# An Intuitive Introduction to the EM Algorithm for Gaussian Mixtures*

Vince Stanford and Erik Stanford (vmstanford@gmail.com)

July 27, 2013 (revised March 14, 2019)

**Abstract**

The EM Algorithm has opened previously intractable problems in important application areas to maximum likelihood estimation. But the actual functioning of this estimation technique is difficult to master. We attempt here another intuitive introduction to estimating parameters by iterative *Expectation*, and *Maximization*. Estimation of Gaussian Mixture Models from the EM Algorithm is used to clarify the construction of so-called *complete data* formulations that are used in EM. The GMM re-estimation formulae are derived in detail using only basic multivariable calculus with Lagrange Multipliers, which we hope makes the method more widely accessible to students, scientists, and engineers.

## EM and applications.

Maximum Likelihood Estimation (MLE) by the Expectation Maximization (EM) Algorithm has significantly broadened the applicability of MLE. EM based algorithms allow estimation of mixtures of random variables from the Exponential Family and are practical for large models. Successful approaches to many problems have Used EM, e.g.:

- Estimation with Missing or censored data
- Economic time series models
- Decoding the Human Genome
- Molecular Spectrometry
- Analysis of censored data, particularly in medical fields where patients are lost to follow-up
- Speech recognition
- Language translation
- Speaker recognition

## Gaussian mixture models

A Gaussian mixture model assumes the overall population is composed of several subpopulations, each of which is Gaussian distributed. Each of the subpopulations is specified by the parameters of a Gaussian component $(\mu, \sigma, \alpha)$ where $\mu$ is the mean, $\sigma$ is the standard deviation, and $\alpha$ is the mixture weight. In Gaussian Mixture distributions, each component $\mu$ is the mean, and each component $\sigma$ is its standard deviation. The mixture weight $\alpha$ is the probability mass of each component of the mixture distribution. To assemble the mixture model, compute the sum of the weighted component distributions. To ensure that the resulting sum is in fact a probability distribution, the constraints $\alpha_j > 0$ and $\sum \alpha_j = 1$ need to be enforced, since the area under the curve defined by the mixture model is necessarily equal to $\sum \alpha_j$.

In order to obtain the formula for a Gaussian mixture density, let $A = \{j = 1...C|\alpha_j\}$ be the set of all mixture weights, $M = \{j = 1...C|\mu_j\}$ be the set of all the means, and $\Sigma = \{j = 1...C|\sigma_j\}$ be the set of all standard deviations of the component distributions. Then the mixture is given by:

$$p\left(x|A, M, \Sigma\right) = \sum_{j=1}^{C} \alpha_j \cdot p\left(x|\mu_j, \sigma_j\right)$$

where $p\left(x|\mu_j, \sigma_j\right) = \frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_j}{\sigma_j}\right)^2}$. An example mixture is shown in figure 1.

In order to attempt a ML estimate of the parameters of the mixture we construct its log-likelihood function. Recall that in general, a likelihood function gives the likelihood of a set of parameter values given a set of observations. In this case, the mixture model will estimate the likelihood that the observed data was generated by component distributions with the specified parameters. Less formally, the Gaussian mixture model tells you how likely it is that you are correct about what components are generating the observed distribution of the data. We can now give the symbolic statement of the model:

$$\begin{aligned} l\left(A, M, \Sigma|X\right) &= \prod_{i=1}^{N} \left( \sum_{j=1}^{C} \alpha_j \cdot p\left(x|\mu_j, \sigma_j\right) \right) \\ log\left(l\left(A, M, \Sigma|X\right)\right) &= log\left( \prod_{i=1}^{N} \sum_{j=1}^{C} \alpha_j \cdot p\left(x|\mu_j, \sigma_j\right) \right) \\ L\left(A, M, \Sigma|X\right) &= \sum_{i=1}^{N} log\left[ \sum_{j=1}^{C} \alpha_j \cdot p\left(x|\mu_j, \sigma_j\right) \right] \end{aligned}$$

We quickly see that $L$ is not easily maximized because the logarithm of the sum does not have partial derivatives with closed form solutions for the mixture parameters we need to estimate. So we will not be able to solve a set of the usual least squares *Normal Equations* to estimate the parameters. So what to do? We can use Newton's method based on second derivatives, but this may not be numerically tractable for mixtures of practical interest. So we will have to consider the implications of a *complete data* formulation of the EM Algorithm in order to arrive at an iterative solution. But what is the *complete data*?

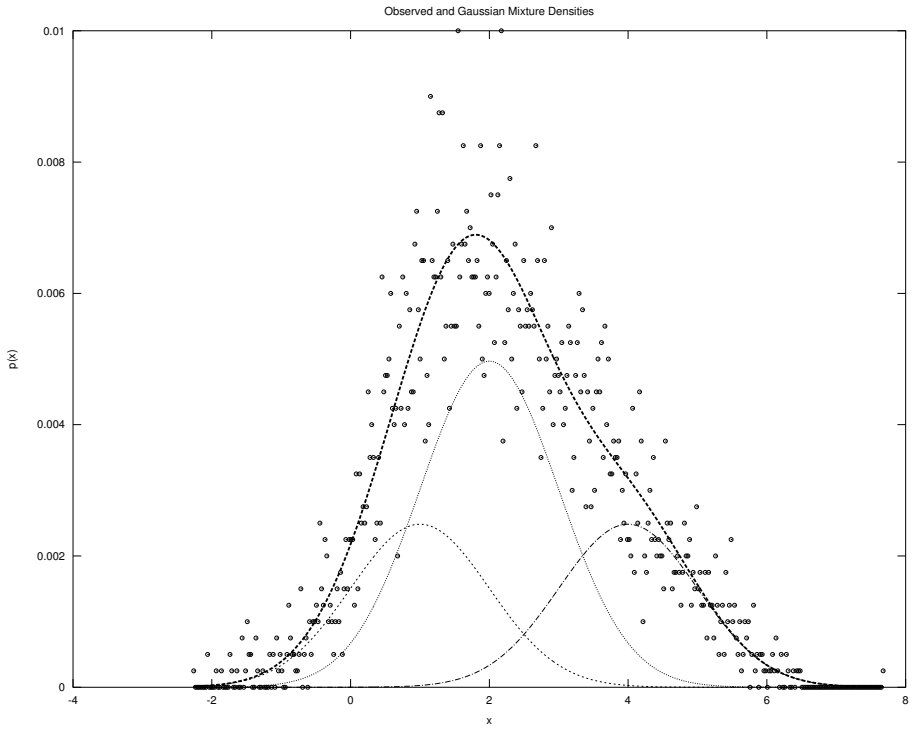


Figure 1: **Gaussian Mixture Model components and sum.** Circles represent the sample histogram, and the dotted lines represent the Gaussian components contributing to the mixture shown as a darker solid line. Estimating the parameters of the individual components of the mixture is our goal.

## Karl Pearson in 1894: On the Dissection of Asymmetrical Frequency-Curves

An early Gaussian mixture fitting algorithm was given by Karl Pearson in 1894 to fit a two-component Gaussian mixture to data [9]. Pearson was asked by a biologist: Do the data on Breadth-to-Forehead ratio of Naples Crabs justify the assertion that two species were being caught? He used the method of moments to set up a ninth-order polynomial whose solutions included the parameters of a univariate Gaussian mixture with two components. Remarkably, he was able to solve for the parameters. He used the first five moments to solve the polynomial and the sixth moment to chose among the various solutions of the ninth-order polynomial. In the late Nineteenth Century what we call the Normal or Gaussian Distribution was known simply as the "Law of Errors" because it appeared so often in scientific measurements of continuous variables like: length, weight, times of astronomical events, and quantities of chemical compounds. The expression of a frequency distribution as the sum of Gaussian distributions was a radical departure at the time.

Pearson solved the following system of six equations to find estimates of the two-component Gaussian Mixture Model, but it was by no means a routine exercise to do so. Equations like these are found using the moment generating function $\left(M_X(t) = E\left[e^{tX}\right]\right)$ of the Normal Distribution. Since this is a linear operator, the hypothesized mixture distribution can be decomposed into the weighted MGF's of the component mixture distributions.

$$z_1 + z_2 = 1$$

$$\gamma_1 z_1 + \gamma_2 z_2 = 0$$

$$\gamma_1^2 z_1 \left(1 + u_1^2\right) + \gamma_2^2 z_2 \left(1 + u_2^2\right) = \mu_2$$

$$\gamma_1^3 z_1 \left(1 + 3u_1^2\right) + \gamma_2^3 z_2 \left(1 + 3u_2^2\right) = \mu_3$$

$$\gamma_1^4 z_1 \left(1 + 6u_1^2 + 3u_1^4\right) + \gamma_2^4 z_2 \left(1 + 6u_2^2 + 3u_2^4\right) = \mu_4$$

$$\gamma_1^5 z_1 \left(1 + 10u_1^2 + 15u_1^4\right) + \gamma_2^5 z_2 \left(1 + 10u_2^2 + 15u_2^4\right) = \mu_5$$

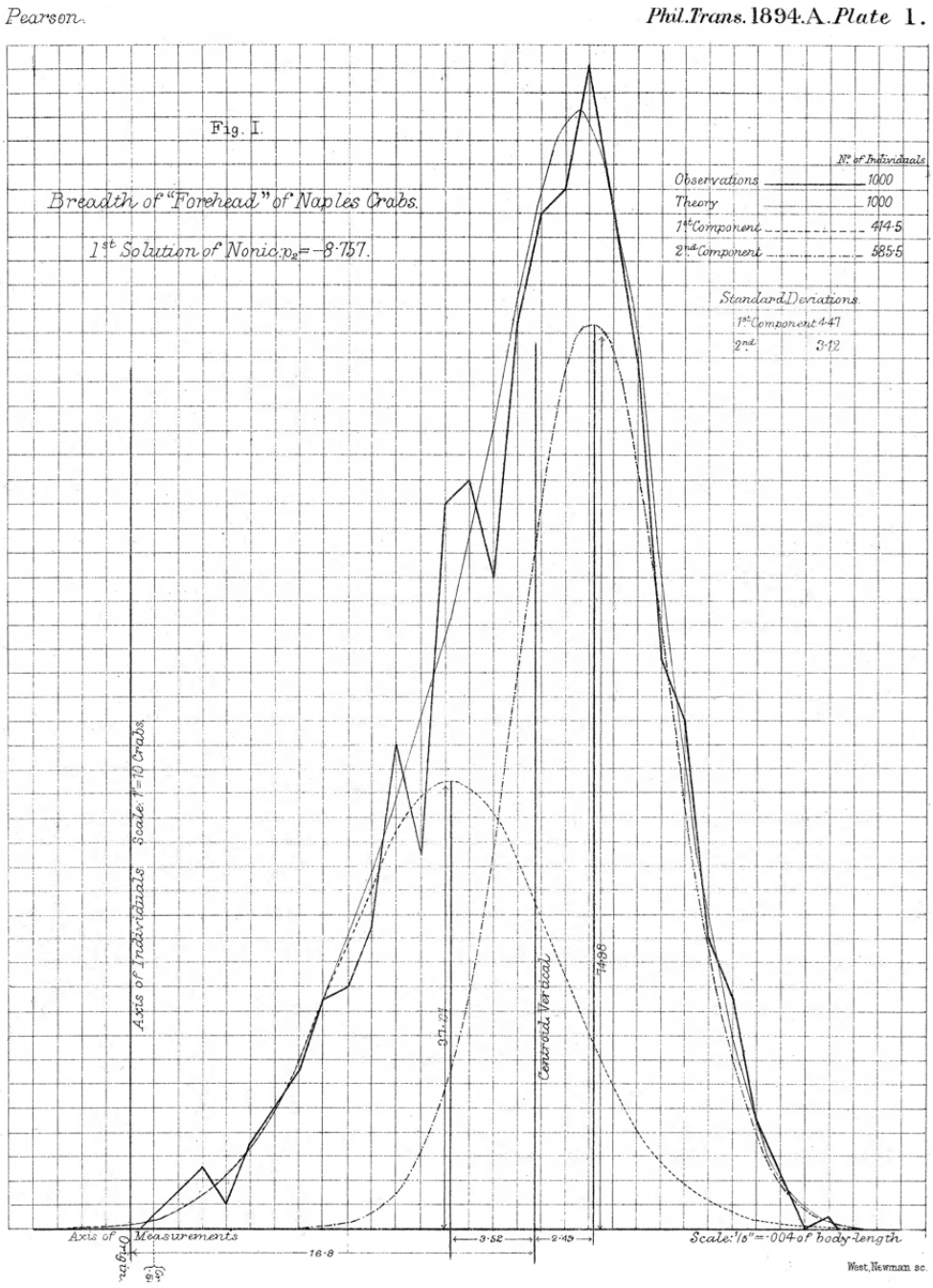


Figure 2: **Pearson's Fig. I.** Do the data on breadth of Forehead of Naples Crabs justify the assertion that two species were being caught? Pearson formulated a $9^{th}$ degree polynomial in the mixture sample moments and solved it *by hand* to obtain estimates of the mixture components shown. This remained the high water mark in mixture estimation for decades, until the advent of digital computers allowed new classes of algorithms, including the EM Algorithm, to be developed.

The method of moments was extended somewhat recently by Kalai, Moitra, and Valient[1] to the case of two component Gaussian mixtures in n-dimensions. This was developed for image processing to avoid the high computational loads of the relatively slow converging EM Algorithm. However, the method of moments does not generalize to larger mixture models, and is fraught with numerical difficulties. In contrast, the EM algorithm has allowed numerically stable computations enabling very large mixture models in high dimensions to be estimated. Mixtures are used for:

- Classification: Are there multiple species here?
- Clustering: Do these things fall into natural groups?
- Estimation with missing data: Big data is never perfectly complete; so what do we do about that?
- Density estimation: What is a good approximation to the Bayesian classifier?

## What the EM Algorithm does

The EM algorithm is used to transform an explicit but intractable problem to a series of tractable subproblems which can be solved as follows:

- Construct a simplified problem involving unknown (or hidden) variables, which if known would allow solution of the explicit problem.
- Initialize the parameters of the explicit problem in some way (e.g. by approximating a uniform prior probability distribution).
- Find the expectation of the hidden variables, given the current estimate of the parameters of the explicit problem and the sample data.
- Use the expected values of the hidden variables to find an improved estimate of the explicit problem parameters.
- Cycle steps 2, 3, and 4 until the algorithm arrives at a fixed point.

This process is illustrated in figure 4. Todd K. Moon's review paper 1996 IEEE Signal Processing Magazine *The Expectation-Maximization Algorithm* contains an excellent account [8]. It is important to understand that this "algorithm" does not fully specify any particular estimation procedure and so may more appropriately be viewed as a design technique for an algorithm. Moreover, it does not inform the algorithm designer about what hidden variables she should imagine to allow a tractable maximum likelihood solution.

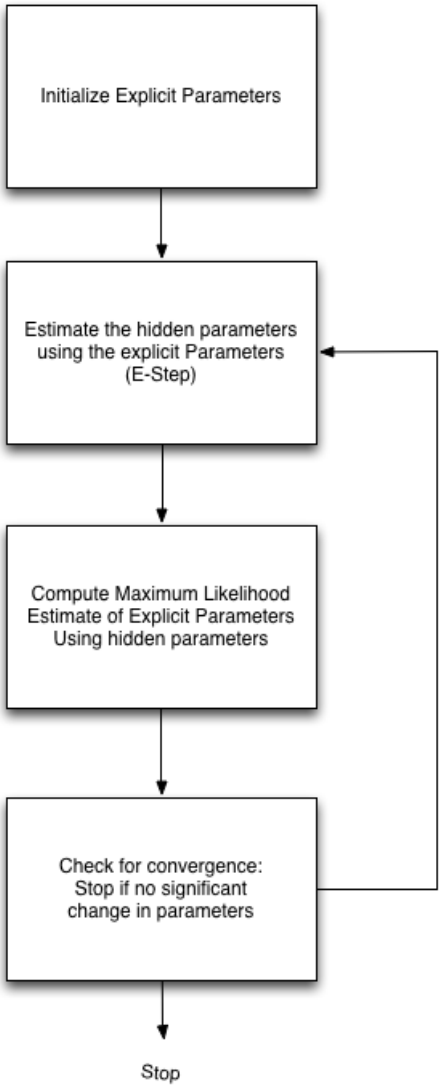


Figure 3: **The EM algorithm.** After initialization, the E-Step and the M-step are alternated until the parameter estimate has converged by some criterion. Examples include a very small increase in the likelihood of the data given the model parameters, or very small changes in the estimate of the model parameters.

## EM Algorithm Theorems

Leonard Baum and his colleagues presented EM in a fairly general form in the late 1960's [5, 4, 3]. But it remained for Dempster, Laird, and Rubin [2] to develop the applicability of this algorithm beyond the Hidden Markov Model of Baum et al.

At a very high level Baum offered the following theorem. let:

$$P(\lambda) = \int_X p(x, \lambda) d\mu(x) \tag{1}$$

and let the auxiliary function $Q$ be defined as:

$$Q(\lambda, \lambda') = \int_X p(x, \lambda) log\, p(x, \lambda') d\mu(x) \tag{2}$$

then $Q(\lambda, \lambda') \geqslant Q(\lambda, \lambda) \implies P(\lambda') \geqslant P(\lambda)$.

where $\lambda$ is the set of parameters of the mixture. Thus we can substitute the maximization of $Q(\lambda, \lambda')$ with respect to $\lambda'$ as a proxy for $P(\lambda')$, which we will see is advantageous for important problems. In much of the literature on the EM Algorithm, the so-called *Q-function* is used to designate the auxiliary function for the specific likelihood to be maximized. Gaussian Mixture Models are key to many modern pattern recognition applications and are used below to sharpen the abstract theorem statement by Baum given above.

Redner and Walker studied this in detail in their widely sited paper *Mixture Densities, Maximum Likelihood and the EM Algorithm* which offers the first general proofs on estimating mixtures of exponential family distributions using EM [10] and which summarizes the the EM algorithm as follows:

> "Suppose that one has a measure space $\mathscr{Y}$ of "complete data" and a measurable map $y \to x(y)$ of $\mathscr{Y}$ to a measure space $\mathscr{X}$ of "incomplete data." Let $f(y|\Phi)$ be a member of a parametric family of probability density functions defined on $\mathscr{Y}$ for $\Phi \in \Phi$, and suppose that $g(x|\Phi)$ is a probability density function on $\mathscr{X}$ induced by $f(y|\Phi)$. For a given $x \in \mathscr{X}$, the purpose of the EM algorithm is to maximize the incomplete data log-likelihood $L(\Phi) = log\, g(x|\Phi)$ over $\Phi \in \Omega$ by exploiting the relationship between $f(\mathbf{y}|\Phi)$ and $g(x|\Phi)$. It is intended especially for applications in which the maximization of the complete data log-likelihood $log\, f(\mathbf{y}|\Phi)$ over $\Phi \in \Phi$ is particularly easy.
>
> For $\mathbf{x} \in \mathscr{X}$, set $\mathscr{Y}(\mathbf{x}) = \{\mathbf{y} : \mathbf{x}(\mathbf{y}) = \mathbf{x}\}$. The conditional density $k(\mathbf{y}|\mathbf{x}, \Phi)$ on $\mathscr{Y}(\mathbf{x})$ is given by $f(\mathbf{y}|\Phi) = k(\mathbf{y}|\mathbf{x}, \Phi) g(\mathbf{x}|\Phi)$. For $\Phi^{'}$ in $\Omega$, one then
>
> $$L(\Phi) = Q(\Phi|\Phi^{'}) - H(\Phi|\Phi^{'}) \tag{3}$$
>
> Where $Q(\Phi|\Phi^{'}) = E(log\, f(\mathbf{y}|\Phi)|\mathbf{x}, \Phi^{'})$ and $H(\Phi|\Phi^{'}) = E(log\, k(\mathbf{y}|\mathbf{x}, \Phi)|\mathbf{x}, \Phi^{'})$. The general EM algorithm of Dempster, Laird, and Rubin [DempsterLairdRubin1977] is the following: Given a current approximation $\Phi^c$ of a maximizer of $L(\Phi)$, obtain a next approximation $\Phi^+$ as follows:
>
> E-Step. Determine $Q(\Phi|\Phi^c)$.
>
> M-Step. Chose $\Phi^+ \in arg\, max_{\Phi \in \Omega} Q(\Phi|\Phi^c)$.
>
> ...
>
> From this general description, it is not clear that the EM algorithm even deserves to be called an algorithm. However, as we indicated above, the EM

algorithm is used most often in applications which permit the easy maximization of $log\, f(y)$ over $\Phi \in \Omega$. In such applications, the M-step maximization of $Q(\Phi|\Phi^c)$ over $\Phi \in \Omega$ is usually carried out with corresponding ease. In fact, as one sees in the sequel, the E-step and the M-step are usually combined into one very easily implemented step in most applications involving mixture density estimation problems."

It is certainly not clear how to proceed from this high level statement of the EM Algorithm, and additional insights will be needed. Another approachable treatment of the EM Algorithm is found in Jeff Bilmes' widely cited *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models* [6]. Zoubin Ghahramani and Michael Jordan also give an approachable treatment to using EM to estimate GMM's when there is missing data: *Learning from incomplete data* [7]. Beyond these excellent resources, it seems that some additional simplification and clarification of the EM Algorithm for GMM's would be helpful to many readers who want to understand and apply the method. We address this below.

## Complete data formulation of the GMM

In *complete data* problem formulations, we are given a set of observations and its distribution, here the GMM defined by equations (3), and (4) so

$$g(x, \Phi) \sim \sum_{j=1}^{C} \frac{\alpha_j}{\sigma_j \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_j}{\sigma_j}\right)^2} \tag{4}$$

where $\Phi$ is the parameter set $\{A, M, \Sigma\}$ of the mixture density. $X$ (our observed sample of data points) is called the *incomplete data*, here the data $\mathbf{x} = \{x_i\}_{i=1}^N$ together with the parameters $\Phi$. This is intractable for most mixture problems using basic maximum likelihood estimation. To *complete* the data we imagine that there are a set of hidden parameters $Y$, which, if we knew their values, would make the maximum likelihood estimation problem tractable. $Z = (X, Y)$ then is the *complete data*. Further we posit that there is a joint probability density of the complete data as Redner and Walker did above:

$$f(\mathbf{y}|\Phi) = k(\mathbf{y}|\mathbf{x}, \Phi) g(\mathbf{x}|\Phi) \tag{5}$$

How does this operate in a GMM? For the Gaussian mixture estimation we suppose that that hidden variables are $Y = \{\mathbf{y}_i\}_{i=1}^N$ where

$$y_{ij} = \begin{cases} 1 & if\, x_i \sim p(x|\alpha_j, \mu_j, \sigma_j) \\ 0 & otherwise \end{cases} \tag{6}$$

In the literature, typically, the complete data emerge *dei ex machina*, which recalls Redner and Walker's statement that: "From this general description, it is not clear that the EM

algorithm even deserves to be called an algorithm. However, as we indicated above, the EM algorithm is used most often in applications which permit the easy maximization of $log\, f(y|\Phi)$ over $\Phi \in \Omega$." Thus mathematical intuition is the key for designing hidden variables for the complete data representation of the problem. So the key intuitions that makes the complete data actually work is extrinsic to the EM Algorithm itself.

Most authors on GMM's simply assert that: if we had a matrix of binary variables that specified the Gaussian component whence each point came, then we could cleanly decompose the problem into a tractable complete likelihood function as follows:

$$l_c(\Phi, X, Y) = \sum_{i=1}^{N}\sum_{j=1}^{C} z_{ij} log\left[p(x_i|y_{ij}, \Phi)\right] p(y_{ij}|\Phi) \tag{7}$$

If we were given a set of binary dummy variables

$$z_{ij} = \begin{cases} 1 & if\, point\, i\, was\, generated\, by\, component\, j \\ 0 & otherwise \end{cases}$$

We can use the fact that zero exponents are always 1 for for non-zero bases we write the joint density as:

$$p(x_i|A, M, \Sigma, Y) = \prod_{j=1}^{C}\left(\frac{\alpha_j}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu_j}{\sigma_j}\right)^2}\right)^{y_{ij}}$$

The likelihood function then becomes

$$L_c\left(A, M, \Sigma|X, Y\right) = \prod_{i=1}^{N} p(x_i|A, M, \Sigma, Y)$$

$$L_c\left(A, M, \Sigma|X, Y\right) = \prod_{i=1}^{N}\prod_{j=1}^{C}\left(\frac{\alpha_j}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu_j}{\sigma_j}\right)^2}\right)^{y_{ij}}$$

$$log\left(L_c\left(A, M, \Sigma|X, Y\right)\right) = log\left(\prod_{i=1}^{N}\prod_{j=1}^{C}\left(\frac{\alpha_j}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu_j}{\sigma_j}\right)^2}\right)^{y_{ij}}\right)$$

$$l_c\left(A, M, \Sigma|X, Y\right) = \sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} log\left(\frac{\alpha_j}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu_j}{\sigma_j}\right)^2}\right)$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} log\left(\frac{\alpha_j}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu_j}{\sigma_j}\right)^2}\right)$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\left(log\left(\alpha_j\right) - log\left(\sigma_j\right) - \frac{1}{2}log\left(2\pi\right) - \frac{1}{2}\left(\frac{x_i-\mu_j}{\sigma_j}\right)^2\right)$$

Since $\mathbf{y}$ is unknown we can proceed by using its expectation:

$$Q(\Phi|\Phi_k) = E[l_c(\Phi, X, Y)|X, \Phi_k)] \tag{8}$$

and

$$\Phi_{k+1} = argmax_{\Phi}\left[Q(\Phi|\Phi_k)\right] \tag{9}$$

For the GMM this becomes:

$$y_{ij}^k = E(y_{ij}|\mathbf{x}, \Phi^k) = \frac{\alpha_j^{(k-1)} p\left(x_i|\mu_j^{k-1}, \sigma_j^{k-1}\right)}{\sum_{l=1}^{C} \alpha_l^{k-1} p\left(x_i|\mu_l^{k-1}, \sigma_l^{k-1}\right)} \tag{10}$$

and this is clearly the probability that component $j$ generated point $i$ by an elementary application of Bayes' Theorem. With these expected values in hand we can proceed to finding the estimates of the mixture parameters. In order to solve for the mixture weights $\alpha_j$ we must use a Lagrange multiplier to constrain their sum to unity. So we must optimize

$$\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\left(log\left(\alpha_j\right) - log\left(\sigma_j\right) - \frac{1}{2}log\left(2\pi\right) - \frac{1}{2}\left(\frac{x_i - \mu_j}{\sigma_j}\right)^2\right) + \lambda\left(1 - \sum_{k=1}^{C}\alpha_k\right)$$

With this we can proceed by the usual partial derivatives to provide a system of equations to solve. It happens that these work out well and can lead us in easy stages to closed form solutions for the parameters of the mixture as follows:

$$\frac{\partial l_c}{\partial \alpha_j} = \frac{1}{\alpha_j}\sum_{i=1}^{N} y_{ij} - \lambda = 0$$

$$\sum_{i=1}^{N} y_{ij} = \alpha_j \lambda$$

Now, summing both sides over the number of Gaussian components will yield a pleasing simplification owing to the stochastic constraints on the $y_{ij}{}'s$ and $\alpha_j{}'s$ as follows:

$$\sum_{j=1}^{C}\sum_{i=1}^{N} y_{ij} = \lambda\sum_{j=1}^{C}\alpha_j$$

which becomes:

$$N = \lambda \cdot 1$$

whence by substitution we obtain:

$$\sum_{i=1}^{N} y_{ij} = \alpha_j N$$

and

$$\alpha_j = \frac{1}{N}\sum_{i=1}^{N} y_{ij}$$

These being the row sums of the conditional class membership matrix given by the $Y_{ij}{}'s$ which is just the probability of the observed data points $x_i$ being generated by the

$j^{th}$ mixture component. So these are the estimates of the mixture weights $\alpha_j$ that we wanted.

Next we seek to find the component means and variances. The choice here of the Gaussian Mixture Model will lead us to closed form solutions for these quantities. As we have noted, some other important exponential family distributions can be solved in an analogous way so this argument extends beyond GMM's. Returning to $l_c(A, M, \Sigma|X, Y)$ now having solved $A = \{\alpha_j\}$ above we have

$$l_c(M, \Sigma|A, X, Y) = \sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\left(log(\alpha_j) - log(\sigma_j) - \frac{1}{2}log(2\pi) - \frac{1}{2}\left(\frac{x_i - \mu_j}{\sigma_j}\right)^2\right)$$

$$\frac{\partial l_c}{\partial \mu_j} = \sum_{i=1}^{N} y_{ij}\left(-\frac{1}{2}\frac{\partial}{\partial \mu_j}\left(\frac{x_i - \mu_j}{\sigma_j}\right)^2\right) = 0$$

$$0 = \sum_{i=1}^{N} y_{ij}\left(\frac{x_i - \mu_j}{\sigma_j}\right)$$

$$0 = \sum_{i=1}^{N}\left(\frac{1}{\sigma_j}y_{ij}x_i - \frac{1}{\sigma_j}y_{ij}\mu_j\right)$$

$$0 = \sum_{i=1}^{N} y_{ij}x_i - \sum_{i=1}^{N} y_{ij}\mu_j$$

$$\mu_j \sum_{i=1}^{N} y_{ij} = \sum_{i=1}^{N} y_{ij}x_i$$

$$\mu_j N\alpha_j = \sum_{i=1}^{N} y_{ij}x_i$$

$$\mu_j = \frac{\sum_{i=1}^{N} y_{ij}x_i}{N\alpha_j}$$

Finally we seek the standard deviations $\sigma_j$.

$$l_c(\Sigma|M, A, X, Y) = \sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\left(log(\alpha_j) - log(\sigma_j) - \frac{1}{2}log(2\pi) - \frac{1}{2}\left(\frac{x_i - \mu_j}{\sigma_j}\right)^2\right)$$

$$\frac{\partial l_c}{\partial \sigma_j} = \sum_{i=1}^{N} y_{ij}\left(-\frac{1}{\sigma_j} + \left(\frac{(x_i - \mu_j)^2}{\sigma_j^3}\right)\right) = 0$$

$$\sum_{i=1}^{N} y_{ij}\left(-1 + \left(\frac{(x_i - \mu_j)^2}{\sigma_j^2}\right)\right) = 0$$

$$\sum_{i=1}^{N} y_{ij}\left(-\sigma_j^2 + (x_i - \mu_j)^2\right) = 0$$

$$\sigma_j^2 \sum_{i=1}^{N} y_{ij} = \sum_{i=1}^{N} y_{ij} \left(x_i - \mu_j\right)^2$$

$$\sigma_j^2 = \frac{\sum_{i=1}^{N} y_{ij} \left(x_i - \mu_j\right)^2}{\sum_{i=1}^{N} y_{ij}}$$

$$= \frac{\sum_{i=1}^{N} y_{ij} \left(x_i - \mu_j\right)^2}{N\alpha_j}$$

Of course, the initial $\alpha_j^{(0)}$, $\mu_j^{(0)}$,and $\sigma_j^{(0)}$, must be supplied below to seed the iteration, perhaps by choosing an approximately uniform prior across the range of the data sample $x$, or some other approximate method, to initialize the EM iteration embodied in equations 11 through 15. This done, we can compute the first iteration of $y_{ij}^{(1)}$. Then having the first estimate of the complete parameters $Y$, the mixture weights $\alpha_j$ the component means $\mu_j$ and the standard deviations $\sigma_j$, in closed form we can write the GMM re-estimation equations as:

$$y_{ij}^{(k)} = \frac{\alpha_j^{(k-1)} p\left(x_i | \mu_j^{(k-1)}, \sigma_j^{(k-1)}\right)}{\sum_{l=1}^{C} \alpha_l^{(k-1)} p\left(x_i | \mu_l^{(k-1)}, \sigma_l^{(k-1)}\right)} \tag{11}$$

For the specific case of the Gaussian Mixture Model being developed here, this is:

$$y_{ij}^{(k)} = \frac{\frac{\alpha_j^{(k)}}{\sigma_j^{(k)}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i - \mu_i^{(k)}}{\sigma_i^{(k)}}\right)^2}}{\sum_{l=1}^{C} \frac{\alpha_l^{(k)}}{\sigma_l^{(k)}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i - \mu_l^{(k)}}{\sigma_l^{(k)}}\right)^2}} \tag{12}$$

Then the mixture parameters are re-estimated as follows:

$$\alpha_j^{(k)} = \frac{1}{N} \sum_{i=1}^{N} y_{ij}^{(k)} \tag{13}$$

$$\mu_j^{(k)} = \frac{\sum_{i=1}^{N} y_{ij}^{(k)} x_i}{N\alpha_j^{(k)}} \tag{14}$$

$$\sigma_j^{(k)} = \sqrt{\frac{\sum_{i=1}^{N} y_{ij}^{(k)} (x_i - \mu_j^{(k)})^2}{N\alpha_j^{(k)}}} \tag{15}$$

So what's going on here? It can be seen that we have constructed an iterative map of the parameter space onto itself, which Redner and Walker showed to be a contractive mapping, which guarantees convergence to a fixed point under appropriate hypothesis. A streamlined statement of this theorem is as follows.

**Theorem**: If the Fisher information matrix $I(\Phi)$ is positive definite at

$\Phi^* = (\alpha_1^*, \ldots, \alpha_m^*, \phi_1^*, \ldots, \phi_m^*)$ is such that $\alpha_i^* > 0$ for $1 \leq i \leq m$. For all $\Phi^{(0)}$ in $\Omega$, denote by $\{\Phi^{(j)}\}_{j=0,1,2,\ldots}$, the sequence $\Omega$ by EM iteration. Then there is a constant $0 \leq \lambda < 1$ for which

$$\left|\Phi^{(j+1)} - \Phi^N\right| \leq \lambda \left|\Phi^{(j)} - \Phi^N\right|$$

whenever $\Phi^{(0)}$ is sufficiently near $\Phi^N$. The theorems of Baum et al. showed that this fixed point is a local maximizer of the likelihood function of the explicit distribution. The interested reader is referred to the bibliography which contains excellent, though sometimes somewhat opaque treatments of the topics presented here.

## Generalization to multivariate GMM

Other mixture distributions in the Exponential Family may be estimated with structurally similar formulae, as is discussed in Redner and Walker. Of course this generalizes readily to the multivariate case as:

$$y_{ij}^{(k)} = E(y_{ij}|\mathbf{x}, \Phi^{(k)}) = \frac{\alpha_j^{(k-1)} \left|\mathbf{\Sigma}_j^{(k-1)}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}\left(\mathbf{x}_i - \mathbf{m}_j^{(k-1)}\right)^T \left(\mathbf{x}_i - \mathbf{m}_j^{(k-1)}\right)}}{\sum_{l=1}^{C} \alpha_l^{(k-1)} \left|\mathbf{\Sigma}_l^{(k-1)}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}\left(\mathbf{x}_i - \mathbf{m}_l^{(k-1)}\right)^T \left(\mathbf{x}_i - \mathbf{m}_l^{(k-1)}\right)}} \tag{16}$$

$$\alpha_j^{(k)} = \frac{1}{N} \sum_{i=1}^{N} y_{ij}^{(k-1)} \tag{17}$$

$$\mathbf{m}_j^{(k)} = \frac{\sum_{i=1}^{N} y_{ij}^{(k)} \mathbf{x}_i}{N\alpha_j^k} \tag{18}$$

$$\Sigma_j^{(k)} = \frac{\sum_{i=1}^{N} y_{ij}^{(k)} \left(\mathbf{x}_i - \mathbf{m}_j^{(k-1)}\right) \left(\mathbf{x}_i - \mathbf{m}_j^{(k-1)}\right)^T}{N\alpha_j^{(k)}} \tag{19}$$

With this comes certain numerical issues to manage, as sometimes particular covariance matrices tend towards singularity. This must be detected and dealt with in the computer codes that execute the estimation process.

## Summary of GMM Complete Data Concepts

As we have mentioned before, Gaussian Mixture Models are an example of an EM algorithm, and the EM algorithm is a design pattern specifying how to construct a more detailed algorithm to solve a complete data problem. The way in which GMMs satisfy the conditions of a complete data problem is illustrated in the table of correspondences below:

| Gaussian Mixture Data | instantiates | Complete Data Problem |
|---|---|---|
| parameter set $\Phi = \{A, M, \Sigma\}$ | $\longrightarrow$ | incomplete, or explicit parameters |
| hidden parameters $Y = \{y_{ij}\}$ | $\longrightarrow$ | hidden parameters that make the MLE tractable |
| prior parameter set $\{A_0, M_0, \Sigma_0\}$ that comes from extrinsic sources | $\longrightarrow$ | initial "guess" or initial priors |
| $\mathbf{x} = \{x_i\}_{i=1}^{N}$ | $\longrightarrow$ | observed data set |
| $X = \{\mathbf{x}, \Phi\}$ | $\longrightarrow$ | incomplete data |
| $\{X, Y\}$ | $\longrightarrow$ | complete data |

Note that the only difference between the "complete data" and the "incomplete data" comes from the choice of parameters, not $X$, even though $X$ is what we would conventionally think of as being "data". This is due to weakness in the underlying terminology, which does not lend itself to naturally conveying what the actual distinction between the terms is. That said, we can show how the parts of the GMM estimation procedure match up with the conceptual components of the EM algorithm as given in Todd Moon's 1996 paper:

| Gaussian Mixture Model | instantiates | EM Algorithm |
|---|---|---|
| parameter set Y=$\{A_2, M_2, \Sigma_2\}$ that comes from a guess | $\longrightarrow$ | Choosing an initial parameter $\theta^{[0]}$ |
| $Q(\Phi\|\Phi') = E(log\, f(Y\|\Phi)\,\|x, \Phi')$ | $\longrightarrow$ | Estimate unobserved data using $\theta^{[k]}$ |
| $\Phi_{k+1} = argmax_{\Phi}[Q(\Phi\|\Phi_k)]$ | $\longrightarrow$ | Compute maximum likelihood estimate of parameter $\theta^{[k+1]}$ using estimated data |
| Is $\Phi_{k+1} = \Phi_k$? | $\longrightarrow$ | Check for parameter fixed point |

# GMM Example Program in Matlab/Octave

```
#!/opt/local/bin/octave
%-------------------------------------------------------------------
% Name:     gmm.m
% Language GNU Octave
%-------------------------------------------------------------------
% Purpose: estimate the parameters of a Gaussian mixture model
%          and plot the stages of the process
%-------------------------------------------------------------------
% Author:  Vince Stanford
% Date:    July 17, 2013
% License: We offer this program under:
%          Creative Commons Attribution 4.0 International
%-------------------------------------------------------------------
%          This program is not warranted to be fit for any
%          particular purpose, nor to be free from defects
%          in design or implementation. You are welcome to
%          use it, extend it, or modify it as you please.
%          But you do this at your own risk.
%-------------------------------------------------------------------
% Note: This program is written for exposition, and is not
%          particularly performant. Further vectorization
%          could yield substantially enhanced performance.
%-------------------------------------------------------------------
ITERATION_LIMIT=25
%-----------------------------
% construct random test data
%-----------------------------
groups=3
G=1:groups;
sampCount(G)=1000
xCount=sum(sampCount)
T=1:xCount;
x=randn(xCount,1);

t=1;
for g=1:groups
  for i=1:sampCount(g)
    x(t++)=(x(t)+2*g)*g;
  endfor
endfor

xBinCount=xCount/10
[xCounts,xBins]=hist(x,xBinCount);
plot(xBins,xCounts)

xMean=mean(x)
xStd=std(x)
range=3*xStd

mid=sum(G)/groups
m=xMean+xStd*(G-mid)
s=xStd^2./G
h=zeros(xCount,groups);
p=ones(groups,1)/groups;
d=zeros(groups,1);

%-----------------------------------------
% compute the expected value of the
% hidden variables h(i,j), which
% represent the probability that sample
% point i belongs to component j.
% (this should be vectorized for speed
% but is instead written to maximize
% structural clarity vis-a-vis the
% published computing formulae)
%-----------------------------------------
for iteration = 1:ITERATION_LIMIT
```

```
  iteration
  for l=1:groups
    h(T,l)=p(l)*normpdf(x(T),m(l),sqrt(s(l)));
  endfor
  totalLikelihood=sum(h');
  for t=1:xCount
    norm=1/totalLikelihood(t);
    for l=1:groups
      h(t,l)=norm*h(t,l);
    endfor
  endfor
  p=sum(h);

  %-------------------------------------------------------
  % maximize the incomplete variables using
  % the expected values of the complete variables
  %-------------------------------------------------------
  for j = 1:groups
    sSum=0;
    mSum=0;
    for t = 1:xCount
      mSum+=h(t,j)*x(t);
      sSum+=h(t,j)*((x(t)-m(j))^2);
    endfor
    s(j)=sSum/p(j);
    m(j)=mSum/p(j);
  endfor
  p=p/xCount
  m
  s
  %-------------------------------------------------------
  % setup and plot the theoretical normal vs observed
  %-------------------------------------------------------
  figure(1);
  xProbs=xCounts/sum(xCounts);
  deltaZ=mean(diff(xBins));

  pdfMixZProbs=zeros(xBinCount,1);

  for i=1:xBinCount
    for j=1:groups
      pdfMixZProbs(i)=pdfMixZProbs(i)+p(j)*normpdf(xBins(i),m(j),sqrt(s(j)));
    endfor
  endfor
  pdfMixZProbs=pdfMixZProbs*deltaZ;

  % Single quote the following strings to run in Matlab.
  plot(xBins,xProbs,"+","markersize",4,xBins,pdfMixZProbs,"linewidth",5);
  title("Observed and Estimated Frequency Densities");
  xlabel("Standard Deviations");
  ylabel("p(x(t))")

  drawnow();
  if (iteration==1)
    input("press any key to iterate");
  endif
endfor

input("press any key to exit");

exit(0)
```

# References


[1] Ankur Moitra Adam Tauman Kalai and Gregory Valiant. Efficiently learning mixtures of two gaussians. 2010.

[2] N.M. Laird A.P. Dempster and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[3] Leonard E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In Oved Shisha, editor, *Inequalities III: Proceedings of the Third Symposium on Inequalities*, pages 1–8, University of California, Los Angeles, 1972. Academic Press.

[4] Leonard E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, May 1967.

[5] Leonard E. Baum and Ted Petrie. Statistical inference for probabilistic functions of finite-state Markov chains. *Annals of Mathematical Statistics*, 37(6):1554–1563, Dec 1966.

[6] Jeff Bilmes. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models.* International Computer Science Institute, Berkley, California, Apr 1998.

[7] Zoubin Ghahramani and Michael Jordan. Learning from incomplete data a.i. memo no. 1599. *Massachusetts Institute of Technology Artificial Intelligence Laboratory A.I. Memos*, December 1994.

[8] Todd K Moon. The expectation maximization algorithm. *IEEE Signal Processing Magazine*, pages 47–60, 1996.

[9] Karl Pearson. Contributions to the mathematical teheory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 186:343–414, 1895.

[10] Richard A. Redner and Homer F. Walker. Mixture densities, maximimum likelihood and the em algorithm. *SIAM Review*, 26(2):195–239, April 1984.