Estimating Gaussian Mixture Densities with EM – A Tutorial

Carlo Tomasi – Duke University

Expectation Maximization (EM) [4, 3, 6] is a numerical algorithm for the maximization of functions of several variables. There are several tutorial introductions to EM, including [8, 5, 2, 7]. These are excellent references for greater generality about EM, several good intuitions, and useful explanations. The purpose of this document is to explain in a more self-contained way how EM can solve a special but important problem, the estimation of the parameters of a mixture of Gaussians from a set of data points. Here is the outline of what follows:

1. A comparison of EM with Newton's method
2. The density estimation problem
3. Membership probabilities
4. Characterization of the local maxima of the likelihood
5. Jensen's inequality
6. The EM algorithm

1 A Comparison of EM with Newton's Method

The abstract idea of EM can be understood by comparison with Newton's method for maximizing a scalar function $f(\theta)$ of a vector θ of variables. Newton's method starts at a given point $\theta^{(0)}$, approximates f in a neighborhood of $\theta^{(0)}$ with a paraboloid $b_0(\theta)$, and finds the maximum of the paraboloid by solving a system of linear equations to obtain a new point $\theta^{(1)}$. This procedure is repeated for $\theta = \theta^{(1)}, \theta^{(2)}, \ldots$ until the change from $\theta^{(i-1)}$ to $\theta^{(i)}$ is small enough, thereby signaling convergence to a point that we call θ^*, a local maximum of f.

EM works in a similar fashion, but instead of approximating f with a paraboloid at $\theta^{(i)}$ it finds a new function $b_i(\theta)$ with the following properties:

- b_i is a lower bound for f, that is $b_i(\theta) \leq f(\theta)$ everywhere in a neighborhood of the current point $\theta^{(i)}$.
- The functions b_i and f touch at $\theta^{(i)}$, that is, $f(\theta^{(i)}) = b_i(\theta^{(i)})$.

Both EM and Newton's method satisfy the second property, $f(\theta^{(i)}) = b_i(\theta^{(i)})$. However, for EM b_i need not be a paraboloid. On the other hand, Newton's paraboloid is not required to be a lower bound for f, while EM's b_i function is. So neither method is an extension of the other.

In general, it is not clear which of the two methods would give better results: it really depends on the function f and, for EM, on the shape of the lower bounds b_i. However, for both methods there are convergence guarantees. For Newton's method, the idea is that every function looks like a paraboloid in a small enough neighborhood of any point θ, so maximizing b_i becomes increasingly closer to maximizing f. Once $\theta^{(i)}$ is close to the maximum θ^*, one can actually give bounds on the number of iterations that it takes to reach the maximum itself.

For the EM method, it is obvious that at the point of maximum $\theta^{(i+1)}$ of b_i we must have $f(\theta^{(i+1)}) \geq f(\theta^{(i)})$. This is because $b_i(\theta^{(i)})$ cannot be smaller than $f(\theta^{(i)})$ (the two functions are required to be equal there), the maximum $b_i(\theta^{(i+1)})$ of b_i cannot be smaller than $b_i(\theta^{(i)})$ (or else it would not be a maximum), and $f(\theta^{(i+1)}) \geq b_i(\theta^{(i+1)})$ (because b_i is a lower bound for f). In summary, $f(\theta^{(i+1)}) \geq b_i(\theta^{(i+1)}) \geq b_i(\theta^{(i)}) = f(\theta^{(i)})$, so that $f(\theta^{(i+1)}) \geq$

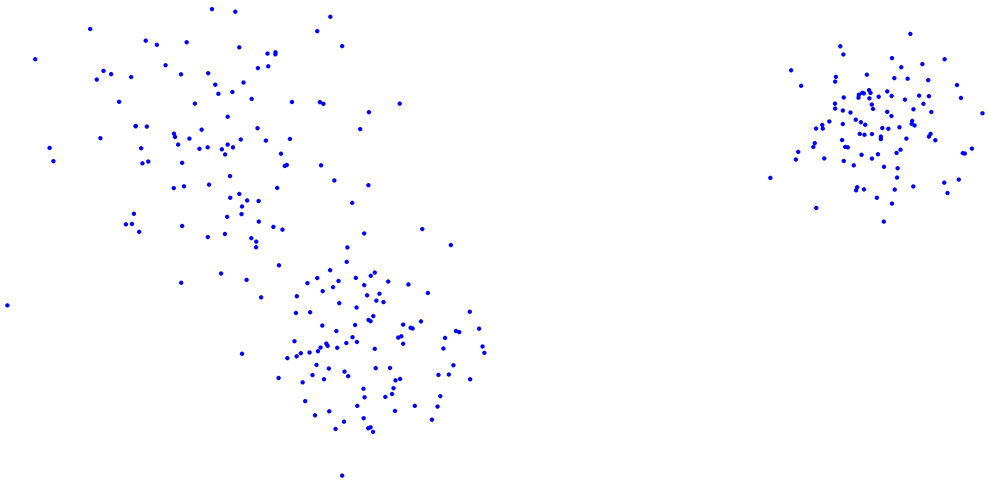

Figure 1: Three-hundred points on the plane.

$f(\theta^{(i)})$, as promised. This by itself is no guarantee of progress, since it could still be that $f(\theta^{(i+1)}) = f(\theta^{(i)})$, but at least EM does not go downhill.

The name of the game, then, is to be clever enough with the way b_i is built to be able to show that the progress made at every iteration is finite and bounded from below. If at every step we go uphill by at least ϵ and if the function f has a maximum, then at some point we are bound to reach the top. If in addition we find bound functions b_i that are very similar to f, then it is possible that EM works even better than Newton's method.

For some functions f this game is hard to play. Other functions seem to be designed so that everything works out just fine. One version of the all-important *density estimation* problem makes the EM idea work very well. In the following section, density estimation is defined for mixtures of Gaussian functions. The sections thereafter show how to use EM to solve the problem.

2 The Density Estimation Problem

Suppose that you are given a set of points as in Figure 1. These points are on the plane, but nothing about the current theory is limited to two dimensions.

The points in the figure seem to be grouped in clusters. One cluster, on the right, is nicely separated from the others. Two more clusters, on the left, are closer to each other, and it is not clear if a meaningful dividing line can be drawn between them.

The *density estimation* problem can be loosely defined as follows: given a set of N points in D dimensions, $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathcal{R}^D$, and a family $\mathcal{F}$ of probability density functions on $\mathcal{R}^D$, find the probability density $f(\mathbf{x}) \in \mathcal{F}$ that is most likely to have generated the given points.

One way to define the family $\mathcal{F}$ is to give each of its members the same mathematical form, and to distinguish different members by different values of a set of parameters θ. For instance, the functions in $\mathcal{F}$ could be *mixtures of Gaussian functions*:

$$f(\mathbf{x};\, \theta) = \sum_{k=1}^{K} p_k \, g(\mathbf{x};\, \mathbf{m}_k, \sigma_k) \tag{1}$$

where

$$g(\mathbf{x};\, \mathbf{m}_k, \sigma_k) = \frac{1}{(\sqrt{2\pi}\sigma_k)^D} \, e^{-\frac{1}{2}\left(\frac{\|\mathbf{x}-\mathbf{m}_k\|}{\sigma_k}\right)^2}$$

is a D-dimensional isotropic[1] Gaussian function and $\theta = (\theta_1, \ldots, \theta_K) = ((p_1, \mathbf{m}_1, \sigma_1), \ldots, (p_K, \mathbf{m}_K, \sigma_K))$ is a $K(D+2)$-dimensional vector containing the *mixing probabilities* p_k as well as the means $\mathbf{m}_k$ and standard deviations σ_k of the K Gaussian functions in the mixture.[2]

Each Gaussian function integrates to one:

$$\int_{\mathcal{R}^D} g(\mathbf{x};\, \mathbf{m}_k, \sigma_k)\, d\mathbf{x} = 1 \ .$$

Since f is a density function, it must be nonnegative and integrate to one as well. We have

$$1 = \int_{\mathcal{R}^D} f(\mathbf{x};\, \theta)\, d\mathbf{x} = \int_{\mathcal{R}^D} \sum_{k=1}^{K} p_k\, g(\mathbf{x};\, \mathbf{m}_k, \sigma_k)\, d\mathbf{x} = \sum_{k=1}^{K} p_k \int_{\mathcal{R}^D} g(\mathbf{x};\, \mathbf{m}_k, \sigma_k)\, d\mathbf{x} = \sum_{k=1}^{K} p_k$$

so that the numbers p_k, which must be nonnegative lest $f(\mathbf{x})$ takes on negative values, must add up to one:

$$p_k \geq 0 \qquad \text{and} \qquad \sum_{k=1}^{K} p_k = 1 \ .$$

This is why the numbers p_k are called mixing *probabilities*.

Mixtures of Gaussian functions are obviously well-suited to modelling clusters of points: each cluster is assigned a Gaussian, with its mean somewhere in the middle of the cluster, and with a standard deviation that somehow measures the spread of that cluster.[3] Another way to view this modelling problem is to note that the cloud of points in Figure 1 could have been generated by repeating the following procedure N times, once for each point $\mathbf{x}_n$:

- Draw a random integer between 1 and K with probability p_k of drawing k. This selects the cluster from which to draw point $\mathbf{x}_n$.
- Draw a random D-dimensional real vector $\mathbf{x}_n \in \mathcal{R}^D$ from the k-th Gaussian density $g(\mathbf{x};\, \mathbf{m}_k, \sigma_k)$.

This is called a *generative model* for the given set of points.[4]

Since the family of mixtures of Gaussian functions is parametric, the density estimation problem can be defined more specifically as the problem of finding the vector θ of parameters that specifies the model from which the points are most likely to be drawn.

What remains to be determined is the meaning of "most likely." We want a function $\Lambda(X;\, \theta)$ that measures the likelihood of a particular model given the set of points: the set X is fixed, because the points $\mathbf{x}_n$ are given, and Λ is required to be large for those vectors θ of parameters such that the mixture of Gaussian functions $f(\mathbf{x};\, \theta)$ is likely to generate sets of points like the given one.

Bayesians will derive Λ from Bayes's theorem, but at the cost of having to specify a prior distribution for θ itself. We take the more straightforward Maximum Likelihood approach: the probability of drawing a value in a small volume of size $d\mathbf{x}$ around $\mathbf{x}$ is $f(\mathbf{x};\, \theta)\, d\mathbf{x}$. If the random draws in the generative model are independent of each other, the probability of drawing a sample of N points where each point is in a volume of size $d\mathbf{x}$ around one of the given $\mathbf{x}_n$ is the product $f(\mathbf{x}_1;\, \theta)\, d\mathbf{x} \cdots f(\mathbf{x}_N;\, \theta)\, d\mathbf{x}$. Since the volume $d\mathbf{x}$ is constant, it can be ignored when looking for the θ that yields the highest probability, and the likelihood function can be defined as follows:

$$\Lambda(X;\, \theta) = \prod_{n=1}^{N} f(\mathbf{x}_n;\, \theta) \ .$$

[1] The restriction to isotropic Gaussian functions is conceptually minor but technically useful. For now, the most important implication of this restriction is that the spread of each Gaussian function is measured by a scalar, rather than by a covariance matrix.

[2] The number K of Gaussian functions could itself be a parameter subject to estimation. In this document, we consider it to be known.

[3] Here the assumption of isotropic Gaussian functions is a rather serious limitation, since elongated clusters cannot be modelled well.

[4] As you may have guessed, the points in Figure 1 were indeed generated in this way.

For mixtures of Gaussian functions, in particular, we have

$$\Lambda(X;\,\theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} p_k\, g(\mathbf{x}_n;\,\mathbf{m}_k, \sigma_k)\ . \tag{2}$$

In summary, the parametric density estimation problem can be defined more precisely as follows:

$$\hat{\theta} = \arg\max_{\theta}\ \Lambda(X;\,\theta)\ .$$

In principle, any maximization algorithm could be used to find $\hat{\theta}$, the maximum likelihood estimate of the parameter vector θ. In practice, as we will see in section 4, EM works particularly well for this problem.

3 Membership Probabilities

In view of further manipulation, it is useful to understand the meaning of the terms

$$q(k,n) = p_k\, g(\mathbf{x}_n;\,\mathbf{m}_k, \sigma_k) \tag{3}$$

that appear in the definition (1) of a mixture density. When defining the likelihood function Λ we have assumed that the event of drawing component k of the generative model is independent of the event of drawing a particular data point $\mathbf{x}_n$ out of a particular component. It then follows that $q(k,n)\,d\mathbf{x}$ is the joint probability of drawing component k and drawing a data point in a volume $d\mathbf{x}$ around $\mathbf{x}_n$. The definition of conditional probability,

$$P(A\,|\,B) = \frac{P(A \cap B)}{P(B)}\ ,$$

then tells us that the conditional probability of having selected component k given that data point $\mathbf{x}_n$ was observed is

$$p(k\,|\,n) = \frac{q(k,n)}{\sum_{m=1}^{K} q(m,n)}\ . \tag{4}$$

Note that the volume $d\mathbf{x}$ cancels in the ratio that defines $p(k\,|\,n)$, so that this result holds even when $d\mathbf{x}$ vanishes to zero. Also, it is obvious from the expression of $p(k\,|\,n)$ that

$$\sum_{k=1}^{K} p(k\,|\,n) = 1\ ,$$

in keeping with the fact that it is certain that some component k has generated data point $\mathbf{x}_n$. Let us call the probabilities $p(k\,|\,n)$ the *membership probabilities*, because they express the probability that point n was generated by component k.

4 Characterization of the Local Maxima of Λ

The logarithm of the likelihood function $\Lambda(X;\,\theta)$ defined in equation (2) is

$$\lambda(X;\,\theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} p_k\, g(\mathbf{x}_n;\,\mathbf{m}_k, \sigma_k)\ .$$

To find expressions that are valid at local maxima of λ (or equivalently Λ) we follow [1], and compute the derivatives of λ with respect to $\mathbf{m}_k$, σ_k, p_k. Since

$$\frac{\partial g(\mathbf{x}_n;\,\mathbf{m}_k, \sigma_k)}{\partial \mathbf{m}_k} = g(\mathbf{x}_n;\,\mathbf{m}_k, \sigma_k)\,\frac{\partial}{\partial \mathbf{m}_k}\left[-\frac{1}{2}\left(\frac{\|\mathbf{x}_n - \mathbf{m}_k\|}{\sigma_k}\right)^2\right]$$

and

$$\frac{\partial}{\partial \mathbf{m}_k}\|\mathbf{x}_n - \mathbf{m}_k\|^2 = \frac{\partial}{\partial \mathbf{m}_k}\left(\mathbf{x}_n^T\mathbf{x}_n + \mathbf{m}_k^T\mathbf{m}_k - 2\mathbf{x}_n^T\mathbf{m}_k\right) = 2(\mathbf{x}_n - \mathbf{m}_k) \ ,$$

we can write

$$\frac{\partial \lambda}{\partial \mathbf{m}_k} = \sum_{n=1}^{N} \frac{1}{\sigma_k^2} \frac{p_k \, g(\mathbf{x}_n;\, \mathbf{m}_k, \sigma_k)}{\sum_{m=1}^{K} p_m \, g(\mathbf{x}_n;\, \theta_m)} (\mathbf{m}_k - \mathbf{x}_n) = \sum_{n=1}^{N} \frac{p(k \,|\, n)}{\sigma_k^2} \, (\mathbf{m}_k - \mathbf{x}_n)$$

where we have used equations (3) and (4).

A similar procedure as for $\mathbf{m}_k$ yields

$$\frac{\partial \lambda}{\partial \sigma_k} = \sum_{n=1}^{N} p(k \,|\, n) \left(-\frac{D}{\sigma_k} + \frac{\|\mathbf{x}_n - \mathbf{m}_k\|^2}{\sigma_k^3} \right)$$

(recall that D is the dimension of the space of the data points).

The derivative of λ with respect to the mixing probabilities p_k requires a little more work, because the values of p_k are constrained to being positive and adding up to one. After [1], this constraint can be handled by writing the variables p_k in turn as functions of unconstrained variables γ_k as follows:

$$p_k = \frac{e^{\gamma_k}}{\sum_{k=1}^{K} e^{\gamma_k}} \ .$$

This trick to express p_k through what is usually called a *softmax* function, enforces both constraints automatically. We then have

$$\frac{\partial p_k}{\partial \gamma_j} = \begin{cases} p_k - p_k^2 & \text{if } j = k \\ -p_j p_k & \text{otherwise} \end{cases} \ .$$

From the chain rule for differentiation, we obtain

$$\frac{\partial \lambda}{\partial \gamma_k} = \sum_{n=1}^{N} (p(k \,|\, n) - p_k) \ .$$

Setting the derivatives we just found to zero, we obtain three groups of equations for the means, standard deviations, and mixing probabilities:

$$\mathbf{m}_k = \frac{\sum_{n=1}^{N} p(k \,|\, n) \mathbf{x}_n}{\sum_{n=1}^{N} p(k \,|\, n)} \tag{5}$$

$$\sigma_k = \sqrt{\frac{1}{D} \frac{\sum_{n=1}^{N} p(k \,|\, n) \|\mathbf{x}_n - \mathbf{m}_k\|^2}{\sum_{n=1}^{N} p(k \,|\, n)}} \tag{6}$$

$$p_k = \frac{1}{N} \sum_{n=1}^{N} p(k \,|\, n) \ . \tag{7}$$

The first two expressions make immediate intuitive sense, because $\mathbf{m}_k$ and σ_k are the sample mean and standard deviation of the sample data, weighted by the conditional probability that data point n was generated by model k. The third equation, for the mixing probabilities, is not immediately obvious, but it is not too surprising either, since it views p_k as the sample mean of the conditional probabilities $p(k \,|\, n)$ assuming a uniform distribution over all the data points.

These equations are intimately coupled with one another, because the terms $p(k \,|\, n)$ in the right-hand sides depend in turn on all the terms on the left-hand sides through equations (3) and (4). Because of this the equations above are hard to solve directly. However, the lower bound idea discussed in section 1 provides an iterative solution. As a preview, this solution involves starting with a guess for θ, the vector of all parameters p_k, $\mathbf{m}_k$, σ_k, and then iteratively cycling through equations (3), (4) (the "E step"), and then equations (5), (6), (7).

The next section shows an important technical result that is used to compute a bound for the logarithm of the likelihood function. The EM algorithm is then derived in section 6, after the principle discussed in section 1.

5 Jensen's Inequality

Jensen's inequality is often used to bound the logarithm of a sum of terms, just like the one in the expression of λ: Given K nonnegative numbers $\pi_1, \ldots, \pi_K$ that add up to one (that is, a discrete probability distribution) and K arbitrary numbers $a_1, \ldots, a_K$, it follows from the convexity of the logarithm that

$$\log \sum_{k=1}^{K} \pi_k a_k \geq \sum_{k=1}^{K} \pi_k \log a_k$$

(see for instance exercise 2.13 in [1] for a proof outline).

From this inequality we can also derive the following useful expression, where π_k is still an element of a discrete probability distribution and c_k are arbitrary numbers:

$$\log \sum_{k=1}^{K} c_k = \log \sum_{k=1}^{K} c_k \frac{\pi_k}{\pi_k} \geq \sum_{k=1}^{K} \pi_k \log \frac{c_k}{\pi_k} \tag{8}$$

obtained from Jensen's inequality with $a_k = c_k/\pi_k$. The inequality (8) is generally useful in probabilistic manipulation because it bounds the logarithm of a sum with the expected value of a logarithm.

6 The EM Algorithm

Assume that approximate (possibly very bad) estimates $p_k^{(i)}$, $\mathbf{m}_k^{(i)}$, $\sigma_k^{(i)}$ are available for the parameters of the likelihood function $\Lambda(X;\, \theta)$ or its logarithm $\lambda(X;\, \theta)$. Then, better estimates $p_k^{(i+1)}$, $\mathbf{m}_k^{(i+1)}$, $\sigma_k^{(i+1)}$ can be computed by first using the old estimates to construct a lower bound $b_i(\theta)$ for the likelihood function, and then maximizing the bound with respect to p_k, $\mathbf{m}_k$, σ_k.

Expectation Maximization (EM) starts with initial values $p_k^{(0)}$, $\mathbf{m}_k^{(0)}$, $\sigma_k^{(0)}$ for the parameters, and iteratively performs these two steps until convergence. Construction of the bound $b_i(\theta)$ is called the "E step," because the bound is the expectation of a logarithm, derived from use of inequality (8). The maximization of $b_i(\theta)$ that yields the new estimates $p_k^{(i+1)}$, $\mathbf{m}_k^{(i+1)}$, $\sigma_k^{(i+1)}$ is called the "M step." This section derives expressions for these two steps.

Given the old parameter estimates $p_k^{(i)}$, $\mathbf{m}_k^{(i)}$, $\sigma_k^{(i)}$, we can compute estimates $p^i(k \,|\, n)$ for the membership probabilities from equation (3) and (4):

$$p^{(i)}(k \,|\, n) = \frac{p_k^{(i)}\, g(\mathbf{x}_n;\, \mathbf{m}_k^{(i)}, \sigma_k^{(i)})}{\sum_{m=1}^{K} p_k^{(i)}\, g(\mathbf{x}_n;\, \mathbf{m}_k^{(i)}, \sigma_k^{(i)})} \,. \tag{9}$$

This is the actual computation performed in the E step. The rest of the "construction" of the bound $b_i(\theta)$ is theoretical, and uses form (8) of Jensen's inequality to bound the logarithm $\lambda(X;\, \theta)$ of the likelihood function as follows.

The membership probabilities $p(k \,|\, n)$ add up to one, and their estimates $p^{(i)}(k \,|\, n)$ do so as well, because they are computed from equation (9), which includes explicit normalization. Thus, we can let $\pi_k = p^{(i)}(k \,|\, n)$ and $c_k = q(k, n)$ in the inequality (8) to obtain

$$\lambda(X;\, \theta) \quad = \quad \sum_{n=1}^{N} \log \sum_{k=1}^{K} q(k, n) \geq \sum_{n=1}^{N} \sum_{k=1}^{K} p^{(i)}(k \,|\, n) \, \log \frac{q(k, n)}{p^{(i)}(k \,|\, n)} = b_i(\theta) \,.$$

The bound $b_i(\theta)$ thus obtained can be rewritten as follows:

$$b_i(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} p^{(i)}(k \,|\, n) \, \log q(k, n) - \sum_{n=1}^{N} \sum_{k=1}^{K} p^{(i)}(k \,|\, n) \, \log p^{(i)}(k \,|\, n) \,.$$

Since the old membership probabilities $p^{(i)}(k \mid n)$ are known, minimizing $b_i(\theta)$ is the same as minimizing the first of the two summations, that is, the function

$$\beta_i(\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} p^{(i)}(k \mid n) \, \log q(k, n) \; .$$

Prima facie, the expression for $\beta_i(\theta)$ would seem to be even more complicated than that for λ. This is not so, however: rather than the logarithm of a sum, $\beta_i(\theta)$ contains a linear combination of K logarithms, and this breaks the coupling of the equations obtained by setting the derivatives of $\beta_i(\theta)$ with respect to the parameters to zero.

The derivative of $\beta_i(\theta)$ with respect to $\mathbf{m}_k$ is easily found to be

$$\frac{\partial \beta_i}{\partial \mathbf{m}_k} = \sum_{n=1}^{N} p^{(i)}(k \mid n) \, \frac{\mathbf{m}_k - \mathbf{x}_n}{\sigma_k^2} \; .$$

Upon setting this expression to zero, the variance σ_k can be cancelled, and the remaining equation contains only $\mathbf{m}_k$ as the unknown:

$$\mathbf{m}_k \sum_{n=1}^{N} p^{(i)}(k \mid n) = \sum_{n=1}^{N} p^{(i)}(k \mid n) \, \mathbf{x}_n \; .$$

This equation can be solved immediately to yield the new estimate of the mean:

$$\mathbf{m}_k^{(i+1)} = \frac{\sum_{n=1}^{N} p^{(i)}(k \mid n) \, \mathbf{x}_n}{\sum_{n=1}^{N} p^{(i)}(k \mid n)} \; ,$$

which is only a function of old values (with superscript i).

The resulting value $\mathbf{m}_k^{(i+1)}$ can now be replaced into the expression for $\beta_i(\theta)$, which can now be differentiated with respect to σ_k through a very similar manipulation to yield

$$\sigma_k^{(i+1)} = \sqrt{\frac{1}{D} \, \frac{\sum_{n=1}^{N} p^{(i)}(k \mid n) \, \| \mathbf{x}_n - \mathbf{m}_k^{(i+1)} \|^2}{\sum_{n=1}^{N} p^{(i)}(k \mid n)}} \; .$$

The derivative of $\beta_i(\theta)$ with respect to p_k, subject to the constraint that the p_k add up to one, can be handled again through the softmax function just as we did in section 4. This yields the new estimate for the mixing probabilities as a function of the old membership probabilities:

$$p_k^{(i+1)} = \frac{1}{N} \sum_{n=1}^{N} p^{(i)}(k \mid n) \; .$$

In summary, given an initial estimate $p_k^{(0)}$, $\mathbf{m}_k^{(0)}$, $\sigma_k^{(0)}$, EM iterates the following computations until convergence to a local maximum of the likelihood function:

- E Step:

$$p^{(i)}(k \mid n) = \frac{p_k^{(i)} \, g(\mathbf{x}_n; \, \mathbf{m}_k^{(i)}, \sigma_k^{(i)})}{\sum_{m=1}^{K} p_k^{(i)} \, g(\mathbf{x}_n; \, \mathbf{m}_k^{(i)}, \sigma_k^{(i)})}$$

- M Step:

$$\begin{aligned}
\mathbf{m}_k^{(i+1)} &= \frac{\sum_{n=1}^{N} p^{(i)}(k \mid n) \, \mathbf{x}_n}{\sum_{n=1}^{N} p^{(i)}(k \mid n)} \\
\sigma_k^{(i+1)} &= \sqrt{\frac{1}{D} \, \frac{\sum_{n=1}^{N} p^{(i)}(k \mid n) \, \| \mathbf{x}_n - \mathbf{m}_k^{(i+1)} \|^2}{\sum_{n=1}^{N} p^{(i)}(k \mid n)}} \\
p_k^{(i+1)} &= \frac{1}{N} \sum_{n=1}^{N} p^{(i)}(k \mid n) \; .
\end{aligned}$$

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[2] F. Dellaert. The expectation maximization algorithm. http://www.cc.gatech.edu/ ˜dellaert/em-paper.pdf, 2002.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data wia the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–22, 1977.

[4] H. Hartley. Maximum likelihood estimation from incomplete data. *Biometrics*, 14:174–194, 1958.

[5] T. P. Minka. Expectation-maximization as lower bound maximization. http://citeseer.nj.nec.com/minka98expectationmaximization.html, 1998.

[6] R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

[7] J. Rennie. A short tutorial on using expectation-maximization with mixture models. http://www.ai.mit.edu/ people/jrennie/writing/mixtureEM.pdf, 2004.

[8] Y. Weiss. Motion segmentation using EM – a short tutorial. http://www.cs.huji.ac.il/˜yweiss/emTutorial.pdf, 1997.