is assumed to be known. Knowing the hidden variable can often simplify the optimization. For example, the complete data log-likelihood may become concave.

Let us use GMM as an example once more. In GMM, the $\boldsymbol{z}_j$ vectors (which form $\mathcal{Z}$) is an N-dimensional vector with $N-1$ 0's and only one dimension with value 1. Hence, the complete data likelihood is

$$p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta}) = \prod_{j=1}^{M}\prod_{i=1}^{N}\left[\frac{\alpha_i}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}_j-\boldsymbol{\mu}_i)\right)\right]^{z_{ij}}\,. \tag{17}$$

This equation can be explained using the two-step sampling process. Let us assume $\boldsymbol{x}_j$ is generated by the i'-th Gaussian component. Then, if $i \neq i'$, we know that $z_{ij} = 0$, otherwise $z_{ij} = z_{i'j} = 1$. In other words, the term inside the two products will equal 1 for $N-1$ times when $z_{ij}=0$, and the remaining one entry will be evaluated to

$$\alpha_{i'} N(\boldsymbol{x}; \boldsymbol{\mu}_{i'}, \Sigma_{i'})\,,$$

which exactly matches the two step sampling procedure. In the two step sampling, the first step has probability $\alpha_{i'}$, and the second step has density $N(\boldsymbol{x}; \boldsymbol{\mu}_{i'}, \Sigma_{i'})$. These two steps are independent of each other, hence the product rule applies to obtain the above density.

Then, the complete data log-likelihood is

$$\sum_{j=1}^{M} \sum_{i=1}^{N} z_{ij} \left(\frac{1}{2} \left(\ln |\Sigma_i^{-1}| - (\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i) \right) + \ln \alpha_i \right) + \texttt{const}\,, \quad (18)$$

in which `const` refers to constant values that are not affected by the parameters $\boldsymbol{\theta}$.

Let us consider the scenario when the hidden variable z_{ij} is known, but $\alpha_i, \boldsymbol{\mu}_i$ and Σ_i are unknown. Here we suppose Σ_i is invertible for $1 \le i \le N$. Instead of considering parameters $(\boldsymbol{\mu}_i, \Sigma_i)$, we consider $(\boldsymbol{\mu}_i, \Sigma_i^{-1})$. In fact, it is more natural to understand this choice as using a variant of the canonical parameterization of a normal distribution, which was explained in detail in Chapter 13.

Now we can look at the three remaining items in Equation 18:

- It is well known that the log-determinant function

  $$\ln |\Sigma_i^{-1}| = \ln \det \left(\Sigma_i^{-1} \right)$$

  is concave with respect to Σ_i^{-1};

- It is also obvious that the quadratic term

  $$(\boldsymbol{z}_j - \boldsymbol{\mu}_i)^T \Sigma_i^{-1} (\boldsymbol{z}_j - \boldsymbol{\mu}_i)$$

  is jointly convex with respect to variables $(\boldsymbol{\mu}_i, \Sigma_i^{-1})$, which directly implies that its negation is concave;

- It is easy to prove the logarithm function $\ln \alpha_i$ is concave with respect to α_i.

Because the sum of concave functions is concave, Equation 18 can be efficiently solved.

From this optimization perspective, we can understand the EM algorithm from a different point of view. Although the original maximum likelihood parameter estimation problem is difficult to solve (jointly non-concave), the EM algorithm can usually (but not always) make concave subproblems, hence becoming efficiently solvable.

3.3 The general EM derivation

Now we talk about EM in the general sense. We have observable variables X and samples $\mathcal{X}$. We also have hidden variables Z and unobservable samples $\mathcal{Z}$. The overall system parameters are denoted by $\boldsymbol{\theta}$.

The parameter learning problem tries to find optimal parameters $\hat{\boldsymbol{\theta}}$ by maximizing the incomplete data log-likelihood

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ln p(\mathcal{X}|\boldsymbol{\theta})\,. \tag{19}$$

As a simplification, we assume Z is discrete, and hence

$$p(\mathcal{X}|\boldsymbol{\theta}) = \sum_{\mathcal{Z}} p(\mathcal{X},\mathcal{Z}|\boldsymbol{\theta})\,. \tag{20}$$

However, this assumption is mainly for notational simplicity. If Z is continuous, we can replace the summation with an integral.

Although we have mentioned previously that we can use the posterior of Z, i.e., $p(\mathcal{Z}|\mathcal{X},\boldsymbol{\theta})$ as our guess, it is also interesting to observe what will happen to the complete data likelihood if we use an *arbitrary* distribution for Z (and hence understand why the posterior is special and why we should use it).

Let q be any valid probability distribution for Z. We can measure how different is q to the posterior using the classic Kullback-Leibler (KL) divergence measure, as

$$\mathrm{KL}(q\|p) = -\sum_{\mathcal{Z}} q(\mathcal{Z}) \ln\left(\frac{p(\mathcal{Z}|\mathcal{X},\boldsymbol{\theta})}{q(\mathcal{Z})}\right)\,. \tag{21}$$

The probability theory tells us that

$$\begin{aligned} p(\mathcal{X}|\boldsymbol{\theta}) &= \frac{p(\mathcal{X},\mathcal{Z}|\boldsymbol{\theta})}{p(\mathcal{Z}|\mathcal{X},\boldsymbol{\theta})} && (22)\\ &= \frac{p(\mathcal{X},\mathcal{Z}|\boldsymbol{\theta})}{q(\mathcal{Z})}\frac{q(\mathcal{Z})}{p(\mathcal{Z}|\mathcal{X},\boldsymbol{\theta})}\,. && (23) \end{aligned}$$

Hence,

$$\begin{aligned} \ln p(\mathcal{X}|\boldsymbol{\theta}) &= \left(\sum_{\mathcal{Z}} q(\mathcal{Z})\right) \ln p(\mathcal{X}|\boldsymbol{\theta}) && (24)\\ &= \sum_{\mathcal{Z}} q(\mathcal{Z}) \ln p(\mathcal{X}|\boldsymbol{\theta}) && (25)\\ &= \sum_{\mathcal{Z}} q(\mathcal{Z}) \ln\left(\frac{p(\mathcal{X},\mathcal{Z}|\boldsymbol{\theta})}{q(\mathcal{Z})}\frac{q(\mathcal{Z})}{p(\mathcal{Z}|\mathcal{X},\boldsymbol{\theta})}\right) && (26)\\ &= \sum_{\mathcal{Z}} \left(q(\mathcal{Z}) \ln\frac{p(\mathcal{X},\mathcal{Z}|\boldsymbol{\theta})}{q(\mathcal{Z})} - q(\mathcal{Z}) \ln\frac{p(\mathcal{Z}|\mathcal{X},\boldsymbol{\theta})}{q(\mathcal{Z})}\right) && (27)\\ &= \sum_{\mathcal{Z}} q(\mathcal{Z}) \ln\frac{p(\mathcal{X},\mathcal{Z}|\boldsymbol{\theta})}{q(\mathcal{Z})} + \mathrm{KL}(q\|p) && (28) \end{aligned}$$

$$= \mathcal{L}(q, \boldsymbol{\theta}) + \mathrm{KL}(q\|p)\,. \tag{29}$$

We have decomposed the incomplete data log-likelihood into two terms. The first term is $\mathcal{L}(q, \boldsymbol{\theta})$, defined as

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{\mathcal{Z}} q(\mathcal{Z}) \ln \frac{p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta})}{q(\mathcal{Z})}\,. \tag{30}$$

The second term is a KL-divergence between q and the posterior

$$\mathrm{KL}(q\|p) = -\sum_{\mathcal{Z}} q(\mathcal{Z}) \ln \left(\frac{p(\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta})}{q(\mathcal{Z})} \right)\,, \tag{31}$$

which was copied from Equation 21.

There are some nice properties of the KL-divergence. For example,

$$D(q\|p) \geq 0 \tag{32}$$

always holds, and the equality is true if and only if $q = p$ (cf. Chapter 10). One direct consequence of this property is that

$$\mathcal{L}(q, \boldsymbol{\theta}) \leq \ln p(\mathcal{X}|\boldsymbol{\theta}) \tag{33}$$

always holds, and

$$\mathcal{L}(q, \boldsymbol{\theta}) = \ln p(\mathcal{X}|\boldsymbol{\theta}) \text{ if and only if } q(\mathcal{Z}) = p(\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta})\,. \tag{34}$$

In other words, we have found a lower bound of $\ln p(\mathcal{X}|\boldsymbol{\theta})$. Hence, in order to maximize $\ln p(\mathcal{X}|\boldsymbol{\theta})$, we can perform two steps.

- The first step is to make the lower bound $\mathcal{L}(q, \boldsymbol{\theta})$ equal $\ln p(\mathcal{X}|\boldsymbol{\theta})$. As aforementioned, we know the equality holds if and only if

  $$\hat{q}(\mathcal{Z}) = p(\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta})\,.$$

  With this optimal $\hat{q}(\mathcal{Z})$ value, we have

  $$\ln p(\mathcal{X}|\boldsymbol{\theta}) = \mathcal{L}(\hat{q}, \boldsymbol{\theta})\,, \tag{35}$$

  and $\mathcal{L}$ only depends on $\boldsymbol{\theta}$ now. This is the Expectation step (E-step) in the EM algorithm.

- In the second step, we can maximize $\mathcal{L}(\hat{q}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Since $\ln p(\mathcal{X}|\boldsymbol{\theta}) = \mathcal{L}(\hat{q}, \boldsymbol{\theta})$, an increase of $\mathcal{L}(\hat{q}, \boldsymbol{\theta})$ also means an increase of the log-likelihood $\ln p(\mathcal{X}|\boldsymbol{\theta})$. And, because we are maximizing $\mathcal{L}(\hat{q}, \boldsymbol{\theta})$ in this step, *the log-likelihood will always increase* if we are not already at a local maximum of the log-likelihood. This is the Maximization step (M-step) in the EM algorithm.

3.4 The E- & M-steps

In the E-step, we already know that we should set

$$\hat{q}(\mathcal{Z}) = p(\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta}) \,, \tag{36}$$

which is straightforward (at least in its mathematical form).

Then, how shall we maximize $\mathcal{L}(\hat{q}, \boldsymbol{\theta})$? We can substitute $\hat{q}$ into the definition of $\mathcal{L}$, and then find the optimal $\boldsymbol{\theta}$ that maximizes $\mathcal{L}$ after plugging in $\hat{q}$. However, note that $\hat{q}$ involves $\boldsymbol{\theta}$ too. Hence, we need some more notations.

Suppose we are in the t-th iteration. In the E-step, $\hat{q}$ is computed using the current parameters, as

$$\hat{q}(\mathcal{Z}) = p(\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta}^{(t)}) \,. \tag{37}$$

Then, $\mathcal{L}$ becomes

$$\begin{aligned}
\mathcal{L}(\hat{q}, \boldsymbol{\theta}) &= \sum_{\mathcal{Z}} \hat{q}(\mathcal{Z}) \ln \frac{p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta})}{\hat{q}(\mathcal{Z})} && (38) \\
&= \sum_{\mathcal{Z}} \hat{q}(\mathcal{Z}) \ln p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta}) - \hat{q}(\mathcal{Z}) \ln \hat{q}(\mathcal{Z}) && (39) \\
&= \sum_{\mathcal{Z}} p(\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta}^{(t)}) \ln p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta}) + \texttt{const} \,, && (40)
\end{aligned}$$

in which $\texttt{const} = -\hat{q}(\mathcal{Z}) \ln \hat{q}(\mathcal{Z})$ does not involve the variable $\boldsymbol{\theta}$, hence can be ignored.

The term remaining is in fact an expectation, which we denote as $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$,

$$\begin{aligned}
\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) &= \sum_{\mathcal{Z}} p(\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta}^{(t)}) \ln p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta}) && (41) \\
&= \mathbb{E}_{\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta}^{(t)}} \left[\ln p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta}) \right] \,. && (42)
\end{aligned}$$

That is, in the E-step, we compute the posterior of Z. In the M-step, we compute the expectation of the complete data log-likelihood $\ln p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta})$ with respect to the posterior distribution $p(\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta}^{(t)})$, and we maximize the expectation to get a better parameter estimate:

$$\begin{aligned}
\boldsymbol{\theta}^{(t+1)} &= \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) && (43) \\
&= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta}^{(t)}} \left[\ln p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta}) \right] \,. && (44)
\end{aligned}$$

Thus, three computations are involved in EM: 1) posterior, 2) expectation, 3) maximization. We treat 1) as the E-step, and 2)+3) as the M-step. Some researchers prefer to treat 1)+2) as the E-step, and 3) as the M-step. However, no matter how the computations are attributed to different steps, the EM algorithm does not change.

3.5 The EM algorithm

Now we are ready to write down the EM algorithm.

Algorithm 1 The Expectation-Maximization Algorithm

```
1: $t \leftarrow 0$
2: Initialize the parameters to $\boldsymbol{\theta}^{(0)}$
3: The **E**(expectation)-step: Find $p(\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta}^{(t)})$
4: The **M**(aximization)-step.1: Find the expectation
       $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\mathcal{Z}|\mathcal{X}, \boldsymbol{\theta}^{(t)}} \left[ \ln p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta}) \right]$   (45)
5: The **M**(aximization)-step.2: Find a new parameter estimate
       $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\arg\max}\, \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$   (46)
6: $t \leftarrow t + 1$
7: If the log-likelihood has not converged, go to the E-step again (Line 3)
```

3.6 Will EM converge?

The analysis of EM's convergence property is a complex topic. However, it is easy to show that the EM algorithm will help achieve higher likelihood and converge to a local maximum.

Let us consider two time steps $t-1$ and t. From Equation 35, we get that:

$$\mathcal{L}(\hat{q}^{(t)}, \boldsymbol{\theta}^{(t)}) = \ln p(\mathcal{X}|\boldsymbol{\theta}^{(t)})\,, \tag{47}$$

$$\mathcal{L}(\hat{q}^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) = \ln p(\mathcal{X}|\boldsymbol{\theta}^{(t-1)})\,. \tag{48}$$

Note that we have added the time index to the superscript of $\hat{q}$ to emphasize that $\hat{q}$ also changes among iterations.

Now because at the $(t-1)$-th iteration

$$\boldsymbol{\theta}^{(t)} = \underset{\boldsymbol{\theta}}{\arg\max}\, \mathcal{L}(\hat{q}^{(t-1)}, \boldsymbol{\theta})\,, \tag{49}$$

we have

$$\mathcal{L}(\hat{q}^{(t-1)}, \boldsymbol{\theta}^{(t)}) \geq \mathcal{L}(\hat{q}^{(t-1)}, \boldsymbol{\theta}^{(t-1)})\,. \tag{50}$$

Similarly, at the t-th iteration, based on Equation 33 and Equation 35, we have

$$\mathcal{L}(\hat{q}^{(t-1)}, \boldsymbol{\theta}^{(t)}) \leq \ln p(\mathcal{X}|\boldsymbol{\theta}^{(t)}) = \mathcal{L}(\hat{q}^{(t)}, \boldsymbol{\theta}^{(t)})\,. \tag{51}$$

Putting these equations together, we get

$$\begin{aligned}
\ln p(\mathcal{X}|\boldsymbol{\theta}^{(t)}) &= \mathcal{L}(\hat{q}^{(t)}, \boldsymbol{\theta}^{(t)}) && \text{[Use (47)]} && (52) \\
&\geq \mathcal{L}(\hat{q}^{(t-1)}, \boldsymbol{\theta}^{(t)}) && \text{[Use (51)]} && (53) \\
&\geq \mathcal{L}(\hat{q}^{(t-1)}, \boldsymbol{\theta}^{(t-1)}) && \text{[Use (50)]} && (54)
\end{aligned}$$

$$= \ln p(\mathcal{X}|\boldsymbol{\theta}^{(t-1)})\,. \qquad \text{[Use (48)]} \tag{55}$$

Hence, EM will converge to a local maximum of the likelihood. However, the analysis of its convergence rate is very complex and beyond the scope of this introductory book.

4 EM for GMM

Now we can apply the EM algorithm to GMM.

The first thing is to compute the posterior. Using Bayes' theorem, we have

$$p(z_{ij}|\boldsymbol{x}_j, \boldsymbol{\theta}^{(t)}) = \frac{p(\boldsymbol{x}_j, z_{ij}|\boldsymbol{\theta}^{(t)})}{p(\boldsymbol{x}_j|\boldsymbol{\theta}^{(t)})}\,, \tag{56}$$

in which z_{ij} can be 0 or 1, and $z_{ij} = 1$ is true if and only if $\boldsymbol{x}_j$ is generated by the i-th Gaussian component.

Next, we will compute the $\mathcal{Q}$ function, which is the expectation of the complete data log-likelihood $\ln p(\mathcal{X}, \mathcal{Z}|\boldsymbol{\theta})$ with respect to the posterior distribution we just found. The GMM complete data log-likelihood was already computed in Equation 18. For easier reference, we copy this equation here:

$$\sum_{j=1}^{M}\sum_{i=1}^{N} z_{ij}\left(\frac{1}{2}\left(\ln|\Sigma_i^{-1}| - (\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)\right) + \ln\alpha_i\right) + \texttt{const}\,, \tag{57}$$

The expectation of Equation 57 with respect to Z is

$$\sum_{j=1}^{M}\sum_{i=1}^{N} \gamma_{ij}\left(\frac{1}{2}\left(\ln|\Sigma_i^{-1}| - (\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}_j - \boldsymbol{\mu}_i)\right) + \ln\alpha_i\right)\,, \tag{58}$$

where the constant term is ignored and γ_{ij} is the expectation of $z_{ij}|\boldsymbol{x}_j, \boldsymbol{\theta}^{(t)}$. In other words, we need to compute the expectation of the conditional distribution defined by Equation 56.

In Equation 56, the denominator does not depend on Z, and $p(\boldsymbol{x}_j|\boldsymbol{\theta}^{(t)})$ equals $\sum_{i=1}^{N}\alpha_i^{(t)}N(\boldsymbol{x}_j; \boldsymbol{\mu}_i^{(t)}, \Sigma_i^{(t)})$. For the numerator, we can directly compute its expectation, as

$$\mathbb{E}\left[p(\boldsymbol{x}_j, z_{ij}|\boldsymbol{\theta}^{(t)})\right] = \mathbb{E}\left[p(z_{ij}|\boldsymbol{\theta}^{(t)})p(\boldsymbol{x}_j|z_{ij}, \boldsymbol{\theta}^{(t)})\right]\,. \tag{59}$$

Note that when $z_{ij} = 0$, we always have $p(\boldsymbol{x}_j|z_{ij}, \boldsymbol{\theta}^{(t)}) = 0$. Thus,

$$\mathbb{E}\left[p(z_{ij}|\boldsymbol{\theta}^{(t)})p(\boldsymbol{x}_j|z_{ij}, \boldsymbol{\theta}^{(t)})\right] = \Pr(z_{ij} = 1)p(\boldsymbol{x}_j|\boldsymbol{\mu}_i^{(t)}, \Sigma_i^{(t)}) \tag{60}$$

$$= \alpha_i^{(t)}N(\boldsymbol{x}_j; \boldsymbol{\mu}_i^{(t)}, \Sigma_i^{(t)})\,. \tag{61}$$

Hence, we have

$$\gamma_{ij} = \mathbb{E}\left[z_{ij} | \boldsymbol{x}_j, \boldsymbol{\theta}^{(t)}\right] \propto \alpha_i^{(t)} N(\boldsymbol{x}_j; \boldsymbol{\mu}_i^{(t)}, \Sigma_i^{(t)})\,, \tag{62}$$

or,

$$\gamma_{ij} = \mathbb{E}\left[z_{ij} | \boldsymbol{x}_j, \boldsymbol{\theta}^{(t)}\right] = \frac{\alpha_i^{(t)} N(\boldsymbol{x}_j; \boldsymbol{\mu}_i^{(t)}, \Sigma_i^{(t)}))}{\sum_{k=1}^{N} \alpha_k^{(t)} N(\boldsymbol{x}_j; \boldsymbol{\mu}_k^{(t)}, \Sigma_k^{(t)})} \tag{63}$$

for $1 \le i \le N$, $1 \le j \le M$.

After γ_{ij} is computed, Equation 58 is completely specified. We start the optimization from α_i. Because there is a constraint that $\sum_{i=1}^{N} \alpha_i = 1$, we use the Lagrange multiplier method. After removing irrelevant terms, we get

$$\sum_{j=1}^{M} \sum_{i=1}^{N} \gamma_{ij} \ln \alpha_i + \lambda \left(\sum_{i=1}^{N} \alpha_i - 1 \right)\,. \tag{64}$$

Setting the derivative to 0 gives us that for any $1 \le i \le N$,

$$\frac{\sum_{j=1}^{M} \gamma_{ij}}{\alpha_i} + \lambda = 0 \tag{65}$$

or,

$$\alpha_i = -\frac{\sum_{j=1}^{M} \gamma_{ij}}{\lambda}\,.$$

Because $\sum_{i=1}^{N} \alpha_i = 1$, we know that

$$\lambda = -\sum_{j=1}^{M} \sum_{i=1}^{N} \gamma_{ij}\,.$$

Hence,

$$\alpha_i = \frac{\sum_{j=1}^{M} \gamma_{ij}}{\sum_{j=1}^{M} \sum_{i=1}^{N} \gamma_{ij}}\,.$$

For notational simplicity, we define

$$m_i = \sum_{j=1}^{M} \gamma_{ij}\,. \tag{66}$$

From the definition of γ_{ij}, it is easy to prove that

$$\sum_{i=1}^{N} m_i = \sum_{i=1}^{N} \sum_{j=1}^{M} \gamma_{ij} \tag{67}$$

$$= \sum_{j=1}^{M} \left(\sum_{i=1}^{N} \gamma_{ij} \right) \tag{68}$$

$$= \sum_{j=1}^{M} 1 \tag{69}$$

$$= M\,. \tag{70}$$

Then, we get the updating rule for α_i:

$$\alpha_i^{(t+1)} = \frac{m_i}{M}\,. \tag{71}$$

Furthermore, using similar steps in deriving the single Gaussian equations, it is easy to show that for any $1 \le i \le N$,

$$\boldsymbol{\mu}_i^{(t+1)} = \frac{\sum_{j=1}^{M} \gamma_{ij} \boldsymbol{x}_j}{m_i}\,, \tag{72}$$

$$\Sigma_i^{(t+1)} = \frac{\sum_{j=1}^{M} \gamma_{ij} \left(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{(t+1)}\right) \left(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{(t+1)}\right)^T}{m_i}\,. \tag{73}$$

Putting these results together, we have the complete set of updating rules for GMM. If at iteration t, the parameter are estimated as $\alpha_i^{(t)}$, $\boldsymbol{\mu}_i^{(t)}$, and $\Sigma_i^{(t)}$ for $1 \le i \le N$, the EM algorithm updates these parameters as (for $1 \le i \le N$, $1 \le j \le M$)

$$\gamma_{ij} = \frac{\alpha_i N(\boldsymbol{x}_j; \boldsymbol{\mu}_i^{(t)}, \Sigma_i^{(t)})}{\sum_{k=1}^{N} \alpha_k^{(t)} N(\boldsymbol{x}_j; \boldsymbol{\mu}_k^{(t)}, \Sigma_k^{(t)})}\,, \tag{74}$$

$$m_i = \sum_{j=1}^{M} \gamma_{ij}\,, \tag{75}$$

$$\boldsymbol{\mu}_i^{(t+1)} = \frac{\sum_{j=1}^{M} \gamma_{ij} \boldsymbol{x}_j}{m_i}\,, \tag{76}$$

$$\Sigma_i^{(t+1)} = \frac{\sum_{j=1}^{M} \gamma_{ij} \left(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{(t+1)}\right) \left(\boldsymbol{x}_j - \boldsymbol{\mu}_i^{(t+1)}\right)^T}{m_i}\,. \tag{77}$$

Exercises

1. Derive the updating equations for Gaussian Mixture Models by yourself. You should not refer to Section 4 during your derivation. If you have just finished reading Section 4, wait for at least 2 to 3 days before working on this problem.

2. In this problem, we will use the Expectation-Maximization method to learn parameters in a hidden Markov model (HMM). As will be shown in this problem, the Baum-Welch algorithm is indeed performing EM updates. To work out the solution for this problem, you will also need knowledge and facts learned in the HMM chapter (Chapter 12) and the information theory chapter (Chapter 10).

   We will use the notations in the HMM chapter. For your convenience, the notations are repeated as follows.

   - There are N discrete states, denoted by symbols $S_1, S_2, \dots, S_N$.
   - There are M output discrete symbols, denoted by $V_1, V_2, \dots, V_M$.
   - Assuming one sequence with T time steps, whose hidden state is Q_t and whose observed output is O_t at time t ($1 \le t \le T$). We use q_t and o_t to denote the indexes for state and output symbols at time t, respectively, i.e., $Q_t = S_{q_t}$ and $O_t = V_{o_t}$.
   - The notation $1:t$ denotes all the ordered time steps between 1 and t. For example, $o_{1:T}$ is the sequence of all observed output symbols.
   - An HMM has parameters $\lambda = (\boldsymbol{\pi}, A, B)$, where $\boldsymbol{\pi} \in \mathbb{R}^N$ specifies the initial state distribution, $A \in \mathbb{R}^{N\times N}$ is the state transition matrix, and $B \in \mathbb{R}^{N\times M}$ is the observation probability matrix. Note that $A_{ij} = \Pr(Q_t = S_j | Q_{t-1} = S_i)$ and $b_j(k) = \Pr(O_t = V_k | Q_t = S_j)$ are elements of A and B, respectively.
   - In this problem, we use a variable r to denote the index of EM iterations, which starts from $r = 1$. Hence, $\lambda^{(1)}$ are the initial parameters.
   - Various probabilities have been defined in the HMM chapter, denoted by $\alpha_t(i)$, $\beta_t(i)$, $\gamma_t(i)$, $\delta_t(i)$ and $\xi_t(i,j)$. In this problem, we assume that at the r-th iterations, $\lambda^{(r)}$ are known and these probabilities are computed using $\lambda^{(r)}$.

   The purpose of this problem is to use the EM algorithm to find $\lambda^{(r+1)}$ using a training sequence $o_{1:T}$ and $\lambda^{(r)}$, by treating Q and O as the hidden and observed random variables, respectively.

   (a) Suppose the hidden variables can be observed as $S_{q_1}, S_{q_2}, \dots, S_{q_T}$. Show that the complete data log-likelihood is

$$\ln \pi_{q_1} + \sum_{t=1}^{T-1} \ln A_{q_t q_{t+1}} + \sum_{t=1}^{T} \ln b_{q_t}(o_t)\,. \tag{78}$$

(b) The expectation of Equation 78 with respect to the hidden variables Q_t (conditioned on $o_{1:T}$ and $\lambda^{(r)}$) forms an auxiliary function $\mathcal{Q}(\lambda, \lambda^r)$ (the E-step). Show that the expectation of the first term in Equation 78 equals $\sum_{i=1}^{N} \gamma_1(i) \ln \pi_i$, i.e.,

$$\mathbb{E}_{Q_{1:T}}[\ln \pi_{Q_1}] = \sum_{i=1}^{N} \gamma_1(i) \ln \pi_i \,. \tag{79}$$

(c) Because the parameter $\boldsymbol{\pi}$ only hinges on Equation 79, the update rule for $\boldsymbol{\pi}$ can be found by maximizing this equation. Prove that we should set

$$\pi_i^{(r+1)} = \gamma_1(i)$$

in the M-step. Note that $\gamma_1(i)$ is computed using $\lambda^{(r)}$ as parameter values. (Hint: The right hand side of Equation 79 is related to the cross entropy.)

(d) The second part of the E-step calculates the expectation of the middle term in Equation 78. Show that

$$\mathbb{E}_{Q_{1:T}}\left[\sum_{t=1}^{T-1} \ln A_{q_t q_{t+1}}\right] = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\sum_{t=1}^{T-1} \xi_t(i,j)\right) \ln A_{ij} \,. \tag{80}$$

(e) For the M-step relevant to A, prove that we should set

$$A_{ij}^{(r+1)} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \,. \tag{81}$$

(f) The final part of the E-step calculates the expectation of the last term in Equation 78. Show that

$$\mathbb{E}_{Q_{1:T}}\left[\sum_{t=1}^{T} \ln b_{q_t}(o_t)\right] = \sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{t=1}^{T} [\![o_t = k]\!] \gamma_t(j) \,. \tag{82}$$

(g) For the M-step relevant to B, prove that we should set

$$b_j^{(r+1)}(k) = \frac{\sum_{t=1}^{T} [\![o_t = k]\!] \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \,, \tag{83}$$

in which $[\![\cdot]\!]$ is the indicator function.

(h) Are these results obtained using EM the same as those in the Baum-Welch?