Expectation Maximization Algorithm

Yixiao Feng, Yukun Tan

Abstract

Expectationmaximization (EM), which is an iterative method for the concave function, attempts to find the maximum likelihood estimator (MLE) of a parameter θ of a parametric probability distribution. The EM algorithm consists of two steps: E step and M step. A function for the expectation of the log likelihood based on the initial guess of the parameters was created in the E step pf the EM algorithm. The expected log-likelihood is maximized at each iteration, which computes parameters found on the E step. New parameter θ are then used to determine the distribution of the latent variables in the next E step. The EM algorithm is used to find (locally) maximum likelihood parameters of a statistical model in cases where the equations cannot be solved directly. Gaussian mixture models (GMMs), and estimating hidden Markov models (HMMs) are the most popular applications of EM. Practically, EM does not guarantee to find the θ that maximizes $p(y|\theta)$. The report is composed of basic steps of EM algorithm. Also, several useful example and simulations were analysis in the following sections. Some theoretical proof about EM algorithm like Jensens Inequality with the monotonicity of the EM algorithm were also analyzed in the report.

Index Terms

Expectation Maximization, MLE, GMM, KullbackLeibler divergence

I. Introduction

THE The EM algorithm was introduced by Ceppellini et al. [1] in the context of gene frequency estimation, the EM algorithm was analyzed more generally by Hartley [2] and by Baum et al.3 in the context of hidden Markov models, where it is commonly known as the Baum-Welch algorithm. The standard reference on the EM algorithm and its convergence is Dempster et al. [4]

The EM algorithm has become a popular tool in statistical estimation problems involving incomplete data, or in problems which can be posed in a similar form, such as mixture estimation [5, 6]. The parameters of a model are what the EM algorithm tries to find by using MLE. Practically, these models come up with some hidden values and some known distributions, which means that either missing data exist, or the model can be formulated more simply by assuming the existence of further unobserved data

points. For example, a mixture model can be described more simply by assuming that each observed data point has a corresponding unobserved data point, or latent variable, specifying the mixture component to which each data point belongs.

The problem for a statistical model with some hidden data is that to solve a derivatives of the likelihood function with some unknown variables is impossible. Instead, EM algorithm provide an alternative way, which substituting one set of equations into the other produces an unsolvable equation because the solution to the parameters requires the values of the latent variables and vice versa. The importance of EM could be reflected on solving incomplete data. Usually, the only data available for training a probabilistic model are incomplete. Missing values can occur, for example, in medical diagnosis, where patient histories generally include results from a limited battery of tests. Alternatively, in gene expression clustering, incomplete data arise from the intentional omission of gene-to-cluster assignments in the probabilistic model. The EM algorithm enables parameter estimation in probabilistic models with incomplete data.

In order to implement EM algorithm: 1) some observed data y; 2) a parametric density $p(y|\theta)$; 3) description of the complete data x that you wish you had; 4) and the parametric density $p(x|\theta)$, which should be given.

Other problems in which the expectation maximization algorithm plays a prominent role include learning profiles of protein domains [8] and RNA families [9], discovery of transcriptional modules [10], tests of linkage disequilibrium [11], protein identification [12] and medical imaging [13].

EM converges to a local optimum by the nature of the concave function, that is to say, the global minimum can not be obtained by EM algorithm. Therefore, several initial guess were made to find the best convergence for EM. It is possible that it can be arbitrarily poor in high dimensions and there can be an exponential number of local optima. Hence, a need exists for alternative methods for guaranteed learning, especially in the high-dimensional setting. Algorithms with guarantees for learning can be derived for a number of important models such as mixture models, HMMs etc. For these spectral methods, no spurious local optima occur, and the true parameters can be consistently estimated under some regularity conditions.

The rest of the report consists of 4 sections. We will cover the methodology of the EM algorithm in section II with some examples. And the simulations with analysis will be given in section III. We also provide the general advantages and disadvantages of EM compared to other options for maximizing the likelihood, like the NewtonRaphson method. Finally, we summarized the report and make a conclusion about the EM algorithm on section IV.

II. METHODOLOGY

A. Material

In order to implement EM algorithm: 1) some observed data y; 2) a parametric density $p(y|\theta)$; 3) description of the complete data x that you wish you had; 4) and the parametric density $p(x|\theta)$, which should be given. We assume that the complete data can be modeled as a continuous (i.e. The treatment of discrete random vectors is a straightforward special case of the continuous treatment: one only needs to replace the probability density function with probability mass function and integral with summation) random vector X with density $p(x|\theta)$, where $\theta \in \Omega$ for some set Ω. X is not observed directly, and what you can observe is the realization y of the random vector Y, which depends on X. For example, Y is the mean or the first component of the random vector X. If you have the observation y, then using the MLE to find the maximum of parameter θ:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \log p(y|\theta) \tag{1}$$

And the $\log$ function is monotonically increasing function, which means we can use $\log$ function as measurement instead of measuring $p(y|\theta)$. However, for some problems it is difficult to solve this equation. Then thats why using EM: an initial guess was made for predicting the whole complete data X. And we want to solve for the θ that maximizes the (expected) log-likelihood of X. Once we have an estimate for θ, we can make a better guess about the complete data X, and iterate.

B. Implementation

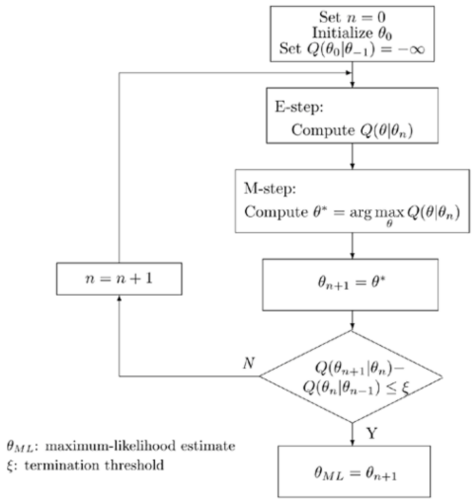

Fig. 1: Diagram of EM algorithm

To implement EM algorithm, you have to make an initial guess about the parameter of the model. The initial guess can be chosen at any values, which will converge to a final value as desired.

After choosing a initial value about the parameter of the model (pretend that the parameter of the model is correct for now), we should be able to formulate the conditional probability distribution $p(x|y, \theta^{(m)})$ for the complete data x. The distribution for the complete data x form the conditional expected log-likelihood function, which is called the Q-function.

Our goal is to maximize the Q function with respect to the parameter θ. And the result after the maximization is the new parameter θ. Using the new parameter for the model and substitute the new parameter θ for formulating the new the conditional probability distribution $p(x|y, \theta^{(m)})$.

The desired parameter of θ will be returned depends on the criterion set on the algorithm. Figure 1 illustrates how the EM algorithm works as well as Table 1 shows the all of the steps that EM algorithm will go through. The EM algorithm is divided into E step, which is categorized into the step 2 and step 3, and M step that consists of the step 4:

E-step: Given the estimate from the previous iteration $\theta^{(m)}$, compute the Q-function:

$$\begin{aligned} Q(\theta|\theta^{(m)}) &= \int_{\mathcal{X}(y)} \log p(x|\theta) p(x|y, \theta^{(m)}) dx \\ &= E_{X|y,\theta^{(m)}}[\log p(X|\theta)] \end{aligned} \tag{2}$$

M-step: The $(m+1)$th guess of θ is:

$$\theta^{(m+1)} = \arg\max_{\theta \in \Omega} Q(\theta|\theta^{(m)}) \tag{3}$$

As mentioned previously, the EM Algorithm is only guaranteed to never get worse in each iteration. Hopefully, it will find the global peak in the likelihood $p(y|\theta)$, howeve, if the likelihood function $p(y|\theta)$ has multiple peaks, EM will not necessarily find the global maximum of the likelihood. Therefore, several initial guess are chosen for EM algorithm, and then choose the one with the largest likelihood as the final guess for θ.

Failure of EM algorithm: singularities of the log-likelihood function. If we have to learn a Gaussian mixture model with 10 components, it may decide that the most likely solution is for one of the Gaussians to only have one data point assigned to it, with the bad result that the Gaussian is estimated as having zero covariance [14]. Imposing some prior information on the solution for θ is a solution to such degeneracies. One approach would be to restrict the set of possible θ. Such a restriction is equivalent to putting a

TABLE I: EM algorithm

Step 1 Make an initial estimate $\theta(m)$ for θ
Step 2 Given the observed data y and pretending for the moment that your currentguess is correct, formulate $p(x\|y,\theta(m))$ for the complete data x
Step 3 Find Q-function
Step 4 Find the θ that maximizes the Q-function
Step 5 Go to step 2 and iterate

uniform prior probability over the restricted set. More generally, we can impose any prior $p(\theta)$, and then modify EM to maximize the posterior rather than the likelihood:

$$\hat{\theta}_{MAP} = \arg\max_{\theta\in\Omega} \log p(\theta|y) = \arg\max_{\theta\in\Omega} (\log p(y|\theta) + \log p(\theta)) \tag{4}$$

Maximum a posteriori (MAP) estimation (extended form EM):

E-step: Given the estimate from the previous iteration $\theta^{(m)}$, compute the conditional expectation:

$$Q(\theta|\theta^{(m)}) = E_{X|y,\theta^{(m)}}[\log p(X|\theta)] \tag{5}$$

M-step: Maximize $Q(\theta|\theta^{(m)}) + \log p(\theta)$:

$$\theta^{(m+1)} = \arg\max_{\theta\in\Omega} (Q(\theta|\theta^{(m)}) + \log p(\theta)) \tag{6}$$

C. Specifying Data

Theoretically, the complete data X have to satisfy the Markov relationship $\theta \to X \to Y$:

$$p(y|x,\theta) = p(y|x) \tag{7}$$

Applications like GMM or HMM, the observed data X plus some hidden data Z is equivalent to the the complete data X. And we can represent this as $Y = T(X)$. In general, the Q function of the missing data problem is an integral over the domain of the hidden value Z (rather than over the domain of X):

$$\begin{aligned} Q(\theta|\theta^{(m)}) &= \int_{\mathcal{X}} \log p(x|\theta) p(x|y,\theta^{(m)}) dx \\ &= \int_{\mathcal{X}} \log p(y,z|\theta) p(y,z|y,\theta^{(m)}) dx \\ &= \int_{\mathcal{Z}} \log p(y,z|\theta) p(z|y,\theta^{(m)}) dz \\ &= E_{Z|y,\theta^{(m)}}[\log p(y,Z|\theta)] \end{aligned} \tag{8}$$

If we can denote the complete data X as a random vector: $X = [X_1 X_2 \cdots X_n]^T$ and the observed data y is only sampled from X, then: Suppose Markov relationship holds for all $i = 1, \cdots, n$, that is:

$$p(y_i|x, y_1, \cdots, y_{i-1}, y_{i+1}, \cdots, y_n, \theta) = p(y_i|x_i) \tag{9}$$

then

$$Q(\theta|\theta^{(m)}) = \sum_{i=1}^{n} Q_i(\theta|\theta^{(m)}), \tag{10}$$

where

$$Q_i(\theta|\theta^{(m)}) = E_{X_i|y_i,\theta^{(m)}}[\log p(X_i|\theta)], i = 1, \cdots, n. \tag{11}$$

D. Theoretical Analysis

What we care about the EM algorithm is that how good of the estimates produced by it? And we want to show that the Q-function provides a lower bound to the true log-likelihood function. Also, we want to show that the parameters we get from each iteration will never getting worse, which is also called the monotonicity of the EM algorithm.

1) Convergence: The M-step of EM algorithm guarantees that:

$$\theta^{(m+1)} = \arg\max_{\theta \in \Omega} Q(\theta|\theta^{(m)}) \tag{12}$$

which gives us:

$$Q(\theta^{(m+1)}|\theta^{(m)}) \geq Q(\theta^{(m)}|\theta^{(m)}) \tag{13}$$

Therefore, we wish to conclude that

$$\ell(\theta^{(m+1)}) \geq \ell(\theta^{(m)}) \tag{14}$$

The monotonicity alone cannot guarantee the convergence of the sequence $\theta^{(m)}$ even if we said that the the parameters of the EM algorithm wont get worse in terms of their likelihood. Indeed, there is no general convergence theorem for the EM algorithm: the convergence of the sequence $\theta^{(m)}$ depends on the characteristics of log likelihood and the Q function, and also the initial guess parameters. Under certain regularity conditions, one can prove that $\theta^{(m)}$ converges to a stationary point of log likelihood. However, this convergence is only linear. Using NewtonRaphson updates instead of using the EM algorithm, and one could locally maximize the likelihood, but this involves calculating the inverse of the Hessian matrix, which has quadratic convergence. Superlinear convergence could instead be achieved using conjugate gradient methods or quasi-Newton updates such as the BroydenFletcherGoldfarbShanno (BFGS) update, which only require computing the gradient of the log-likelihood [27, 45]. The NewtonRaphson method

can be expected to hone in on final guess parameters fast once $\theta^{(m)}$ is close, but EM may be more effective given a poor initial guess, in part because the Hessian matrix for the NewtonRaphson method may not be positive definite and hence makes the inversion unstable. EM convergence is discussed in details in [15,16,17,18], and [19] provides the rate of convergence of the EM algorithm. And [20] discuss the convergence of EM on fitting Gaussian Mixture Models.

Jensens Inequality: Let f be convex function, and X be a random variable, then, $E[f(X)] \geq f(E[X])$. $E[f(X)] = f(E[X])$ holds true if and only if $X = E[X]$ with probability 1. Figure 2 shows the Jensens inequality by using a convex function f. And f is shown by the solid line. X is a random variable, which has a probability of 0.5 taking the value a, and 0.5 taking the value b. Thus, the expected value of X is given by the midpoint between a and b. And $f(a)$, $f(b)$ and $f(E[X])$ indicated on the y-axis with the value $E[f(X)]$ is the midpoint on the y-axis between $f(a)$ and $f(b)$. Therefore, due to f is convex, it must be the case that $E[f(X)] \geq f(EX)$. Conversely, if f is a concave function, we have that, $E[f(X)] \leq f(EX)$.

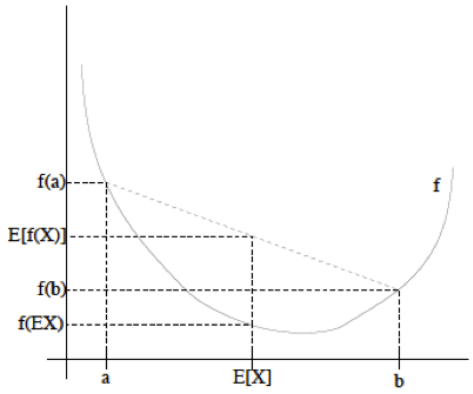

Fig. 2: Jensens Inequality illustration

Hence, the monotonicity of EM algorithm can be proved (Figure 3 gives a visualization of the EM convergence):

$$\begin{aligned}
\ell(\theta) &= \log p(y|\theta) \\
&= \log \int_{\mathcal{X}(y)} p(x, y|\theta) dx \\
&= \log \int_{\mathcal{X}(y)} (\frac{p(x, y|\theta)}{p(x|y, \theta^{(m)})}) p(x|y, \theta^{(m)}) dx \\
&= \log E_{X|y,\theta^{(m)}} \left[\frac{p(X, y|\theta)}{p(X|y, \theta^{(m)})} \right] \\
&\geq E_{X|y,\theta^{(m)}} \left[\log \frac{p(X, y|\theta)}{p(X|y, \theta^{(m)})} \right]
\end{aligned} \tag{15}$$

$$
\begin{aligned}
&= E_{X|y,\theta^{(m)}}\left[\log \frac{p(X|\theta)p(y|X)}{p(X|\theta^{(m)})p(y|X)/p(y|\theta^{(m)})}\right] \\
&= E_{X|y,\theta^{(m)}}\left[\log \frac{p(X|\theta)p(y|\theta^{(m)})}{p(X|\theta^{(m)})}\right] \\
&= E_{X|y,\theta^{(m)}}[\log p(X|\theta)] - E_{X|y,\theta^{(m)}}[\log p(X|\theta^{(m)})] + \log p(y|\theta^{(m)}) \\
&= Q(\theta|\theta^{(m)}) - Q(\theta^{(m)}|\theta^{(m)}) + \ell(\theta^{(m)})
\end{aligned} \tag{16}
$$

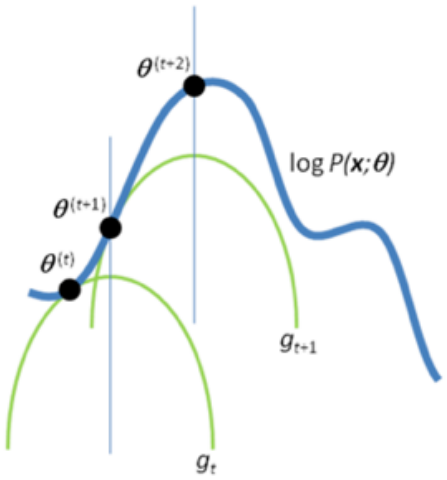

Fig. 3: Visualization of EM convergence

we can conclude the first part of the proof as a lower bound on the log-likelihood function:

$$\ell(\theta) \geq \ell(\theta^{(m)}) + Q(\theta|\theta^{(m)}) - Q(\theta^{(m)}|\theta^{(m)}) \tag{17}$$

Next, since assume that $Q(\theta|\theta^{(m)}) \geq Q(\theta^{(m)}|\theta^{(m)})$, then:

$$\ell(\theta) \geq \ell(\theta^{(m)}) + (Q(\theta|\theta^{(m)}) - Q(\theta^{(m)}|\theta^{(m)})) \geq \ell(\theta^{(m)}), \tag{18}$$

which completes the proof.

In the case of extension of EM to MAP:

$$\ell(\theta) + \log p(\theta) \geq \ell(\theta^{(m)}) + \log p(\theta^{(m)}), \tag{19}$$

if

$$Q(\theta|\theta^{(m)}) + \log p(\theta) \geq Q(\theta^{(m)}|\theta^{(m)}) + \log p(\theta^{(m)}). \tag{20}$$

2) MaximizationMaximization: Another way to view the EM algorithm is as a joint maximization procedure that iteratively maximizes a better and better lower bound F to the log-likelihood function $\ell(\theta)$ [21]. Specifically, we will guess that X has distribution $\tilde{P}$ with support $X(y)$ and density $\tilde{p}(x)$. Let

P_θ denote the conditional distribution with density $p(x|y,\theta)$. Then consider maximizing the following objective function alternately with respect to $\tilde{P}$ and θ:

$$F(\tilde{P},\theta) = \ell(\theta) - D_{KL}(\tilde{P}||P_\theta), \tag{21}$$

where $D_{KL}(\tilde{P}||P_\theta)$ is the KullbackLeibler divergence between the current guess $\tilde{P}$ of the distribution over the complete data, and the likelihood P_θ of the complete data given the parameter θ. Maximizing $F(\tilde{P},\theta)$ with respect to θ maximizes a lower bound on the log-likelihood function $\ell(\theta)$ since the KL divergence is always nonnegative. Then maximizing $F(\tilde{P},\theta)$ with respect to $\tilde{P}$ attempts to tighten the lower bound for your current estimate of θ. Since both steps perform maximization, this view of the EM algorithm is called maximizationmaximization. This joint maximization view of EM is useful as it has led to variants of the EM algorithm that use alternative strategies to maximize $F(\tilde{P},\theta)$, for example by performing partial maximization in the first maximization step [21].

This interpretation establishes EM as belonging to the class of methods called alternating optimization or alternating minimization methods. This class of methods also includes projection onto convex sets (POCS) and the BlahutArimoto algorithms [22,23].

Max Step 1:

$$\tilde{P}^{(m)} = \arg\max_{\tilde{P}} F(\tilde{P},\theta^{(m-1)}) \tag{22}$$

Max Step 2:

$$\theta^{(m)} = \arg\max_{\theta\in\Omega} F(\tilde{P},\theta) \tag{23}$$

and,

$$\begin{aligned}
\tilde{P}^{(m)} &= \arg\max_{\tilde{P}}(\ell(\theta^{(m-1)}) - D_{KL}(\tilde{P}||P_{\theta^{(m-1)}})) \\
&= \arg\min_{\tilde{P}} D_{KL}(\tilde{P}||P_{\theta^{(m-1)}}) \\
&= P_{\theta(m-1)} \\
\theta^{(m)} &= \arg\max_{\theta\in\Omega} \ell(\theta) - D_{KL}(\tilde{P}^{(m)}||P_\theta) \\
&= \arg\max_{\theta\in\Omega} \log p(y|\theta) - D_{KL}(\tilde{P}^{(m)}||P_\theta) \\
&= \arg\max_{\theta\in\Omega} \log p(y|\theta) \int_{\mathcal{X}_{(y)}} p(x|y,\theta^{(m-1)})dx - D_{KL}(\tilde{P}^{(m)}||P_\theta) \\
&= \arg\max_{\theta\in\Omega} \int_{\mathcal{X}_{(y)}} p(x|y,\theta^{(m-1)}) \log p(y|\theta)dx - D_{KL}(\tilde{P}^{(m)}||P_\theta)
\end{aligned}$$

$$
\begin{aligned}
&= \arg\max_{\theta\in\Omega} \int_{\mathcal{X}_{(y)}} p(x|y,\theta^{(m-1)}) \log \frac{p(y|x)p(x|\theta)}{p(x|y,\theta)} dx - D_{KL}(\tilde{P}^{(m)}||P_\theta) \\
&= \arg\max_{\theta\in\Omega} \int_{\mathcal{X}_{(y)}} p(x|y,\theta^{(m-1)}) \log \frac{p(x|\theta)}{p(x|y,\theta)} dx - D_{KL}(\tilde{P}^{(m)}||P_\theta) \\
&= \arg\max_{\theta\in\Omega} \int_{\mathcal{X}_{(y)}} p(x|y,\theta^{(m-1)}) \log \frac{p(x|\theta)}{p(x|y,\theta)} dx - \int_{\mathcal{X}_{(y)}} p(x|y,\theta^{(m-1)}) \log \frac{p(x|y,\theta^{(m-1)})}{p(x|y,\theta)} dx \\
&= \arg\max_{\theta\in\Omega} \int_{\mathcal{X}_{(y)}} p(x|y,\theta^{(m-1)}) \log p(x|\theta) dx - \int_{\mathcal{X}_{(y)}} p(x|y,\theta^{(m-1)}) \log p(x|y,\theta^{(m-1)}) dx \\
&= \arg\max_{\theta\in\Omega} \int_{\mathcal{X}_{(y)}} p(x|y,\theta^{(m-1)}) \log p(x|\theta) dx \\
&= \arg\max_{\theta\in\Omega} E_{X|y,\theta^{(m-1)}}[\log p(X|\theta)] \\
&= \arg\max_{\theta\in\Omega} Q(\theta|\theta^{(m-1)}) \qquad (24)
\end{aligned}
$$

which is just the standard M-step.

III. Simulations and Results

In this section, we will use a simulation on GMM to better illustrate the EM algorithm. First, we will give an example about GMM, and we will compare the EM algorithm with a similar method called K-means. Furthermore, the simulation results will be given at the end of this section. More about mixture models can be found in [24].

A. Learning GMM

Figure 4 shows the probability density function of a one-dimensional GMM with three components.

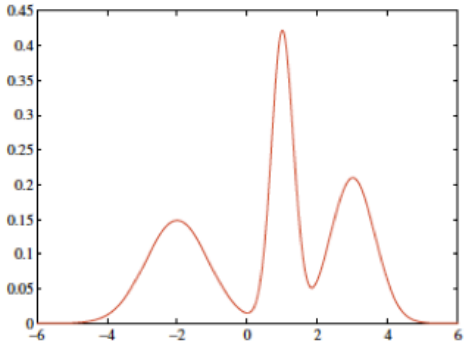

Fig. 4: Probability density of a one-dimensional GMM with three Gaussian components with means $\mu_1 = 2$, $\mu_2 = 1$, $\mu_3 = 3$, $\sigma_1^2 = 0.8$, $\sigma_2^2 = 0.1$, $\sigma_3^2 = 0.4$, and relative weights $w_1 = w_2 = w_3 = 1/3$.

Now Consider a two-component GMM with relative weights $w1 = 0.6$ and $w2 = 0.4$. Figure 5 gives its density, which also shows 1000 samples randomly drawn from this, distribution; samples from the first and second components are marked red and blue, respectively. And the final estimates are:

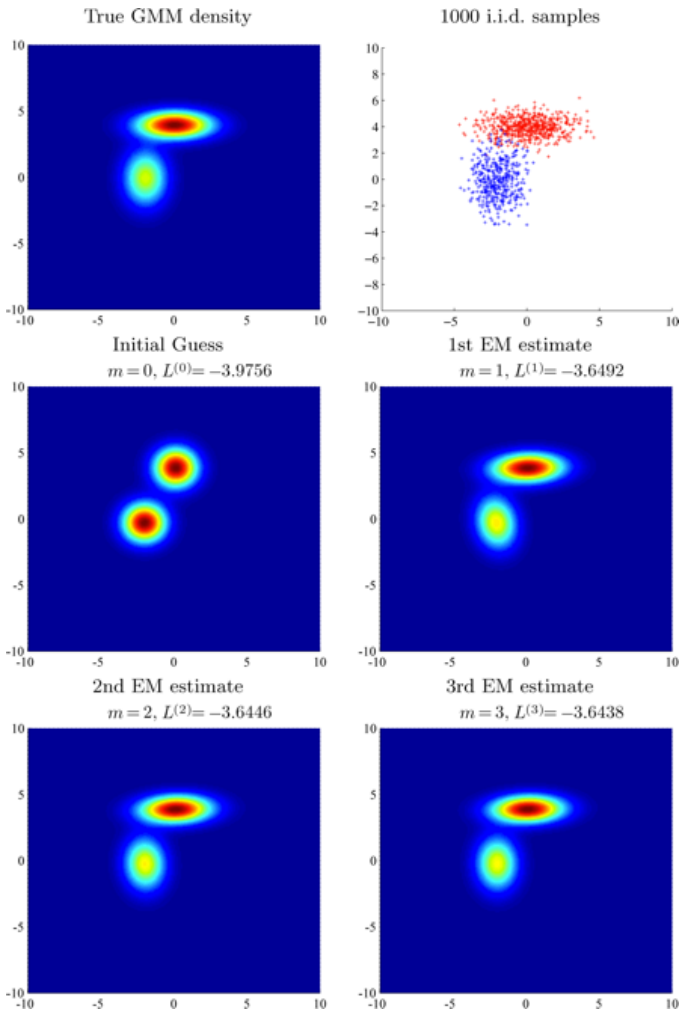

Fig. 5: GMM fitting example

B. *K-means Method*

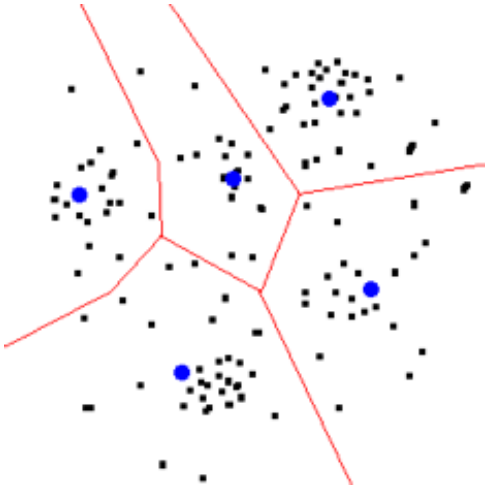

Fig. 6: K-means illustration

Figure 4 shows the how K-means Method works. Considering a simple variant of EM: A simple variant

is to use only the m^{th} maximum likelihood estimate $x^{(m)}$ of the complete data x:

E-like step:

$$x^m = \arg \max_{x \in \mathcal{X}(y)} p(x|y, \theta^m) \tag{25}$$

M-like step:

$$\theta^{(m+1)} = \arg \max_{\theta \in \Omega} p(x^m|\theta) \tag{26}$$

First, one makes an initial guess of the k cluster centers. Then in the E-like step, one assigns each of the n points to the closest cluster based on the estimated cluster centers $\theta^{(m)}$. Then in the M-like step, one takes all the points assigned to each cluster, and computes the mean of those points to form a new estimate of the clusters centroid. Underlying k-means is a model that the clusters are defined by Gaussian distributions with unknown means (the θ to be estimated) and identity covariance matrices.

EM clustering differs from k-means clustering in that at each iteration you do not choose a single x^m, that is, one does not force each observed point yi to belong to only one cluster. Instead, each observed point y_i is probabilistically assigned to the k clusters by estimating $p(x|y, \theta^{(m)})$.

And sometimes EM algorithm can perform better than K-means as we illustrated in Figure 5.

Fig. 7: different cluster analysis results

C. Simulation

Figure 5 shows the two true Gaussian model mixed together, and we want to clustering the two Gaussian model. For the parameters of the two Gaussian model, we randomly generate 200 points for each Gaussian model and mixed them together. The first Gaussian model has mean μ_1 = [0, 5], and [5,0] for the second. [[5, 0], [0, 2]] and [[3, 0], [0, 8]] are the covariance matrix for the Gaussian models.

Follow the EM algorithm steps mentioned previously on section 2, we calculate the probability density function for the data and maximize the Q-function to update the parameters for the model. The stop criterion we set for the algorithm to stop is the absolute distance between the new parameters and the last parameters we get from the M step. And the value for the stop criterion is 0.01.

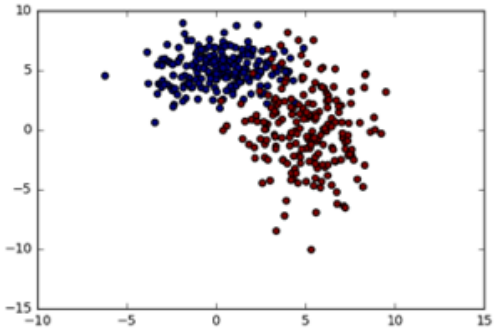

Fig. 8: True Gaussian mixture model with 200 points each

D. Simulation

Figure 9 gives 11 iterations of the EM algorithm and we can see from each of the iteration that the new parameters result from the maximization step is never get worse. And the area of the red and blue is getting separate more and more obvious as the iteration increases.

IV. Conclusion

EM algorithm provides a way to solve the difficult problem faced by some statistical model. Although it can not guarantee to find the global maximum for the likelihood function, by making several initial guess we have the possibility to find the best parameters.

EM was formalized as an approach to solving arbitrary maximum likelihood problems and named EM in a seminal 1977 paper by Dempster et al. [25] Baum et al. and Welch developed an algorithm for fitting hidden Markov models (HMMs) that is often called the BaumWelch algorithm, which is equivalent to applying the EM algorithm.

In this report, we also discussed about the clustering of two mixture Gaussian models, the simulation results are great illustration of EM algorithm. However, the disadvantage of the EM algorithm may reflect on the computational speed.

For EM algorithm, it is computational complex, and has to update the parameters each iteration and substitute the new parameters into the Q function to formulate the probabilistic density function in order to maximize it to get the next new parameters.

EM algorithm is not general purpose, unlike Newton-type methods, which can be automated very nicely using methods such as Automatic differentiation. Thus, we have to do a lot of math and program it specially for each problem. EM algorithm is robustness far from the optimal value is a strength but this

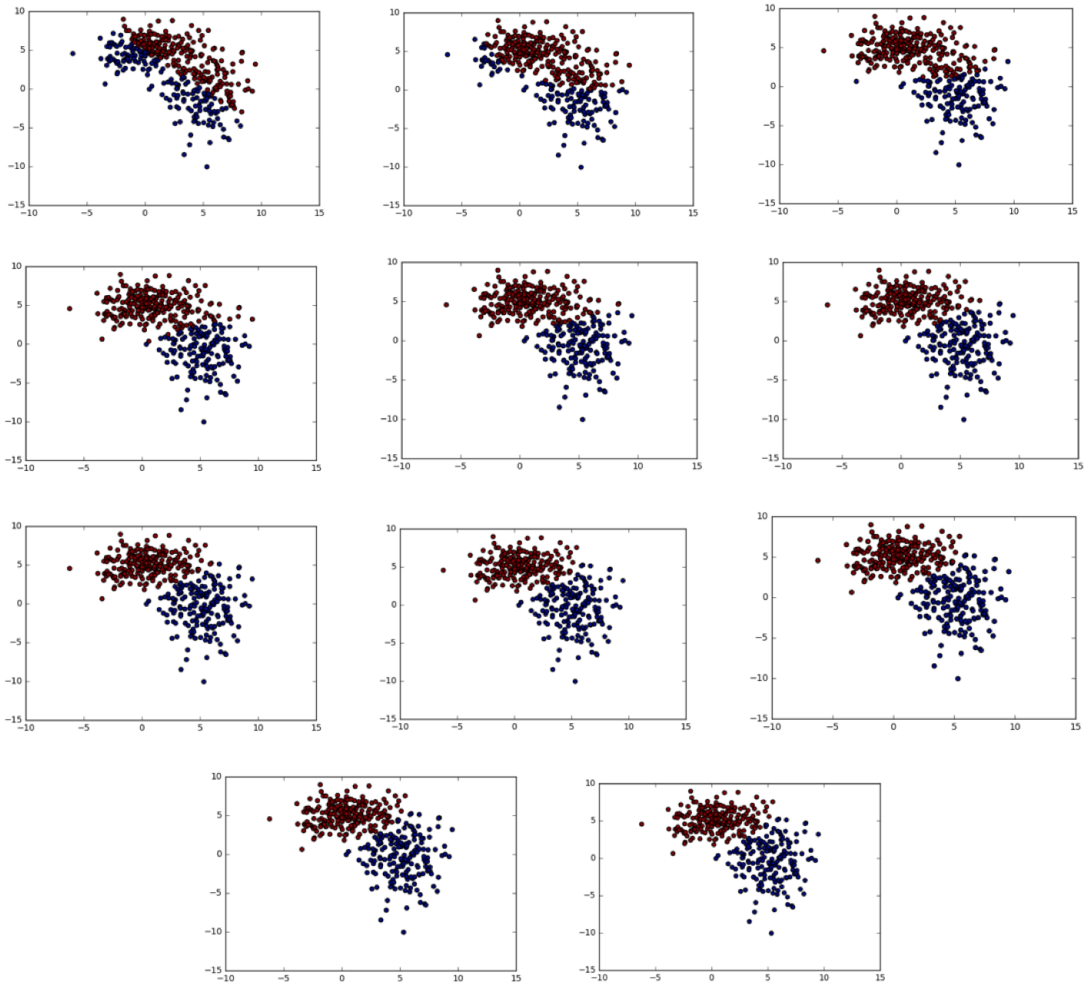

Fig. 9: Iteration 1 to iteration 11 with initial guess at 0.5, and finally converge in iteration 11. (The color for the last iteration and the true image of the Gaussian model is different, which is not matter in this case because EM clustering the two Gaussian model at the end)

ends up becoming a weakness near the optimal value because it converges very slowly there. Newton-type methods have the opposite issue.

V. Reference

[1] Ceppellini, R., Siniscalco, M. & Smith, C.A. Ann. Hum. Genet. 20, 97115 (1955).

[2] Hartley, H. Biometrics 14, 174194 (1958).

[3] Baum, L.E., Petrie, T., Soules, G. & Weiss, N. Ann. Math. Stat. 41, 164171 (1970).

[4] Dempster, A.P., Laird, N.M. & Rubin, D.B. J. R. Stat. Soc. Ser. B 39, 138 (1977).

[5] Geoffrey McLachlan and Thriyambakam Krishnan. The EM Algorithm and Extensions. John Wiley & Sons, New York, 1996.

[6] Geoffrey McLachlan and David Peel. Finite Mixture Models. John Wiley & Sons, New York, 2000.

[8] Krogh, A., Brown, M., Mian, I.S., Sjlander, K. & Haussler, D. J. Mol. Biol. 235, 15011543 (1994)

[9] Eddy, S.R. & Durbin, R. Nucleic Acids Res. 22, 20792088 (1994).

[10] Segal, E., Yelensky, R. & Koller, D. Bioinformatics 19, i273i282 (2003).

[11] Slatkin, M. & Excoffier, L. Heredity 76, 377383 (1996).

[12] Nesvizhskii, A.I., Keller, A., Kolker, E. & Aebersold, R. Anal. Chem. 75, 46464658 (2003).

[13] De Pierro, A.R. IEEE Trans. Med. Imaging 14, 132137 (1995).

[14] Maya R. Gupta and Yihua Chen. Theory and Use of the EM Algorithm. Foundations and Trends in Signal Processing. Vol. 4, No.3 (2010) 223-296.

[15] C. F. J. Wu, On the convergence properties of the EM algorithm, The Annals of Statistics, vol. 11, no. 1, pp. 95103, March 1983.

[16] R. A. Boyles, On the convergence of the EM algorithm, Journal of the Royal Statistical Society, Series B (Methodological), vol. 45, no. 1, pp. 4750, 1983.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B (Methodological), vol. 39, no. 1, pp. 138, 1977.

[18] R. A. Redner and H. F. Walker, Mixture densities, maximum likelihood and the EM algorithm, SIAM Review, vol. 26, no. 2, pp. 195239, April 1984.

[19] X.-L. Meng and D. B. Rubin, On the global and component wise rates of convergence of the EM algorithm, Linear Algebra and its Applications, vol. 199, pp. 413425, March 1994.

[20] R. M. Neal and G. E. Hinton, A view of the EM algorithm that justifies incremental, sparse, and other variants, in Learning in Graphical Models, (M. I. Jordan, ed.), MIT Press, November 1998.

[21] H. Stark and Y. Yang, Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics. New York, NY: John Wiley & Sons, 1998.

[22] R. W. Yeung, A First Course in Information Theory. New York, NY: Springer, 2002.

[23] G. J. McLachlan and D. Peel, Finite Mixture Models. New York, NY: John Wiley & Sons, 2000.

[24] M. Hazen and M. R. Gupta, A multiresolutional estimated gradient architecture for global optimization, in Proceedings of the IEEE Congress on Evolutionary Computation, pp. 30133020, 2006.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B (Methodological), vol. 39, no. 1, pp. 138, 1977.