INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • (510) 643-9153 • FAX (510) 643-7684

A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models

Jeff A. Bilmes (bilmes@cs.berkeley.edu)
International Computer Science Institute
Berkeley CA, 94704
and
Computer Science Division
Department of Electrical Engineering and Computer Science
U.C. Berkeley

TR-97-021
April 1998

Abstract

We describe the maximum-likelihood parameter estimation problem and how the Expectation-Maximization (EM) algorithm can be used for its solution. We first describe the abstract form of the EM algorithm as it is often given in the literature. We then develop the EM parameter estimation procedure for two applications: 1) finding the parameters of a mixture of Gaussian densities, and 2) finding the parameters of a hidden Markov model (HMM) (i.e., the Baum-Welch algorithm) for both discrete and Gaussian mixture observation models. We derive the update equations in fairly explicit detail but we do not prove any convergence properties. We try to emphasize intuition rather than mathematical rigor.

1 Maximum-likelihood

Recall the definition of the maximum-likelihood estimation problem. We have a density function $p(\mathbf{x}|\Theta)$ that is governed by the set of parameters Θ (e.g., p might be a set of Gaussians and Θ could be the means and covariances). We also have a data set of size N, supposedly drawn from this distribution, i.e., $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. That is, we assume that these data vectors are independent and identically distributed (i.i.d.) with distribution p. Therefore, the resulting density for the samples is

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} p(\mathbf{x}_i|\Theta) = \mathcal{L}(\Theta|\mathcal{X}).$$

This function $\mathcal{L}(\Theta|\mathcal{X})$ is called the likelihood of the parameters given the data, or just the likelihood function. The likelihood is thought of as a function of the parameters Θ where the data $\mathcal{X}$ is fixed. In the maximum likelihood problem, our goal is to find the Θ that maximizes $\mathcal{L}$. That is, we wish to find Θ^* where

$$\Theta^* = \underset{\Theta}{\text{argmax}}\ \mathcal{L}(\Theta|\mathcal{X}).$$

Often we maximize $\log(\mathcal{L}(\Theta|\mathcal{X}))$ instead because it is analytically easier.

Depending on the form of $p(\mathbf{x}|\Theta)$ this problem can be easy or hard. For example, if $p(\mathbf{x}|\Theta)$ is simply a single Gaussian distribution where $\Theta = (\mu, \sigma^2)$, then we can set the derivative of $\log(\mathcal{L}(\Theta|\mathcal{X}))$ to zero, and solve directly for μ and σ^2 (this, in fact, results in the standard formulas for the mean and variance of a data set). For many problems, however, it is not possible to find such analytical expressions, and we must resort to more elaborate techniques.

2 Basic EM

The EM algorithm is one such elaborate technique. The EM algorithm [ALR77, RW84, GJ95, JJ94, Bis95, Wu83] is a general method of finding the maximum-likelihood estimate of the parameters of an underlying distribution from a given data set when the data is incomplete or has missing values.

There are two main applications of the EM algorithm. The first occurs when the data indeed has missing values, due to problems with or limitations of the observation process. The second occurs when optimizing the likelihood function is analytically intractable but when the likelihood function can be simplified by assuming the existence of and values for additional but *missing* (or *hidden*) parameters. The later application is more common in the computational pattern recognition community.

As before, we assume that data $\mathcal{X}$ is observed and is generated by some distribution. We call $\mathcal{X}$ the *incomplete data*. We assume that a complete data set exists $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ and also assume (or specify) a joint density function:

$$p(\mathbf{z}|\Theta) = p(\mathbf{x}, \mathbf{y}|\Theta) = p(\mathbf{y}|\mathbf{x}, \Theta)p(\mathbf{x}|\Theta)$$

.

Where does this joint density come from? Often it "arises" from the marginal density function $p(\mathbf{x}|\Theta)$ and the assumption of hidden variables and parameter value guesses (e.g., our two examples, Mixture-densities and Baum-Welch). In other cases (e.g., missing data values in samples of a distribution), we must assume a joint relationship between the missing and observed values.

With this new density function, we can define a new likelihood function, $\mathcal{L}(\Theta|\mathcal{Z}) = \mathcal{L}(\Theta|\mathcal{X},\mathcal{Y}) = p(\mathcal{X},\mathcal{Y}|\Theta)$, called the complete-data likelihood. Note that this function is in fact a random variable since the missing information $\mathcal{Y}$ is unknown, random, and presumably governed by an underlying distribution. That is, we can think of $\mathcal{L}(\Theta|\mathcal{X},\mathcal{Y}) = h_{\mathcal{X},\Theta}(\mathcal{Y})$ for some function $h_{\mathcal{X},\Theta}(\cdot)$ where $\mathcal{X}$ and Θ are constant and $\mathcal{Y}$ is a random variable. The original likelihood $\mathcal{L}(\Theta|\mathcal{X})$ is referred to as the incomplete-data likelihood function.

The EM algorithm first finds the expected value of the complete-data log-likelihood $\log p(\mathcal{X},\mathcal{Y}|\Theta)$ with respect to the unknown data $\mathcal{Y}$ given the observed data $\mathcal{X}$ and the current parameter estimates. That is, we define:

$$Q(\Theta,\Theta^{(i-1)}) = E\left[\log p(\mathcal{X},\mathcal{Y}|\Theta)|\mathcal{X},\Theta^{(i-1)}\right] \tag{1}$$

Where $\Theta^{(i-1)}$ are the current parameters estimates that we used to evaluate the expectation and Θ are the new parameters that we optimize to increase Q.

This expression probably requires some explanation. [1] The key thing to understand is that $\mathcal{X}$ and $\Theta^{(i-1)}$ are constants, Θ is a normal variable that we wish to adjust, and $\mathcal{Y}$ is a random variable governed by the distribution $f(\mathbf{y}|\mathcal{X},\Theta^{(i-1)})$. The right side of Equation 1 can therefore be re-written as:

$$E\left[\log p(\mathcal{X},\mathcal{Y}|\Theta)|\mathcal{X},\Theta^{(i-1)}\right] = \int_{\mathbf{y}\in\boldsymbol{\Upsilon}} \log p(\mathcal{X},\mathbf{y}|\Theta) f(\mathbf{y}|\mathcal{X},\Theta^{(i-1)})d\mathbf{y}. \tag{2}$$

Note that $f(\mathbf{y}|\mathcal{X},\Theta^{(i-1)})$ is the marginal distribution of the unobserved data and is dependent on both the observed data $\mathcal{X}$ and on the current parameters, and $\boldsymbol{\Upsilon}$ is the space of values $\mathbf{y}$ can take on. In the best of cases, this marginal distribution is a simple analytical expression of the assumed parameters $\Theta^{(i-1)}$ and perhaps the data. In the worst of cases, this density might be very hard to obtain. Sometimes, in fact, the density actually used is $f(\mathbf{y},\mathcal{X}|\Theta^{(i-1)}) = f(\mathbf{y}|\mathcal{X},\Theta^{(i-1)})f(\mathcal{X}|\Theta^{(i-1)})$ but this doesn't effect subsequent steps since the extra factor, $f_{\mathcal{X}}(\mathcal{X}|\Theta^{(i-1)})$ is not dependent on Θ.

As an analogy, suppose we have a function $h(\cdot,\cdot)$ of two variables. Consider $h(\theta,\mathbf{Y})$ where θ is a constant and $\mathbf{Y}$ is a random variable governed by some distribution $f_{\mathbf{Y}}(y)$. Then $q(\theta) = E_{\mathbf{Y}}[h(\theta,\mathbf{Y})] = \int_{\mathbf{y}} h(\theta,\mathbf{y})f_{\mathbf{Y}}(y)d\mathbf{y}$ is now a deterministic function that could be maximized if desired.

The evaluation of this expectation is called the E-step of the algorithm. Notice the meaning of the two arguments in the function $Q(\Theta,\Theta')$. The first argument Θ corresponds to the parameters that ultimately will be optimized in an attempt to maximize the likelihood. The second argument Θ' corresponds to the parameters that we use to evaluate the expectation.

The second step (the M-step) of the EM algorithm is to maximize the expectation we computed in the first step. That is, we find:

$$\Theta^{(i)} = \underset{\Theta}{\operatorname{argmax}}\; Q(\Theta,\Theta^{(i-1)}).$$

These two steps are repeated as necessary. Each iteration is guaranteed to increase the log-likelihood and the algorithm is guaranteed to converge to a local maximum of the likelihood function. There are many rate-of-convergence papers (e.g., [ALR77, RW84, Wu83, JX96, XJ96]) but we will not discuss them here.

[1] Recall that $E[h(Y)|X=x] = \int_y h(y)f_{Y|X}(y|x)dy$. In the following discussion, we drop the subscripts from different density functions since argument usage should should disambiguate different ones.

A modified form of the M-step is to, instead of maximizing $Q(\Theta, \Theta^{(i-1)})$, we find some $\Theta^{(i)}$ such that $Q(\Theta^{(i)}, \Theta^{(i-1)}) > Q(\Theta, \Theta^{(i-1)})$. This form of the algorithm is called Generalized EM (GEM) and is also guaranteed to converge.

As presented above, it's not clear how exactly to "code up" the algorithm. This is the way, however, that the algorithm is presented in its most general form. The details of the steps required to compute the given quantities are very dependent on the particular application so they are not discussed when the algorithm is presented in this abstract form.

3 Finding Maximum Likelihood Mixture Densities Parameters via EM

The mixture-density parameter estimation problem is probably one of the most widely used applications of the EM algorithm in the computational pattern recognition community. In this case, we assume the following probabilistic model:

$$p(\mathbf{x}|\Theta) = \sum_{i=1}^{M} \alpha_i p_i(\mathbf{x}|\theta_i)$$

where the parameters are $\Theta = (\alpha_1, \ldots, \alpha_M, \theta_1, \ldots, \theta_M)$ such that $\sum_{i=1}^{M} \alpha_i = 1$ and each p_i is a density function parameterized by θ_i. In other words, we assume we have M component densities mixed together with M mixing coefficients α_i.

The incomplete-data log-likelihood expression for this density from the data $\mathcal{X}$ is given by:

$$\log(\mathcal{L}(\Theta|\mathcal{X})) = \log \prod_{i=1}^{N} p(x_i|\Theta) = \sum_{i=1}^{N} \log \left(\sum_{j=1}^{M} \alpha_j p_j(x_i|\theta_j) \right)$$

which is difficult to optimize because it contains the log of the sum. If we consider $\mathcal{X}$ as incomplete, however, and posit the existence of unobserved data items $\mathcal{Y} = \{y_i\}_{i=1}^{N}$ whose values inform us which component density "generated" each data item, the likelihood expression is significantly simplified. That is, we assume that $y_i \in 1, \ldots, M$ for each i, and $y_i = k$ if the i^{th} sample was generated by the k^{th} mixture component. If we know the values of $\mathcal{Y}$, the likelihood becomes:

$$\log(\mathcal{L}(\Theta|\mathcal{X}, \mathcal{Y})) = \log(P(\mathcal{X}, \mathcal{Y}|\Theta)) = \sum_{i=1}^{N} \log\left(P(x_i|y_i)P(y)\right) = \sum_{i=1}^{N} \log\left(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i})\right)$$

which, given a particular form of the component densities, can be optimized using a variety of techniques.

The problem, of course, is that we do not know the values of $\mathcal{Y}$. If we assume $\mathcal{Y}$ is a random vector, however, we can proceed.

We first must derive an expression for the distribution of the unobserved data. Let's first guess at parameters for the mixture density, i.e., we guess that $\Theta^g = (\alpha_1^g, \ldots, \alpha_M^g, \theta_1^g, \ldots, \theta_M^g)$ are the appropriate parameters for the likelihood $\mathcal{L}(\Theta^g|\mathcal{X}, \mathcal{Y})$. Given Θ^g, we can easily compute $p_j(x_i|\theta_j^g)$ for each i and j. In addition, the mixing parameters, α_j can be though of as prior probabilities of each mixture component, that is $\alpha_j = p(\text{component j})$. Therefore, using Bayes's rule, we can compute:

$$p(y_i|x_i, \Theta^g) = \frac{\alpha_{y_i}^g p_{y_i}(x_i|\theta_{y_i}^g)}{p(x_i|\Theta^g)} = \frac{\alpha_{y_i}^g p_{y_i}(x_i|\theta_{y_i}^g)}{\sum_{k=1}^{M} \alpha_k^g p_k(x_i|\theta_k^g)}$$

and

$$p(\mathbf{y}|\mathcal{X},\Theta^g) = \prod_{i=1}^{N} p(y_i|x_i,\Theta^g)$$

where $\mathbf{y} = (y_1,\ldots,y_N)$ is an instance of the unobserved data independently drawn. When we now look at Equation 2, we see that in this case we have obtained the desired marginal density by assuming the existence of the hidden variables and making a guess at the initial parameters of their distribution.

In this case, Equation 1 takes the form:

$$\begin{aligned}
Q(\Theta,\Theta^g) &= \sum_{\mathbf{y}\in\Upsilon} \log\left(\mathcal{L}(\Theta|\mathcal{X},\mathbf{y})\right) p(\mathbf{y}|\mathcal{X},\Theta^g) \\
&= \sum_{\mathbf{y}\in\Upsilon} \sum_{i=1}^{N} \log\left(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i})\right) \prod_{j=1}^{N} p(y_j|x_j,\Theta^g) \\
&= \sum_{y_1=1}^{M} \sum_{y_2=1}^{M} \cdots \sum_{y_N=1}^{M} \sum_{i=1}^{N} \log\left(\alpha_{y_i} p_{y_i}(x_i|\theta_{y_i})\right) \prod_{j=1}^{N} p(y_j|x_j,\Theta^g) \\
&= \sum_{y_1=1}^{M} \sum_{y_2=1}^{M} \cdots \sum_{y_N=1}^{M} \sum_{i=1}^{N} \sum_{\ell=1}^{M} \delta_{\ell,y_i} \log\left(\alpha_\ell p_\ell(x_i|\theta_\ell)\right) \prod_{j=1}^{N} p(y_j|x_j,\Theta^g) \\
&= \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log\left(\alpha_\ell p_\ell(x_i|\theta_\ell)\right) \sum_{y_1=1}^{M} \sum_{y_2=1}^{M} \cdots \sum_{y_N=1}^{M} \delta_{\ell,y_i} \prod_{j=1}^{N} p(y_j|x_j,\Theta^g) \qquad (3)
\end{aligned}$$

In this form, $Q(\Theta,\Theta^g)$ looks fairly daunting, yet it can be greatly simplified. We first note that for $\ell \in 1,\ldots,M$,

$$\begin{aligned}
&\sum_{y_1=1}^{M} \sum_{y_2=1}^{M} \cdots \sum_{y_N=1}^{M} \delta_{\ell,y_i} \prod_{j=1}^{N} p(y_j|x_j,\Theta^g) \\
&\quad = \left(\sum_{y_1=1}^{M} \cdots \sum_{y_{i-1}=1}^{M} \sum_{y_{i+1}=1}^{M} \cdots \sum_{y_N=1}^{M} \prod_{j=1,j\neq i}^{N} p(y_j|x_j,\Theta^g) \right) p(\ell|x_i,\Theta^g) \\
&\quad = \prod_{j=1,j\neq i}^{N} \left(\sum_{y_j=1}^{M} p(y_j|x_j,\Theta^g) \right) p(\ell|x_i,\Theta^g) = p(\ell|x_i,\Theta^g) \qquad (4)
\end{aligned}$$

since $\sum_{i=1}^{M} p(i|x_j,\Theta^g) = 1$. Using Equation 4, we can write Equation 3 as:

$$\begin{aligned}
Q(\Theta,\Theta^g) &= \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log\left(\alpha_\ell p_\ell(x_i|\theta_\ell)\right) p(\ell|x_i,\Theta^g) \\
&= \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log(\alpha_\ell) p(\ell|x_i,\Theta^g) + \sum_{\ell=1}^{M} \sum_{i=1}^{N} \log(p_\ell(x_i|\theta_\ell)) p(\ell|x_i,\Theta^g) \qquad (5)
\end{aligned}$$

To maximize this expression, we can maximize the term containing α_ℓ and the term containing θ_ℓ independently since they are not related.

To find the expression for α_ℓ, we introduce the Lagrange multiplier λ with the constraint that $\sum_\ell \alpha_\ell = 1$, and solve the following equation:

$$\frac{\partial}{\partial \alpha_\ell}\left[\sum_{\ell=1}^{M}\sum_{i=1}^{N}\log(\alpha_\ell)p(\ell|x_i,\Theta^g) + \lambda\left(\sum_\ell \alpha_\ell - 1\right)\right] = 0$$

or

$$\sum_{i=1}^{N}\frac{1}{\alpha_\ell}p(\ell|x_i,\Theta^g) + \lambda = 0$$

Summing both sizes over ℓ, we get that $\lambda = -N$ resulting in:

$$\alpha_\ell = \frac{1}{N}\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)$$

For some distributions, it is possible to get an analytical expressions for θ_ℓ as functions of everything else. For example, if we assume d-dimensional Gaussian component distributions with mean μ and covariance matrix Σ, i.e., $\theta = (\mu, \Sigma)$ then

$$p_\ell(x|\mu_\ell,\Sigma_\ell) = \frac{1}{(2\pi)^{d/2}|\Sigma_\ell|^{1/2}}e^{-\frac{1}{2}(x-\mu_\ell)^T\Sigma_\ell^{-1}(x-\mu_\ell)}. \tag{6}$$

To derive the update equations for this distribution, we need to recall some results from matrix algebra.

The trace of a square matrix $\mathrm{tr}(A)$ is equal to the sum of A's diagonal elements. The trace of a scalar equals that scalar. Also, $\mathrm{tr}(A+B) = \mathrm{tr}(A) + \mathrm{tr}(B)$, and $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ which implies that $\sum_i x_i^T A x_i = \mathrm{tr}(AB)$ where $B = \sum_i x_i x_i^T$. Also note that $|A|$ indicates the determinant of a matrix, and that $|A^{-1}| = 1/|A|$.

We'll need to take derivatives of a function of a matrix $f(A)$ with respect to elements of that matrix. Therefore, we define $\frac{\partial f(A)}{\partial A}$ to be the matrix with i,j^{th} entry $[\frac{\partial f(A)}{\partial a_{i,j}}]$ where $a_{i,j}$ is the i,j^{th} entry of A. The definition also applies taking derivatives with respect to a vector. First, $\frac{\partial x^T A x}{\partial x} = (A + A^T)x$. Second, it can be shown that when A is a symmetric matrix:

$$\frac{\partial |A|}{\partial a_{i,j}} = \begin{cases} \mathcal{A}_{i,j} & \text{if } i = j \\ 2\mathcal{A}_{i,j} & \text{if } i \neq j \end{cases}$$

where $\mathcal{A}_{i,j}$ is the i,j^{th} cofactor of A. Given the above, we see that:

$$\frac{\partial \log|A|}{\partial A} = \left\{\begin{array}{ll} \mathcal{A}_{i,j}/|A| & \text{if } i = j \\ 2\mathcal{A}_{i,j}/|A| & \text{if } i \neq j \end{array}\right\} = 2A^{-1} - \mathrm{diag}(A^{-1})$$

by the definition of the inverse of a matrix. Finally, it can be shown that:

$$\frac{\partial \mathrm{tr}(AB)}{\partial A} = B + B^T - \mathrm{Diag}(B).$$

Taking the log of Equation 6, ignoring any constant terms (since they disappear after taking derivatives), and substituting into the right side of Equation 5, we get:

$$\begin{aligned}
&\sum_{\ell=1}^{M}\sum_{i=1}^{N}\log\left(p_\ell(x_i|\mu_\ell,\Sigma_\ell)\right)p(\ell|x_i,\Theta^g) \\
&= \sum_{\ell=1}^{M}\sum_{i=1}^{N}\left(-\frac{1}{2}\log(|\Sigma_\ell|)-\frac{1}{2}(x_i-\mu_\ell)^T\Sigma_\ell^{-1}(x_i-\mu_\ell)\right)p(\ell|x_i,\Theta^g) \qquad (7)
\end{aligned}$$

Taking the derivative of Equation 7 with respect to μ_ℓ and setting it equal to zero, we get:

$$\sum_{i=1}^{N}\Sigma_\ell^{-1}(x_i-\mu_\ell)p(\ell|x_i,\Theta^g)=0$$

with which we can easily solve for μ_ℓ to obtain:

$$\mu_\ell=\frac{\sum_{i=1}^{N}x_i p(\ell|x_i,\Theta^g)}{\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)}.$$

To find Σ_ℓ, note that we can write Equation 7 as:

$$\begin{aligned}
&\sum_{\ell=1}^{M}\left[\frac{1}{2}\log(|\Sigma_\ell^{-1}|)\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)-\frac{1}{2}\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)\mathrm{tr}\left(\Sigma_\ell^{-1}(x_i-\mu_\ell)(x_i-\mu_\ell)^T\right)\right] \\
&= \sum_{\ell=1}^{M}\left[\frac{1}{2}\log(|\Sigma_\ell^{-1}|)\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)-\frac{1}{2}\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)\mathrm{tr}\left(\Sigma_\ell^{-1}N_{\ell,i}\right)\right]
\end{aligned}$$

where $N_{\ell,i}=(x_i-\mu_\ell)(x_i-\mu_\ell)^T$.

Taking the derivative with respect to Σ_ℓ^{-1}, we get:

$$\begin{aligned}
&\frac{1}{2}\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)\left(2\Sigma_\ell-\mathrm{diag}(\Sigma_\ell)\right)-\frac{1}{2}\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)\left(2N_{\ell,i}-\mathrm{diag}(N_{\ell,i})\right) \\
&= \frac{1}{2}\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)\left(2M_{\ell,i}-\mathrm{diag}(M_{\ell,i})\right) \\
&= 2S-\mathrm{diag}(S)
\end{aligned}$$

where $M_{\ell,i}=\Sigma_\ell-N_{\ell,i}$ and where $S=\frac{1}{2}\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)M_{\ell,i}$. Setting the derivative to zero, i.e., $2S-\mathrm{diag}(S)=0$, implies that $S=0$. This gives

$$\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)\left(\Sigma_\ell-N_{\ell,i}\right)=0$$

or

$$\Sigma_\ell=\frac{\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)N_{\ell,i}}{\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)}=\frac{\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)(x_i-\mu_\ell)(x_i-\mu_\ell)^T}{\sum_{i=1}^{N}p(\ell|x_i,\Theta^g)}$$

Summarizing, the estimates of the new parameters in terms of the old parameters are as follows:

$$\alpha_\ell^{new} = \frac{1}{N}\sum_{i=1}^{N} p(\ell|x_i,\Theta^g)$$

$$\mu_\ell^{new} = \frac{\sum_{i=1}^{N} x_i p(\ell|x_i,\Theta^g)}{\sum_{i=1}^{N} p(\ell|x_i,\Theta^g)}$$

$$\Sigma_\ell^{new} = \frac{\sum_{i=1}^{N} p(\ell|x_i,\Theta^g)(x_i-\mu_\ell^{new})(x_i-\mu_\ell^{new})^T}{\sum_{i=1}^{N} p(\ell|x_i,\Theta^g)}$$

Note that the above equations perform both the expectation step and the maximization step simultaneously. The algorithm proceeds by using the newly derived parameters as the guess for the next iteration.

4 Learning the parameters of an HMM, EM, and the Baum-Welch algorithm

A Hidden Markov Model is a probabilistic model of the joint probability of a collection of random variables $\{O_1,\ldots,O_T,Q_1,\ldots,Q_T\}$. The O_t variables are either continuous or discrete observations and the Q_t variables are "hidden" and discrete. Under an HMM, there are two conditional independence assumptions made about these random variables that make associated algorithms tractable. These independence assumptions are 1), the t^{th} hidden variable, given the $(t-1)^{st}$ hidden variable, is independent of previous variables, or:

$$P(Q_t|Q_{t-1},O_{t-1},\ldots,Q_1,O_1) = P(Q_t|Q_{t-1}),$$

and 2), the t^{th} observation, given the t^{th} hidden variable, is independent of other variables, or:

$$P(O_t|Q_T,O_T,Q_{T-1},O_{T-1},\ldots,Q_{t+1},O_{t+1},Q_t,Q_{t-1},O_{t-1},\ldots,Q_1,O_1) = P(O_t|Q_t).$$

In this section, we derive the EM algorithm for finding the maximum-likelihood estimate of the parameters of a hidden Markov model given a set of observed feature vectors. This algorithm is also known as the Baum-Welch algorithm.

Q_t is a discrete random variable with N possible values $\{1\ldots N\}$. We further assume that the underlying "hidden" Markov chain defined by $P(Q_t|Q_{t-1})$ is time-homogeneous (i.e., is independent of the time t). Therefore, we can represent $P(Q_t|Q_{t-1})$ as a time-independent stochastic transition matrix $A = \{a_{i,j}\} = p(Q_t = j|Q_{t-1} = i)$. The special case of time $t = 1$ is described by the initial state distribution, $\pi_i = p(Q_1 = i)$. We say that we are in state j at time t if $Q_t = j$. A particular sequence of states is described by $q = (q_1,\ldots,q_T)$ where $q_t \in \{1\ldots N\}$ is the state at time t.

A particular observation sequence O is described as $O = (O_1 = o_1,\ldots,O_T = o_T)$. The probability of a particular observation vector at a particular time t for state j is described by: $b_j(o_t) = p(O_t = o_t|Q_t = j)$. The complete collection of parameters for all observation distributions is represented by $B = \{b_j(\cdot)\}$.

There are two forms of output distributions we will consider. The first is a discrete observation assumption where we assume that an observation is one of L possible observation symbols $o_t \in$

$V = \{v_1, \ldots, v_L\}$. In this case, if $o_t = v_k$, then $b_j(o_t) = p(O_t = v_k | q_t = j)$. The second form of probably distribution we consider is a mixture of M multivariate Gaussians for each state where $b_j(o_t) = \sum_{\ell=1}^{M} c_{j\ell} \mathcal{N}(o_t | \mu_{j\ell}, \Sigma_{j\ell}) = \sum_{\ell=1}^{M} c_{j\ell} b_{j\ell}(o_t)$.

We describe the complete set of HMM parameters for a given model by: $\lambda = (A, B, \pi)$. There are three basic problems associated with HMMs:

1. Find $p(O|\lambda)$ for some $O = (o1, \ldots, o_T)$. We use the forward (or the backward) procedure for this since it is much more efficient than direct evaluation.

2. Given some O and some λ, find the best state sequence $q = (q1, \ldots, q_T)$ that explains O. The Viterbi algorithm solves this problem but we won't discuss it in this paper.

3. Find $\lambda^* = \underset{\lambda}{\operatorname{argmax}}\ p(O|\lambda)$. The Baum-Welch (also called forward-backward or EM for HMMs) algorithm solves this problem, and we will develop it presently.

In subsequent sections, we will consider only the first and third problems. The second is addressed in [RJ93].

4.1 Efficient Calculation of Desired Quantities

One of the advantages of HMMs is that relatively efficient algorithms can be derived for the three problems mentioned above. Before we derive the EM algorithm directly using the Q function, we review these efficient procedures.
Recall the forward procedure. We define

$$\alpha_i(t) = p(O_1 = o_1, \ldots, O_t = o_t, Q_t = i|\lambda)$$

which is the probability of seeing the partial sequence $o_1, \ldots, o_t$ and ending up in state i at time t. We can efficiently define $\alpha_i(t)$ recursively as:

1. $\alpha_i(1) = \pi_i b_i(o_1)$

2. $\alpha_j(t+1) = \left[\sum_{i=1}^{N} \alpha_i(t) a_{ij}\right] b_j(o_{t+1})$

3. $p(O|\lambda) = \sum_{i=1}^{N} \alpha_i(T)$

The backward procedure is similar:

$$\beta_i(t) = p(O_{t+1} = o_{t+1}, \ldots, O_T = o_T | Q_t = i, \lambda)$$

which is the probability of the ending partial sequence $o_{t+1}, \ldots, o_T$ given that we started at state i at time t. We can efficiently define $\beta_i(t)$ as:

1. $\beta_i(T) = 1$

2. $\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_j(t+1)$

3. $p(O|\lambda) = \sum_{i=1}^{N} \beta_i(1) \pi_i b_i(o_1)$

We now define

$$\gamma_i(t) = p(Q_t = i|O, \lambda)$$

which is the probability of being in state i at time t for the state sequence O. Note that:

$$p(Q_t = i|O, \lambda) = \frac{p(O, Q_t = i|\lambda)}{P(O|\lambda)} = \frac{p(O, Q_t = i|\lambda)}{\sum_{j=1}^{N} p(O, Q_t = j|\lambda)}$$

Also note that because of Markovian conditional independence

$$\alpha_i(t)\beta_i(t) = p(o_1, \ldots, o_t, Q_t = i|\lambda)p(o_{t+1}, \ldots, o_T|Q_t = i, \lambda) = p(O, Q_t = i|\lambda)$$

so we can define things in terms of $\alpha_i(t)$ and $\beta_i(t)$ as

$$\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{N} \alpha_j(t)\beta_j(t)}$$

We also define

$$\xi_{ij}(t) = p(Q_t = i, Q_{t+1} = j|O, \lambda)$$

which is the probability of being in state i at time t and being in state j at time $t+1$. This can also be expanded as:

$$\xi_{ij}(t) = \frac{p(Q_t = i, Q_{t+1} = j, O|\lambda)}{p(O|\lambda)} = \frac{\alpha_i(t)a_{ij}b_j(o_{t+1})\beta_j(t+1)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i(t)a_{ij}b_j(o_{t+1})\beta_j(t+1)}$$

or as:

$$\xi_{ij}(t) = \frac{p(Q_t = i|O)p(o_{t+1}\ldots o_T, Q_{t+1} = j|Q_t = i, \lambda)}{p(o_{t+1}\ldots o_T|Q_t = i, \lambda)} = \frac{\gamma_i(t)a_{ij}b_j(o_{t+1})\beta_j(t+1)}{\beta_i(t)}$$

If we sum these quantities across time, we can get some useful values. I.e., the expression

$$\sum_{t=1}^{T} \gamma_i(t)$$

is the expected number of times in state i and therefore is the expected number of transitions away from state i for O. Similarly,

$$\sum_{t=1}^{T-1} \xi_{ij}(t)$$

is the expected number of transitions from state i to state j for O. These follow from the fact that

$$\sum_t \gamma_i(t) = \sum_t E[I_t(i)] = E[\sum_t I_t(i)]$$

and

$$\sum_t \xi_{ij}(t) == \sum_t E[I_t(i,j)] = E[\sum_t I_t(i,j)]$$

where $I_t(i)$ is an indicator random variable that is 1 when we are in state i at time t, and $I_t(i,j)$ is a random variable that is 1 when we move from state i to state j after time t.

Jumping the gun a bit, our goal in forming an EM algorithm to estimate new parameters for the HMM by using the old parameters and the data. Intuitively, we can do this simply using relative frequencies. I.e., we can define update rules as follows:

The quantity

$$\tilde{\pi}_i = \gamma_i(1) \tag{8}$$

is the expected relative frequency spent in state i at time 1.

The quantity

$$\tilde{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} \tag{9}$$

is the expected number of transitions from state i to state j relative to the expected total number of transitions away from state i.

And, for discrete distributions, the quantity

$$\tilde{b}_i(k) = \frac{\sum_{t=1}^{T} \delta_{o_t,v_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)} \tag{10}$$

is the expected number of times the output observations have been equal to v_k while in state i relative to the expected total number of times in state i.

For Gaussian mixtures, we define the probability that the ℓ^{th} component of the i^{th} mixture generated observation o_t as

$$\gamma_{i\ell}(t) = \gamma_i(t) \frac{c_{i\ell} b_{i\ell}(o_t)}{b_i(o_t)} = p(Q_t = i, X_{it} = \ell | O, \lambda)$$

where X_{it} is a random variable indicating the mixture component at time t for state i.

From the previous section on Gaussian Mixtures, we might guess that the update equations for this case are:

$$c_{i\ell} = \frac{\sum_{t=1}^{T} \gamma_{i\ell}(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

$$\mu_{i\ell} = \frac{\sum_{t=1}^{T} \gamma_{i\ell}(t) o_t}{\sum_{t=1}^{T} \gamma_{i\ell}(t)}$$

$$\Sigma_{i\ell} = \frac{\sum_{t=1}^{T} \gamma_{i\ell}(t)(o_t - \mu_{i\ell})(o_t - \mu_{i\ell})^T}{\sum_{t=1}^{T} \gamma_{i\ell}(t)}$$

When there are E observation sequences the e^{th} being of length T_e, the update equations become:

$$\pi_i = \frac{\sum_{e=1}^{E} \gamma_i^e(1)}{E}$$

$$c_{i\ell} = \frac{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{i\ell}^e(t)}{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_i^e(t)}$$

$$\mu_{i\ell} = \frac{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{i\ell}^e(t) o_t^e}{\sum_{e=1}^{E} \sum_{t=1}^{T_e} \gamma_{i\ell}^e(t)}$$

$$\Sigma_{i\ell} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{i\ell}^{e}(t)(o_t^e-\mu_{i\ell})(o_t^e-\mu_{i\ell})^T}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{i\ell}^{e}(t)}$$

and

$$a_{ij} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\xi_{ij}^{e}(t)}{\sum_{e=1}^{E}\sum_{t=1}^{T_e}\gamma_{i}^{e}(t)}$$

These relatively intuitive equations are in fact the EM algorithm (or Balm-Welch) for HMM parameter estimation. We derive these using the more typical EM notation in the next section.

4.2 Estimation formula using the Q function.

We consider $O = (o_1, \ldots, o_T)$ to be the observed data and the underlying state sequence $q = (q_1, \ldots, q_T)$ to be hidden or unobserved. The incomplete-data likelihood function is given by $P(O|\lambda)$ whereas the complete-data likelihood function is $P(O, q|\lambda)$. The Q function therefore is:

$$Q(\lambda, \lambda') = \sum_{q\in\mathcal{Q}} \log P(O, q|\lambda) P(O, q|\lambda')$$

where λ' are our initial (or guessed, previous, etc.)[2] estimates of the parameters and where $\mathcal{Q}$ is the space of all state sequences of length T.

Given a particular state sequence q, representing $P(O, q|\lambda')$ is quite easy.[3] I.e.,

$$P(O, q|\lambda) = \pi_{q_0}\prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t)$$

The Q function then becomes:

$$Q(\lambda, \lambda') = \sum_{q\in\mathcal{Q}} \log \pi_{q_0} P(O, q|\lambda') + \sum_{q\in\mathcal{Q}} \left(\sum_{t=1}^{T} \log a_{q_{t-1}q_t}\right) p(O, q|\lambda') + \sum_{q\in\mathcal{Q}} \left(\sum_{t+1}^{T} \log b_{q_t}(o_t)\right) P(O, q|\lambda') \tag{11}$$

Since the parameters we wish to optimize are now independently split into the three terms in the sum, we can optimize each term individually.

The first term in Equation 11 becomes

$$\sum_{q\in\mathcal{Q}} \log \pi_{q_0} P(O, q|\lambda') = \sum_{i=1}^{N} \log \pi_i p(O, q_0 = i|\lambda')$$

since by selecting all $q \in \mathcal{Q}$, we are simply repeatedly selecting the values of q_0, so the right hand side is just the marginal expression for time $t = 0$. Adding the Lagrange multiplier γ, using the constraint that $\sum_i \pi_i = 1$, and setting the derivative equal to zero, we get:

$$\frac{\partial}{\partial \pi_i}\left(\sum_{i=1}^{N} \log \pi_i p(O, q_0 = i|\lambda') + \gamma(\sum_{i=1}^{N} \pi_i - 1)\right) = 0$$

[2]For the remainder of the discussion any *primed* parameters are assumed to be the initial, guessed, or previous parameters whereas the unprimed parameters are being optimized.

[3]Note here that we assume the initial distribution starts at $t = 0$ instead of $t = 1$ for notational convenience. The basic results are the same however.

Taking the derivative, summing over i to get γ, and solving for π_i, we get:

$$\pi_i = \frac{P(O, q_0 = i|\lambda')}{P(O|\lambda')}$$

The second term in Equation 11 becomes:

$$\sum_{q \in \mathcal{Q}} \left(\sum_{t=1}^{T} \log a_{q_{t-1} q_t} \right) p(O, q|\lambda') = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T} \log a_{ij} P(O, q_{t-1} = i, q_t = j|\lambda')$$

because for this term, we are, for each time t looking over all transitions from i to j and weighting that by the corresponding probability – the right hand side is just sum of the joint-marginal for time $t-1$ and t. In a similar way, we can use a Lagrange multiplier with the constraint $\sum_{j=1}^{N} a_{ij} = 1$ to get:

$$a_{ij} = \frac{\sum_{t=1}^{T} P(O, q_{t-1} = i, q_t = j|\lambda')}{\sum_{t=1}^{T} P(O, q_{t-1} = i|\lambda')}$$

The third term in Equation 11 becomes:

$$\sum_{q \in \mathcal{Q}} \left(\sum_{t=1}^{T} \log b_{q_t}(o_t) \right) P(O, q|\lambda') = \sum_{i=1}^{N} \sum_{t=1}^{T} \log b_i(o_t) p(O, q_t = i|\lambda')$$

because for this term, we are, for each time t, looking at the emissions for all states and weighting each possible emission by the corresponding probability – the right hand side is just the sum of the marginal for time t.

For discrete distributions, we can, again, use use a Lagrange multiplier but this time with the constraint $\sum_{j=1}^{L} b_i(j) = 1$. Only the observations that are equal to v_k contribute to the k^{th} probability value, so we get:

$$b_i(k) = \frac{\sum_{t=1}^{T} P(O, q_t = i|\lambda') \delta_{o_t, v_k}}{\sum_{t=1}^{T} P(O, q_t = i|\lambda')}$$

For Gaussian Mixtures, the form of the Q function is slightly different, i.e., the hidden variables must include not only the hidden state sequence, but also a variable indicating the mixture component for each state at each time. Therefore, we can write Q as:

$$Q(\lambda, \lambda') = \sum_{q \in \mathcal{Q}} \sum_{m \in \mathcal{M}} \log P(O, q, m|\lambda) P(O, q, m|\lambda')$$

where m is the vector $m = \{m_{q_1 1}, m_{q_2 2}, \ldots, m_{q_T T}\}$ that indicates the mixture component for each state at each time. If we expand this as in Equation 11, the first and second terms are unchanged because the parameters are independent of m which is thus marginalized away by the sum. The third term in Equation 11 becomes:

$$\sum_{q \in \mathcal{Q}} \sum_{m \in \mathcal{M}} \left(\sum_{t=1}^{T} \log b_{q_t}(o_t, m_{q_t t}) \right) P(O, q, m|\lambda') = \sum_{i=1}^{N} \sum_{\ell=1}^{M} \sum_{t=1}^{T} \log(c_{i\ell} b_{i\ell}(o_t)) p(O, q_t = i, m_{q_t t} = \ell|\lambda')$$

This equation is almost identical to Equation 5, except for an addition sum component over the hidden state variables. We can optimize this in an exactly analogous way as we did in Section 3, and we get:

$$c_{il} = \frac{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = \ell|O, \lambda')}{\sum_{t=1}^{T} \sum_{\ell=1}^{M} P(q_t = i, m_{q_t t} = \ell|O, \lambda')},$$

$$\mu_{il} = \frac{\sum_{t=1}^{T} o_t P(q_t = i, m_{q_t t} = \ell | O, \lambda')}{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = \ell | O, \lambda')},$$

and

$$\Sigma_{il} = \frac{\sum_{t=1}^{T} (o_t - \mu_{i\ell})(o_t - \mu_{i\ell})^T P(q_t = i, m_{q_t t} = \ell | O, \lambda')}{\sum_{t=1}^{T} P(q_t = i, m_{q_t t} = \ell | O, \lambda')}.$$

As can be seen, these are the same set of update equations as given in the previous section.

The update equations for HMMs with multiple observation sequences can similarly be derived and are addressed in [RJ93].

References

[ALR77] A.P.Dempster, N.M. Laird, and D.B. Rubin. Maximum-likelihood from incomplete data via the em algorithm. *J. Royal Statist. Soc. Ser. B.*, 39, 1977.

[Bis95] C. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[GJ95] Z. Ghahramami and M. Jordan. Learning from incomplete data. Technical Report AI Lab Memo No. 1509, CBCL Paper No. 108, MIT AI Lab, August 1995.

[JJ94] M. Jordan and R. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.

[JX96] M. Jordon and L. Xu. Convergence results for the em approach to mixtures of experts architectures. *Neural Networks*, 8:1409–1431, 1996.

[RJ93] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series, 1993.

[RW84] R. Redner and H. Walker. Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2), 1984.

[Wu83] C.F.J. Wu. On the convergence properties of the em algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

[XJ96] L. Xu and M.I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8:129–151, 1996.