Ivo D. Dinov
http://www.stat.ucla.edu/~dinov
UCLA Statistics
Courses & Students

Generalized Expectation Maximization [1]

1 Maximum Likelihood Estimation (MLE)

First let's recall the definition of the maximum-likelihood estimation problem. We have a density function $f(x \mid \Theta)$, that is governed by the set of parameters Θ (e.g., f might be a Gaussian and Θ could be the means (vecrtor) and covariance (matrix)). We also have a data set of size N, supposedly drawn from this distribution with density f, i.e., $\mathcal{X} = \{x_1, \ldots, x_N\}$. That is, we assume that these data vectors are independent and identically distributed (IID) with distribution f. Therefore, the resulting joint density for the samples is:

$$p(\mathcal{X} \mid \Theta) = \prod_{i=1}^{N} p(x_i \mid \Theta) = L(\Theta \mid \mathcal{X})$$

$L(\Theta \mid \mathcal{X})$ is called the **likelihood function** of the parameters (Θ) given the data, or just the likelihood. The likelihood is thought of as a function of the parameters (Θ) where the data $\mathcal{X}$ is fixed (observed). In the **maximum likelihood problem**, our goal is to find a parameter vector Θ that maximizes $L(\Theta \mid \mathcal{X})$. In other words, we look for Θ^*, where

$$\Theta^* = \underset{\Theta}{\text{ArgMax}}\, L(\Theta \mid X)$$

Oftentimes we choose to maximize $Log(L(\Theta \mid \mathcal{X}))$ instead because it is analytically easier or computationally appealing.

Depending on the form of $f(x \mid \Theta)$ this problem can be easy or hard. For example, if $f(x \mid \Theta)$ is simply a single Gaussian distribution where the parameter vector $\Theta=(\mu, \sigma^2)$, then we can solve the maximum likelihood problem of determining estimates (MLE) of μ & σ^2 by setting the partial derivatives of $Log(L(\Theta \mid \mathcal{X}))$ to zero (in fact, this is how the familiar formulas for the population mean and variance are obtained). For many problems, however, it is not possible to find such analytical expressions, and we must resort to more elaborate techniques (e.g., EM technique).

[1] This set of class notes is based on previous work by Bilmer, Bishop, Dempster, Ghahramami, Jordon, Rabiner, Redner, Wu and Xu see references at the end of the document.

Example 1: Suppose $\{X_1, \ldots, X_n\}$ IID $N(\mu, \sigma^2)$, where μ is unknown. We estimate μ by:

$$\text{MLE}(\mu) = \hat{\mu} = \underset{\mu}{\text{ArgMax}}\, L(\mu \,|\, (\{X_1, \ldots, X_N\}).$$

$$MLE(\mu) = Log\left(\prod_{i=1}^{n} \frac{e^{-(x_i-\mu)^2 / 2\sigma^2}}{\sqrt{2\pi\sigma^2}} \right) = L(\mu)$$

$$0 = L'(\hat{\mu}) = \frac{1}{\left(2\pi\sigma^2\right)^{n/2}} \left(e^{-\sum_{i=1}^{n}(x_i-\hat{\mu})^2 / 2\sigma^2} \right) \frac{\sum_{i=1}^{n} 2(x_i - \hat{\mu})}{2\sigma^2}$$

$$\Leftrightarrow 0 = 2\sum_{i=1}^{n}(x_i - \hat{\mu}) \Leftrightarrow \hat{\mu} = \sum_{i=1}^{n} x_i / n .$$

Similarly can show that : $MLE(\sigma) = \hat{\sigma} = \sum_{i=1}^{n}\left(x_i - \mu\right)^2 / (n-1)$.

Example 2: Suppose $\{X_1, \ldots, X_N\}$ IID Poisson(λ), where (the population mean, and standard deviation) λ is unknown. Estimate λ by:

$$\text{MLE}(\lambda) = \hat{\lambda} = \underset{\lambda}{\text{ArgMax}}\, L(\lambda \,|\, (\{X_1, \ldots, X_N\})$$

$$MLE(\lambda) = Log\left(\prod_{i=1}^{n} \frac{e^{-\lambda} \lambda^{x_i}}{(x_i)!} \right) = L(\lambda)$$

$$0 = L'(\hat{\lambda}) = \frac{\partial}{\partial\lambda} Log\left(\frac{e^{-n\lambda} \lambda^{\sum_{i=1}^{n} x_i}}{\prod_{i=1}^{n}(x_i)!} \right) =$$

$$= \frac{\partial}{\partial\lambda}\left(-n\lambda + Log(\lambda)\sum_{i=1}^{n} x_i \right) = -n + \frac{1}{\lambda}\sum_{i=1}^{n} x_i \Leftrightarrow \hat{\lambda} = \sum_{i=1}^{n} x_i / n .$$

2 General Expectation Maximization (GEM) Algorithms

The EM algorithm is one such technique, which allows estimating parameter vectors in the cases when such analytic solutions to the likelihood minimization are difficult or impossible. The EM algorithm [*see references at the end*] is a general method of finding the maximum-likelihood

estimates 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.

Application 1: The first application of EM algorithm occurs when the data indeed has missing values, due to problems with or limitations of the observation process.

Application 2: The second occurs when optimizing the likelihood function is analytically intractable, however, we still need to assume the likelihood function can be simplified by assuming the existence of and values for additional but *missing* (or *hidden*) parameters. This second application is more common in the computational pattern recognition community.

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

$$p(\boldsymbol{Z} \mid \Theta) = p(\boldsymbol{X},\boldsymbol{Y} \mid \Theta) = p(\boldsymbol{Y} \mid \boldsymbol{X},\Theta)\, p(\boldsymbol{X} \mid \Theta)$$

Recall that for each probability measure, P, $P(A,B)=P(A|B)P(B)$ and hence, in terms of the conditional probability, $P(A,B|C) = P(A|B,C)P(B|C)$. This joint density often times arises from the marginal density function $p(\boldsymbol{X} \mid \Theta)$ and the assumption of hidden variables and parameter value guesses (e.g., mixture-densities, this example is coming up; and Baum-Welch algorithm). 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,

$$L(\Theta \mid \boldsymbol{Z}) = L(\Theta \mid \boldsymbol{X},\boldsymbol{Y}) = p(\boldsymbol{X},\boldsymbol{Y} \mid \Theta),$$

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

The Expectation Maximization algorithm then proceeds in two steps – expectation followed by its maximization.

Step 1 of EM (Expectation): The EM algorithm needs to first find the expected value of the complete-data log-likelihood $Log\left(p(\boldsymbol{X},\boldsymbol{Y} \mid \Theta)\right)$ with respect to the unknown data $\boldsymbol{Y}$ given the observed data $\boldsymbol{X}$ and the current parameter estimates Θ^{i}, the ***E-Step***. Below we define the expectation of $\Theta^{(i-1)}$, the second argument represents the parameters that we use to evaluate the expectation. The first argument Θ simply indicates the parameter (vector) that ultimately will be optimized in an attempt to maximize the likelihood:

$$Q(\Theta, \Theta^{(i-1)}) = E[Log\, f(\boldsymbol{X},\boldsymbol{Y} \mid \Theta) \mid \boldsymbol{X}, \Theta^{(i-1)}]. \qquad (1)$$

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 (maximize) Q. Note that the expression above is a *conditional expectation* w.r.t. ($\boldsymbol{X}$ & $\Theta^{(i-1)}$), i.e.,

$$E[\,h(Y) \mid X=x] := E[\,h(Y) \mid X = x] := \int_y h(y) \times f_{Y|X}(y \mid x)dy\,.$$

The key thing to understand is that $\boldsymbol{X}$ and $\Theta^{(i-1)}$ are *given* constants, Θ is a random variable that we wish to adjust/estimate, and $\boldsymbol{Y}$ is a random variable governed by the distribution $f(\boldsymbol{Y} \mid X, \Theta^{(i-1)})$. The right side of equation (1) can therefore be expressed as:

$$E[Log\ \ p(X,Y \mid \Theta) \mid X, \Theta^{(i-1)}] = \int_{y \in \boldsymbol{\mathcal{Y}}} Log\ p(X,y \mid \Theta) f(y \mid X, \Theta^{(i-1)})dy \quad (2)$$

The $f(\boldsymbol{Y} \mid \boldsymbol{X}, \Theta^{(i-1)})$ is the **marginal** distribution of the unobserved data ($\boldsymbol{Y}$) and is dependent on: the observed data $\boldsymbol{X}$, on the current parameters $\Theta^{(i-1)}$, and $\boldsymbol{\mathcal{Y}}$ is the space of values $\boldsymbol{y}$ can take on. In the best-case situations, this marginal distribution is a simple analytical expression of the assumed parameters $\Theta^{(i-1)}$, and perhaps the observed data ($\boldsymbol{X}$). In the worst-case scenario, this density might be very hard to obtain. In fact, sometimes the actually used density is:

$$f(\boldsymbol{y}, \boldsymbol{X} \mid \Theta^{(i-1)}) = f(\boldsymbol{y} \mid \boldsymbol{X}, \Theta^{(i-1)})\, f(\boldsymbol{X} \mid \Theta^{(i-1)}),$$

but this doesn't effect subsequent steps since the extra factor, $f(\boldsymbol{X} \mid \Theta^{(i-1)})$ is not dependent on Θ.

As an analogy, suppose we have a function $h(\,.\,;\,.\,)$ of two variables. Consider $h(\,\theta\,;\,\boldsymbol{Y}\,)$ where θ is a constant and $\boldsymbol{Y}$ is a random variable governed by some distribution $f_{\boldsymbol{Y}}(y)$. Then

$$q(\theta) = E_{\boldsymbol{Y}}[h(\theta; Y)] = \int_{\mathrm{Y}} h(\theta; Y) f_Y(y)dy$$

is now a deterministic function that could be maximized if desired, w.r.t. θ. The evaluation of this expectation is called the ***E-step*** of the algorithm. Notice the meaning of the two arguments in the function $Q(\Theta, \Theta^{(i-1)})$. The first argument Θ corresponds to the parameters that ultimately will be optimized in an attempt to maximize the likelihood. The second argument, $\Theta^{(i-1)}$, corresponds to the parameters that we use to evaluate the expectation at each iteration (E→M → E→M → E→M → …).

Step 2 of EM (Maximization): The second step (the ***M-step***) of the EM algorithm is to maximize the expectation we computed in the first step. That is, we iteratively compute:

$$\Theta^{(i)} = \underset{\Theta}{ArgMax}\left(Q\left(\Theta, \Theta^{(i-1)}\right)\right)$$

That is we *maximize the expectation of the log-likelihood function*. 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 theoretical and empirical rate-of-convergence papers (see references below).

A modified form of the M-step is to, instead of maximizing the rather difficult function $Q(\Theta, \Theta^{(i-1)})$, 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**. This description of the GEM does not yield a direct computer implementation scheme (it's not constructive (the coding algorithm is not explicit). 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, but rather detailed for each specific application.

3 EM Application: 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 (mixture-density) model:

$$p(x \mid \Theta) = \sum_{i=1}^{M} a_i p_i(x \mid \Theta) \quad M = \#\,\text{Mixture Distributions} \qquad (3)$$

where the parameter vector is $\Theta=(\alpha_1, \ldots, \alpha_M ; \theta_1, \ldots, \theta_M)$, where the mixture-model weights satisfy $\sum_{i=1}^{M} \alpha_i = 1$ and each $\boldsymbol{p_i}$ is a density function parameterized, in general, by its own parameter vector 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 $\boldsymbol{X=\{x_1, \ldots, x_N\}}$, $\boldsymbol{N}$ = # Observations, is given by:

$$Log\left(L(\Theta \mid \boldsymbol{X})\right) = Log\left[\prod_{i=1}^{N} p(x_i \mid \Theta)\right] = \sum_{i=1}^{N} Log\left[\sum_{j=1}^{M} a_j p_j(x \mid \theta_j)\right]$$

which is difficult to optimize because it contains the logarithm function of the sum [if the sum and the log were interchanged then optimizing the outside logarithm would have been equivalent to optimizing its argument – the sum – as the log function has always a positive derivative over its domain $(0; \infty)$]. If we consider X as incomplete, however, and assume the existence of unobserved data items $Y=\{y_i\}_{i=1}^{N}$, whose values inform us which component density "generated" each data item, the likelihood expression is significantly simplified. That is, if we assume that $y_i \in \{1,2,...,M\}$ for each $1 \le i \le N$, and $y_i=k$ if the i^{th} sample, x_i, was generated by the k^{th} mixture component p_k. If we know the values of $\boldsymbol{Y}$, the likelihood becomes:

$$Log\left(L(\Theta \mid \boldsymbol{X,Y})\right) = Log\left[P(\boldsymbol{X,Y} \mid \Theta)\right] = \sum_{i=1}^{N} Log\left[P(x_i \mid y_i)P(y_i)\right] = \sum_{i=1}^{N} Log\left[\alpha_{y_i} p_{y_i}(x_i \mid \theta_{y_i})\right]$$

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 $\boldsymbol{Y}$. If we assume $\boldsymbol{Y}$ is a random vector, however, we can proceed. We first must derive an expression for the distribution of the unobserved (missing) data, $\boldsymbol{Y}$. 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 $L(\Theta^g \mid \boldsymbol{X,Y})$. Given Θ^g, we can compute $p_j(x_i \mid \theta_j^g)$ for each i and j. In addition, the mixing parameters, α_i, can be though of as *prior* probabilities of each mixture component, that is $\alpha_i = p$(component i). Therefore, using Bayes's rule – $P(Y_i \mid X_i)=P(X_i \mid Y_i)P(Y_i)/P(X_i)$, we can compute:

$$p(y_i \mid x_i, \Theta^g) = \frac{\alpha_{y_i}^g \, p_{y_i}\left(x_i \mid \theta_{y_i}^g\right)}{p(x_i \mid \Theta^g)} = \frac{\alpha_{y_i}^g \, p_{y_i}\left(x_i \mid \theta_{y_i}^g\right)}{\Sigma_{k=1}^M \alpha_k^g \, p_k(x_i \mid \Theta_k^g)}$$

and therefore, $p(y \mid X, \Theta^g) = \prod_{i=1}^N p(y_i \mid x_i, \Theta^g)$, where y = ($y_1$, …, 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 (of $\boldsymbol{Y}$), $f(y \mid \boldsymbol{X}, \Theta^{(i-1)})$, 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 specific form:

$$\begin{aligned}
Q\left(\Theta, \Theta^{(g)}\right) &= \Sigma_{y \in \boldsymbol{Y}} \log\left(L(\Theta \mid X, y)\right) p(y \mid X, \Theta^g) \\
&= \Sigma_{y \in Y} \Sigma_{i=1}^N Log\left(\alpha_{y_i} \; p_{y_i}(x_i \mid \theta_{y_i})\right) \prod_{j=1}^N p(y_j \mid x_j, \Theta^g) \\
&= \Sigma_{y_1=1}^M \Sigma_{y_2=1}^M \ldots \Sigma_{y_N=1}^M \Sigma_{i=1}^N Log\left(\alpha_{y_i} \; p_{y_i}(x_i \mid \theta_{y_i})\right) \prod_{j=1}^N p(y_j \mid x_j, \Theta^g) \\
&= \Sigma_{y_1=1}^M \Sigma_{y_2=1}^M \ldots \Sigma_{y_N=1}^M \Sigma_{i=1}^N \Sigma_{l=1}^M \delta_{l,y_i} \, Log\left(\alpha_l \; p_l(x_i \mid \theta_l)\right) \prod_{j=1}^N p(y_j \mid x_j, \Theta^g) \\
&= \Sigma_{l=1}^M \Sigma_{i=1}^N Log\left(\alpha_l \; p_l(x_i \mid \theta_l)\right) \Sigma_{y_1=1}^M \Sigma_{y_2=1}^M \ldots \Sigma_{y_N=1}^M \delta_{l,y_i} \prod_{j=1}^N p(y_j \mid x_j, \Theta^g) \quad (4)
\end{aligned}$$

where $\delta_{l,y_i} = \begin{cases} 1, \text{if } l = y_i \\ 0, \text{otherwise} \end{cases}$. In this form, $Q\left(\Theta, \Theta^{(g)}\right)$ appears computationally challenging, however, it can be simplified, since for $l \in \{1,2,\ldots,M\}$,

$$\begin{aligned}
&\Sigma_{y_1=1}^M \Sigma_{y_2=1}^M \ldots \Sigma_{y_N=1}^M \delta_{l,y_i} \prod_{j=1}^N p(y_j \mid x_j, \Theta^g) = \\
&= \left(\Sigma_{y_1=1}^M \ldots \Sigma_{y_{i-1}=1}^M \Sigma_{y_i+1=1}^M \ldots \Sigma_{y_N=1}^M \prod_{j=1, j \neq i}^N p(y_j \mid x_j, \Theta^g) \right) p(l \mid x_i, \Theta^g) \\
&= \prod_{j=1, j \neq i}^N \left[\Sigma_{y_j=1}^M p(y_j \mid x_j, \Theta^g) \right] p(l \mid x_i, \Theta^g) = p(l \mid x_i, \Theta^g) \quad (5)
\end{aligned}$$

This is because $\Sigma_{l=1}^M p(i \mid x_j, \Theta^g) = 1$. Using equation (5), we can write equation (4) as:

$$Q\left(\Theta,\Theta^{(g)}\right)=\Sigma_{l=1}^{M}\Sigma_{i=1}^{N}Log\left(a_l p_l(x_i\mid\theta_l)\right)p(l\mid x_i,\Theta^g)$$
$$=\Sigma_{l=1}^{M}\Sigma_{i=1}^{N}Log\left(a_l\right)Log\left(p_l(x_i\mid\Theta^g)\right)+\Sigma_{l=1}^{M}\Sigma_{i=1}^{N}Log\left(p_l(x_i\mid\theta_l)\right)p(l\mid x_i,\Theta^g) \quad (6)$$
$$=\Phi(\{\alpha_l\})+\Psi(\{\theta_l\}).$$

Now, to maximize this expression, we can maximize independently Φ and Ψ, the terms containing α_l and θ_l, since they are not related. To find the expression for α_l, which maximizes $Q(\Theta,\Theta^{(g)})$, we introduce the Lagrange multiplier λ with the constraint that $g=\Sigma_l\alpha_l-1=0$, and solve the following equation:

$$\frac{\partial}{\partial\alpha_l}\left[\Sigma_{l=1}^{M}\Sigma_{i=1}^{N}Log\left(\alpha_l\right)Log\left(p_l(x_i\mid\Theta^g)\right)+\lambda\left(\Sigma_{l=1}^{M}\alpha_l-1\right)\right]=0$$

This yields: $\Sigma_{i=1}^{N}\frac{1}{\alpha_l}p_l(x_i\mid\Theta^g)+\lambda=0$

Summing over l we get:

$$\Sigma_{l=1}^{M}\left[\Sigma_{i=1}^{N}\frac{1}{\alpha_l}p_l(x_i\mid\Theta^g)+\lambda\right]=\Sigma_{i=1}^{N}\Sigma_{l=1}^{M}\frac{1}{\alpha_l}p_l(x_i\mid\Theta^g)+M\lambda=0$$

Therefore, $\lambda=-N$, which yields that $\alpha_l=\frac{1}{N}\Sigma_{i=1}^{N}p_l(x_i\mid\Theta^g)$. This is how we determine the mixture parameters $\{\alpha_l\}$, most of the time.

Now let us try to estimate the second part of the parameter vector $\Theta=(\alpha_1,\ldots,\alpha_M;\theta_1,\ldots,\theta_M)$, i.e., the distribution specific parameters $(\theta_1,\ldots,\theta_M)$. Clearly, these need to be estimated in a case by case manner, as different distributions have different number and type of parameters. We consider again a couple of cases that illustrate the basic strategy for estimating $(\theta_1,\ldots,\theta_M)$ using EM approaches. For some distributions, it will be possible to get analytic expressions for θ_l directly, as functions of all other variables.

Example 1: Suppose that the mixture model in equation (3) involves Poisson(λ_l) distributions, $1\le l\le M$. Then the

$$Q\left(\Theta,\Theta^{(g)}\right)=\Sigma_{l=1}^{M}\Sigma_{i=1}^{N}Log\left(p_l(x_i\mid\mu_l,\Sigma_l)\right)p(l\mid x_i,\Theta^g)$$
$$=\Sigma_{l=1}^{M}\Sigma_{i=1}^{N}\left[-\lambda_l+x_i Log(\lambda_l)-Log\left((x_i)!\right)\right]p(l\mid x_i,\Theta^g)$$

Taking the derivatives w.r.t. λ_l and setting these equal to zero yields,

$$0 = \Sigma_{i=1}^{N}\left[-1+\frac{x_i}{\lambda_l}\right]p\left(l \mid x_i, \Theta^g\right) \Rightarrow -N + \frac{1}{\lambda_l}\Sigma_{i=1}^{N} x_i\, p\left(l \mid x_i, \Theta^g\right) = 0$$

$$\lambda_l = \frac{\Sigma_{i=1}^{N} x_i p\left(l \mid x_i, \Theta^g\right)}{\Sigma_{i=1}^{N} p\left(l \mid x_i, \Theta^g\right)}.$$

Therefore, we have explicit expressions for iterative calculation of the estimates of the mixture parameters, $\Theta=(\alpha_1, \ldots, \alpha_M)$, and the *Poisson* distribution parameters, $(\theta_1, \ldots, \theta_M)=(\lambda_1, \ldots, \lambda_M)$.

Example 2: If we assume *d*-dimensional Gaussian component distributions with a mean vector μ and covariance matrix Σ, i.e., $\theta = (\mu, \Sigma)$ then the probability density function is

$$p_l(x \mid \theta) = p_l(x \mid \mu_l, \Sigma_l) = \frac{1}{(2\pi)^{d/2}\left|\Sigma_l\right|^{1/2}} Exp\left(-\frac{1}{2}(x-\mu_l)^T \Sigma_l^{-1}(x-\mu_l)\right) \quad (7)$$

Then we may derive the update equations [equations (1), (4), (6)] for this specific distribution, we need to recall some results from matrix algebra. The **trace** of a square matrix *tr(A)* is equal to the sum of *A*'s diagonal elements. In 1D the trace of a scalar equals that scalar. Also, *tr(A + B) = tr(A) + tr(B)*, and *tr(AB) = tr(BA)*, which implies that if $B = \Sigma_i x_i x_i^T \Rightarrow \Sigma_i x_i^T A x_i = tr(AB)$. Also note that $|A|$ indicates the determinant of the matrix *A*, and $|A^{-1}|=1/|A|$. Differentiation of a function of a matrix *f(A)* is accomplished by differentiating with respect to elements of that matrix. Therefore, we define *df(A)/dA* to be the matrix with (i,j)-th entry equal to $[df(A)/da_{i,j}]$, where $A=(a_{i,j})$. This definition also applies taking derivatives with respect to a vector. First, $d(x^TAx)/dx=(A+A^T)x$. Second, it can be shown that when *A* is a symmetric matrix:

$$\frac{\partial \mid A \mid}{\partial a_{i,j}} = \begin{cases} A_{i,j}, \text{if } i = j \\ 2A_{i,j}, \text{if } i \neq j \end{cases}$$

where $A_{i,j}$ is the *(i,j)*-the cofactor of *A*. Given the above, we see that:

$$\frac{\partial Log \mid A \mid}{\partial A} = \begin{Bmatrix} A_{i,j}/|A|, \text{if } i = j \\ 2A_{i,j}/|A|, \text{if } i \neq j \end{Bmatrix} = 2A^{-1} - diag(A^{-1})$$

by the definition of the inverse of a matrix. Finally, it can be shown that $d\text{tr}(AB)/dA = B + B^T - diag(B)$.

Now, for the *d*-dimensional Gaussian distribution example, if we take a log of equation (7), ignoring any constant terms (since they disappear after taking derivatives), and substituting into the right side of equation (6), we get:

$$Q\left(\Theta,\Theta^{(g)}\right)=\Sigma_{l=1}^{M}\Sigma_{i=1}^{N}Log\left(p_l(x_i\mid\mu_l,\Sigma_l)\right)p(l\mid x_i,\Theta^g)$$
$$=\Sigma_{l=1}^{M}\Sigma_{i=1}^{N}\left(-\frac{1}{2}Log(\mid\Sigma_l\mid)-\frac{1}{2}\left(x-\mu_l\right)^T\Sigma_l^{-1}\left(x-\mu_l\right)\right)p(l\mid x_i,\Theta^g) \quad (8)$$

Taking the derivative of equation (8) with respect to μ_l and setting it equal to zero, we get:

$$\Sigma_{i=1}^{N}\left(\Sigma_l^{-1}(x_i-\mu_l)p\left(l\mid x_i,\Theta^g\right)\right)=0$$

which solving for μ_l yields:

$$\mu_l=\frac{\Sigma_{i=1}^{N}x_i\,p\left(l\mid x_i,\Theta^g\right)}{\Sigma_{i=1}^{N}p\left(l\mid x_i,\Theta^g\right)}$$

To find Σ_l, note that we can write equation (8) as:

$$\Sigma_{l=1}^{M}\left[\frac{1}{2}Log\left|\Sigma_l^{-1}\right|\Sigma_{i=1}^{N}p(l\mid x_i,\Theta^g)-\frac{1}{2}\Sigma_{i=1}^{N}p(l\mid x_i,\Theta^g)\ \mathrm{tr}\left[\Sigma_l^{-1}\left(x-\mu_l\right)\left(x-\mu_l\right)^T\right]\right]=$$
$$=\Sigma_{l=1}^{M}\left[\frac{1}{2}Log\left|\Sigma_l^{-1}\right|\Sigma_{i=1}^{N}p(l\mid x_i,\Theta^g)-\frac{1}{2}\Sigma_{i=1}^{N}p(l\mid x_i,\Theta^g)\ \mathrm{tr}\left[\Sigma_l^{-1}N_{l,i}\right]\right] \quad (9)$$

where $N_{l,i}=\left(x-\mu_l\right)\left(x-\mu_l\right)^T$. In equation (9) we now take the derivative with respect to the matrix Σ_l^{-1}, and we obtain:

$$\frac{1}{2}\Sigma_{i=1}^{N}p(l\mid x_i,\Theta^g)\left(2\Sigma_l-diag(\Sigma_l)\right)-\frac{1}{2}\Sigma_{i=1}^{N}p(l\mid x_i,\Theta^g)\left(2N_{l,i}-diag(N_{l,i})\right)=$$
$$=\frac{1}{2}\Sigma_{i=1}^{N}p(l\mid x_i,\Theta^g)\left(2M_{l,i}-diag(M_{l,i})\right)=2S-diag(S), \quad (10)$$

where $N_{l,i}=\left(x-\mu_l\right)\left(x-\mu_l\right)^T$, $M_{l,i}=\Sigma_l-N_{l,i}$ and $S=\frac{1}{2}\Sigma_{i=1}^{N}p(l\mid x_i,\Theta^g)M_{l,i}$. To find the extreme values (maxima) of the , equation (9) we set the derivative to zero [equation (10)], i.e., ***2S – diag(S) = 0***. This implies that $S=0$ ➔ $\Sigma_{i=1}^{N}p(l\mid x_i,\Theta^g)\left(\Sigma_l-N_{l,i}\right)=0$. So, we obtain an exact expression (variance-covariance matrix estimate, $1\le l\le M$) for Σ_l.

$$\Sigma_l = \frac{\sum_{i=1}^{N} p(l \mid x_i, \Theta^g) N_{l,i}}{\sum_{i=1}^{N} p(l \mid x_i, \Theta^g)} = \frac{\sum_{i=1}^{N} \left[p(l \mid x_i, \Theta^g) \left((x_i - \mu_l)(x_i - \mu_l)^T \right) \right]}{\sum_{i=1}^{N} p(l \mid x_i, \Theta^g)}$$

Summarizing, the estimates of the new parameters in terms of the old parameters (guessed parameters super-indexed by ^{g}, $\Theta^g = (\alpha_1^g, \ldots, \alpha_M^g ; \theta_1^g, \ldots, \theta_M^g)$) are as follows:

$$\alpha_l^{new} = \frac{1}{N} \sum_{i=1}^{N} p(l \mid x_i, \Theta^g)$$

$$\mu_l^{new} = \frac{\sum_{i=1}^{N} x_i p(l \mid x_i, \Theta^g)}{\sum_{i=1}^{N} p(l \mid x_i, \Theta^g)} \qquad (11)$$

$$\Sigma_l^{new} = \frac{\sum_{i=1}^{N} \left[p(l \mid x_i, \Theta^g) \left((x_i - \mu_l^{new})(x_i - \mu_l^{new})^T \right) \right]}{\sum_{i=1}^{N} p(l \mid x_i, \Theta^g)}$$

Note that the above equations (11) 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.

See Online SOCR Demo at:
http://socr.stat.ucla.edu/Applets.dir/MixtureEM.html

References:

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.

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

J.A. Bilmes. A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Department of Electrical Engineering and Computer Science, U.C. Berkeley, Berkeley, CA 94704, *TR-97-021*, April 1998.

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.

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

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

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

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

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

L. Xu and M.I. Jordan. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Computation*, 8:129–151, 1996.