Component-wise Density Smoothing for Parameter Estimation of Mixture Models

Chandan K. Reddy and Hsiao-Dong Chiang

Department of Electrical and Computer Engineering
Cornell University, Ithaca, NY-14853.

Abstract

Obtaining an optimal set of parameters to fit a mixture model to a given data corresponds to finding the global maximum of its highly nonlinear likelihood surface. This task has many applications in science and engineering and is widely accepted as one of the most challenging problems. This paper introduces a new smoothing algorithm for learning a mixture model from multivariate data. Our algorithm is based on the conventional Expectation-Maximization (EM) approach applied to a smoothened likelihood surface. A family or hierarchy of smooth log-likelihood surfaces is constructed using convolution based approaches. In simple terms, our method first smoothens the likelihood function and then applies the EM algorithm to obtain a promising solution on the smooth surface. This solution is used as an initial guess for the EM algorithm applied to the next smooth surface in the hierarchy. This process is repeated until the original likelihood surface appears. The smoothing process reduces the overall gradient of the surface and the number of local maxima. This effective optimization procedure eliminates extensive search in the non-promising regions of the parameter space. Results on benchmark datasets demonstrate significant improvements of the proposed algorithm compared to other approaches. Reduction in the number of unique local maxima has also been demonstrated empirically on several datasets.

Key words: expectation maximization, unsupervised learning, parameter estimation, mixture models, data clustering, convolution, kernels, hierarchical smoothing.

Email address: ckr6@cornell.edu (Chandan K. Reddy and Hsiao-Dong Chiang).

1 Introduction

In the field of statistical pattern recognition, finite mixtures allow a probabilistic model-based approach to unsupervised learning [1]. One of the most popular methods used for fitting mixture models to the observed data is the *Expectation-Maximization* (EM) algorithm which converges to the maximum likelihood estimate of the mixture parameters locally [2,3]. The usual steepest descent, conjugate gradient, or Newton-Raphson methods are too complicated for use in solving this problem [4]. EM has become a popular method since it takes advantages of problem specific properties. EM based methods have been widely used and applied successfully to solve wide range of problems in pattern recognition [5,6], clustering [7], information retrieval [8], computer vision [9], econometrics [10] etc.

One of the key problems with the EM algorithm is that it is a 'greedy' method which is sensitive to initialization. The log-likelihood surface on which the EM algorithm is applied is very rugged with many local maxima. Because of its greedy nature, EM algorithm tends to get stuck at a local maximum that corresponds to erroneous set of parameters for the mixture components. Usually, a global method which incorporates the global structure of the problem guides the EM algorithm to obtain a more precise set of parameters correspond to a higher likelihood function value. Obtaining an improved likelihood function value not only provides better parameter estimates but also enhances the generalization capability of the given mixture models [11]. The main concerns that motivated the new algorithm presented in this paper are :

- The log-likelihood surface is very rugged and the number of local maxima grows very fast with the data points and the components.
- The most widely used EM algorithm for mixture modeling converges to a local maximum of the likelihood function very quickly.
- Model selection criteria usually assumes that the global optimal solution of the log-likelihood function can be obtained. However, achieving this is computationally intractable.
- The local maxima are not uniformly distributed across the entire search space and most of the regions are not promising.

Of all the concerns mentioned above, the fact that the local maxima are not uniformly distributed makes it important for us to develop algorithms that help in avoiding search in non-promising regions. More focus needs to be given for searching the promising subspace by obtaining promising initial estimates. This can be achieved by smoothing the surface and obtaining promising regions and then gradually trace back these solutions onto the original surface. In this paper, we develop a hierarchical smoothing algorithm for the mixture modeling problem using convolution-based approach. We propose the following desired

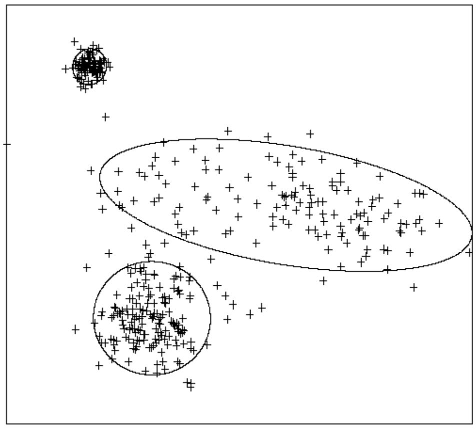
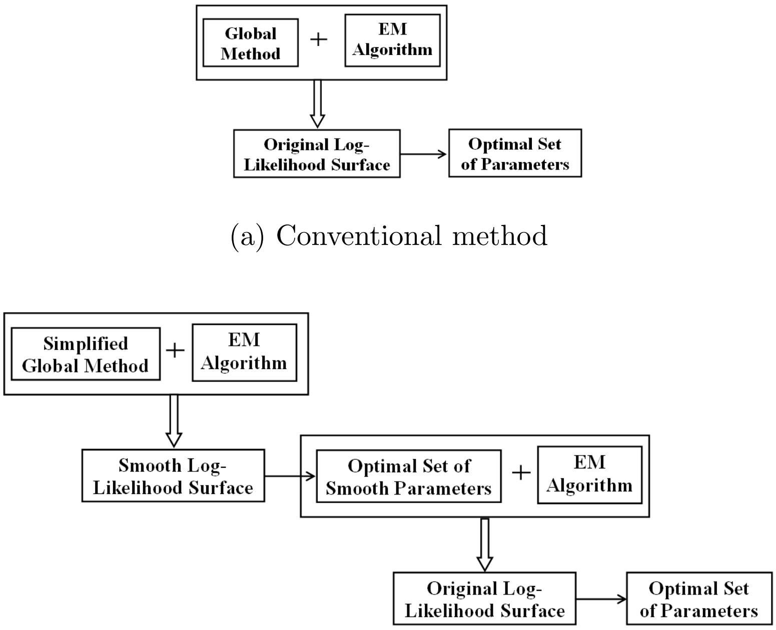


Fig. 1. Data consisting of three Gaussian components with different mean and variance. Note that each data point doesn't have a hard membership that it belongs to only one component. Most of the points in the first component will have high probability with which they belong to it. In this case, the other components do not have much influence. Components 2 and 3 data points are not clear. The problem of learning mixture models involve not only estimating the parameters of the Gaussian components but also finding the probabilities with which each data sample belongs to the component.

properties for smoothing algorithms:

- Preserve the presence of the global optimal solution.
- Enhance the converge regions of the global optimal solutions and other promising sub-optimal solutions.
- Reduce the overall gradient of the surface.
- Reduce the number of local maximum or minimum.
- Smooth different regions of the search space differently.
- Avoid over smoothing which might make the surface too flat and cause convergence problems for the local solvers.

The rest of the paper is organized as follows: Section 2 gives some relevant background about various methods proposed in the literature for solving the parameter estimation problem. Section 3 discusses the basic ideas of maximum likelihood estimate (MLE) and Expectation Maximization (EM). Section 4 describes the different smoothing strategies and gives the corresponding EM updates. Section 5 discusses our new smoothing framework and details about its implementation. Section 6 shows the experimental results of our algorithm on both synthetic and real datasets. Finally, Section 7 concludes our discussion with future research directions.


Global Method
+
EM Algorithm
Original Log-Likelihood Surface
Optimal Set of Parameters
Simplified Global Method
+
EM Algorithm
Smooth Log-Likelihood Surface
Optimal Set of Smooth Parameters
+
EM Algorithm
Original Log-Likelihood Surface
Optimal Set of Parameters

(a) Conventional method

(b) Smoothing approach

Fig. 2. Block diagram of the traditional approach and the smoothing approach.

2 Relevant Background

Since EM is very sensitive to the initial set of parameters that it begins with, several methods are proposed in the literature to identify good initial points. More generic techniques like deterministic annealing [12], genetic algorithms [13] have been applied to obtain promising initial set of parameters. Some problem specific algorithms like split and merge EM [14], component-wise EM [15], greedy learning [16], incremental version for sparse representations [17], parameter space grid [18] are also proposed in the literature. Some of these algorithms are either computationally very expensive or infeasible when learning mixtures in high dimensional spaces [18]. Though many approaches are available, modifying the log-likelihood surface using smoothing techniques has not been studied so far.

Most of these algorithms eventually apply the EM algorithm to move to a locally maximal point on the likelihood surface. In practice, simpler approaches like running EM from several random initializations, and then choosing the final estimate that leads to the highest local maximum of the likelihood are also successful to certain extent [19,20]. Specifically, for a problem in this context, where there is a non-uniform distribution of local maxima, it is important avoid searching non-promising regions [21]. These methods, however, have difficulties in finding a good solution when the error surface are very rugged since they often get trapped in poor sub-optimal solutions.

Different smoothing strategies have been successfully used in various applications for solving a diverse set of problems. Smoothing techniques are used to reduce irregularities or random fluctuations in time series data [10,22]. In the field of natural language processing, smoothing techniques are also used for adjusting maximum likelihood estimate to produce more accurate probabilities for language models [23]. Convolution based smoothing approaches are predominantly used in the field of digital image processing for image enhancement by noise removal [24,25]. Other variants of smoothing techniques include continuation methods [26,27] which are used successfully in various applications. Different multi-level procedures other than smoothing and its variants are clearly illustrated in [28].

For optimization problems, smoothing procedure helps in reducing the ruggedness of the surface and helps the local methods to prevent the local minima problem. It was used for the structure prediction of molecular clusters [29]. The smoothing procedure described in this paper is a way to estimate the optimal set of parameters of the Gaussian components in an effective manner. It is an initialization procedure which has the capability to avoid searching non-promising regions. Not much focus is given about the model selection criterion [11]. In other words, our algorithm assumes that the number of components are known before hand. In summary, the main contributions of this paper are:

- Develop convolution-based smoothing algorithms for obtaining optimal set of parameters.
- Demonstrate that the density-based convolution on the entire dataset will result smoothing the likelihood surface with respect to the parameters.
- Empirically show that the number of local maxima on the log-likelihood surface is reduced.
- Show that smoothing helps in obtaining promising initial set of parameters

Fig. 2 compares the conventional approach with the smoothing approach. In the traditional approach, a global method in combination with the EM algorithm is used to find the optimal set of parameters on the log-likelihood surface. In the smoothing approach, a simplified version of the global method is applied in combination with the EM algorithm to obtain an optimal set of parameters on the smooth surface which are again used in combination with the EM algorithm to obtain optimal set of parameters on the original log-likelihood surface. Since the smoothened log-likelihood surface is easy to traverse (has fewer local maxima), one can gain significant computational benefits by applying a simplified global method compared to that of the conventional global method on the original log-likelihood surface which are usually very expensive.

3 Preliminaries

We need to introduce the following preliminaries on mixture models, EM algorithm and convolution kernels. We first describe the notation we used throughout the paper:

Table 1
Description of the Notations used

Notation	Description
d	number of features
n	number of data points
k	number of components
s	total number of parameters
Θ	parameter space
$\tilde{\Theta}$	smooth parameter space
θ_i	parameters of a single i^{th} component
θ_0	parameters of the smoothing kernel
α_i	mixing weights for i^{th} component
$\mathcal{X}$	observed data
$\mathcal{Z}$	missing data
$\mathcal{Y}$	complete data
t	timestep for the estimates

3.1 Mixture Models

Lets assume that there are k Gaussians in the mixture model. The form of the probability density function is as follows:

$$p(x|\Theta) = \sum_{i=1}^{k} \alpha_i p(x|\theta_i) \tag{1}$$

where $x = [x_1, x_2, ..., x_d]^T$ is the feature vector of d dimensions. The α_k's represent the *mixing weights* (which sum to one). Θ represents the collection of parameters $(\alpha_1, \alpha_2, ...\alpha_k, \theta_1, \theta_2, ...\theta_k)$ and p is a multivariate density function parameterized by θ_i. Also, it should be noticed that being probabilities α_i must satisfy

$$0 \leq \alpha_i \leq 1 \ , \ \forall i = 1, .., k, \ \textit{and} \ \sum_{i=1}^{k} \alpha_i = 1 \tag{2}$$

Given a set of n i.i.d samples $\mathcal{X} = \{x^{(1)}, x^{(2)}, .., x^{(n)}\}$, the log-likelihood corresponding to a mixture is

$$log \ p(\mathcal{X}|\Theta) = log \prod_{j=1}^{n} \ p(x^{(j)}|\Theta) = \sum_{j=1}^{n} log \sum_{i=1}^{k} \alpha_i \ p(x^{(j)}|\theta_i) \tag{3}$$

The goal of learning mixture models is to obtain the parameters $\widehat{\Theta}$ from a set of n data points which are the samples of a distribution with density given by (1). The *Maximum Likelihood Estimate* (MLE) is given by :

$$\widehat{\Theta}_{MLE} = arg \max_{\Theta} \ \{ \ log \ p(\mathcal{X}|\Theta) \ \} \tag{4}$$

Since, this MLE cannot be found analytically for mixture models [11], one has to rely on iterative procedures that can find the global maximum of $log \ p(\mathcal{X}|\Theta)$. The EM algorithm [30] described in the next section has been successfully used to obtain a local maximum of the given likelihood function.

3.2 Expectation Maximization

The EM algorithm assumes $\mathcal{X}$ to be *observed* data. The missing part is a set of n labels $\mathcal{Z} = \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, .., \mathbf{z}^{(n)}\}$ associated with n samples, indicating which component produced each sample [30]. Each label $\mathbf{z}^{(j)} = [z_1^{(j)}, z_2^{(j)}, .., z_k^{(j)}]$ is a binary vector where $z_i^{(j)} = 1$ and $z_m^{(j)} = 0 \ \forall m \neq i$, means the sample $x^{(j)}$ was produced by the i^{th} component. Now, the complete log-likelihood (i.e. the one from which we would estimate Θ if the *complete data* $\mathcal{Y} = \ \{ \ \mathcal{X}, \mathcal{Z} \ \}$ is

$$log \ p(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{j=1}^{n} \ log \prod_{i=1}^{k} \ [\ \alpha_i \ p(x^{(j)}|z_i^{(j)}, \theta_i) \ ]^{z_i^{(j)}}$$

$$log \ p(\mathcal{Y}|\Theta) = \sum_{j=1}^{n} \sum_{i=1}^{k} z_i^{(j)} \ log \ [\ \alpha_i \ p(x^{(j)}|\theta_i) \ ] \tag{5}$$

The EM algorithm produces a sequence of estimates $\{\widehat{\Theta}(t), t = 0, 1, 2, ...\}$ by alternately applying two steps until convergence:

E-Step : Compute the conditional expectation of the complete likelihood, given $\mathcal{X}$ and the current estimate $\widehat{\Theta}(t)$. Since $log\ p(\mathcal{X}, \mathcal{Z}|\Theta)$ is linear with respect to the missing data $\mathcal{Z}$, we simply have to compute the conditional expectation $\mathcal{W} \equiv E[\mathcal{Z}|\mathcal{X}, \widehat{\Theta}(t)]$, and plug it into $log\ p(\mathcal{X}, \mathcal{Z}|\Theta)$. This gives the Q-function as follows:

$$Q(\Theta|\widehat{\Theta}(t)) \equiv E_Z[log\ p(\mathcal{X}, \mathcal{Z})|\mathcal{X}, \widehat{\Theta}(t)] \tag{6}$$

Since $\mathcal{Z}$ is a binary vector, its conditional expectation is given by :

$$\begin{aligned} w_i^{(j)} &\equiv E\ [\ z_i^{(j)}|\mathcal{X}, \widehat{\Theta}(t)\ ] = Pr\ [\ z_i^{(j)} = 1|x^{(j)}, \widehat{\Theta}(t)\ ] \\ &= \frac{\widehat{\alpha}_i(t)p(x^{(j)}|\widehat{\theta}_i(t))}{\sum_{i=1}^{k} \widehat{\alpha}_i(t)p(x^{(j)}|\widehat{\theta}_i(t))} \end{aligned} \tag{7}$$

where the last equality is simply the Bayes law (α_i is the a priori probability that $z_i^{(j)} = 1$), while $w_i^{(j)}$ is the a posteriori probability that $z_i^{(j)} = 1$ given the observation $x^{(j)}$.

M-Step : The estimates of the new parameters are updated using the following equation :

$$\widehat{\Theta}(t+1) = arg \max_{\Theta}\{Q(\Theta, \widehat{\Theta}(t))\} \tag{8}$$

Several variants of the EM algorithm have been extensively used to solve this problem. The convergence properties of the EM algorithm for Gaussian mixtures are thoroughly discussed in [4]. One of the main challenges of the EM algorithm is the initialization step. The final result obtained as a result of the iteration steps will significantly depend on the initial estimate of the parameters. In this paper, we explore the idea of smoothing the log-likelihood surface in order to reduce the number of local maxima thus diminishing the sensitivity of the initial parameters used. The approach taken is convolution based smoothing which is described in the next section.

3.3 Convolution Kernels

For smoothing the mixture model, any kernel can be used for convolution if it can yield a closed form solution in each E and M step. Three widely used

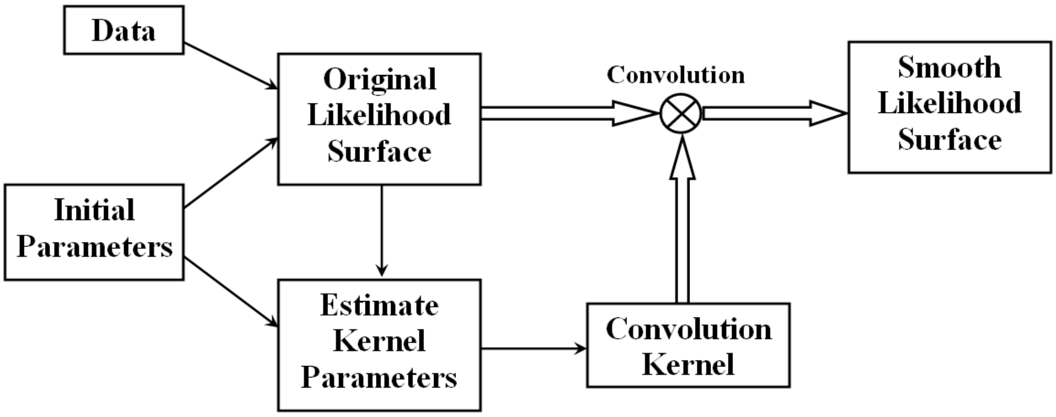
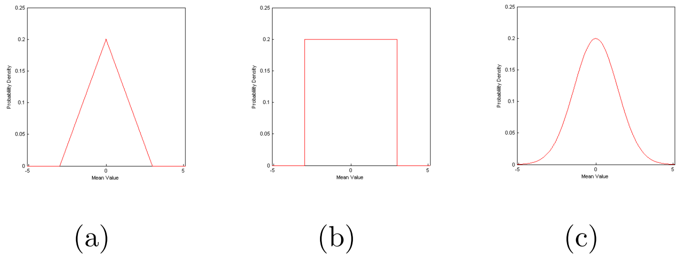

Data
Original Likelihood Surface
Convolution
Smooth Likelihood Surface
Initial Parameters
Estimate Kernel Parameters
Convolution Kernel

Fig. 3. Block Diagram of the smoothing approach. Smooth likelihood surface is obtained by convolving the original likelihood surface with a convolution kernel which is chosen to be a Gaussian kernel in our case.

kernels are shown in Fig. 4. We chose to use Gaussian kernel for smoothing the original log-likelihood function for the following reasons:

- When the underlying distribution is assumed to be generated from Gaussian components, Gaussian kernels give the optimal performance.
- The analytic form of the likelihood surface obtained after smoothing is very similar to the original likelihood surface.
- Since the parameters of the original components and the kernels will be of the same scale, changing the parameters correspondingly to scale will be much easier.



(a) (b) (c)

Fig. 4. Different convolution kernels (a) Triangular function (b) Step function and (c) Gaussian function

Instead of convolving the entire likelihood surface directly, we convolve individual components of the mixture model separately. Lets consider a Gaussian density function parameterized by θ_i(i.e. μ_i and σ_i) under the assumption that all the components have the same functional form (d variate Gaussians):

$$p(x|\theta_i) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \tag{9}$$

Lets consider the following Gaussian kernel :

$$g(x) = \frac{1}{\sigma_0\sqrt{2\pi}} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}} \tag{10}$$

$$\begin{aligned} p'(x|\theta_i) &= p(x|\theta_i) \otimes g(x) \\ &= \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \otimes \frac{1}{\sigma_0\sqrt{2\pi}} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}} \\ &= \frac{1}{\sqrt{2\pi(\sigma_i^2+\sigma_0^2)}} e^{-\frac{(x-(\mu_i+\mu_0))^2}{2(\sigma_i^2+\sigma_0^2)}} \end{aligned} \tag{11}$$

(Convolution of Gaussians): When a Gaussian density function with parameters μ_1 and σ_1 is convolved with a Gaussian kernel with parameters μ_0 and σ_0, then the resultant density function is also Gaussian with mean $(\mu_1 + \mu_0)$ and variance $(\sigma_1^2 + \sigma_0^2)$. The proof for this claim is given in Appendix-A.

Now, the new smooth density can be obtained by convolving with the gaussian kernel given by (10). Convolving two Gaussians to obtain another Gaussian is shown graphically in Fig. 5. It can also be observed that if the mean of one of the Gaussians is zero, then the mean of the resultant Gaussian is not shifted. The only change is in the variance parameter. Since, shifting mean is not a good choice for optimization problems and we are more interested in reducing the peaks, we chose to increase the variance parameter without shifting the mean.

4 Smoothing Log-Likelihood Surface

The overall log-likelihood surface can be convolved using a Gaussian kernel directly. This is not a feasible approach because of the following reasons:

- It results in an analytic expression that is not easy to work on and computing the EM updates will become cumbersome.
- It is Computationally very expensive.
- Different regions of search space must be smoothened differently. Choosing parameters to do this task is almost impossible.

The main trick here is to perform convolution component-wise. Our approach can smooth different regions of search space differently. Since the log-likelihood surface is obtained from individual densities, smoothing each component's individual density function will smoothen the overall log-likelihood surface. This will also give the flexibility to chose the kernel parameters which is discussed in following subsection.

After computing the new density (p'), we can define the

$$p'(x|\Theta) = \sum_{i=1}^{k} \alpha_i \; p'(x|\theta_i) \tag{12}$$

Now, the smooth log-likelihood function is given by:

$$f'(\mathcal{X}, \Theta) = \sum_{j=1}^{n} \; log \; \sum_{i=1}^{k} \alpha_i \; p'(x^{(j)}|\theta_i) \tag{13}$$

Theorem 1 *(Density Smoothing): Convolution of a Gaussian density function with parameters μ_1 and σ_1 with a Gaussian kernel with parameters $\mu_0 = 0$ and σ_0 is equivalent to convolving the function with respect to μ_1.*

Proof: See Appendix-B.

Fig. 3 shows the block diagram of the smoothing procedure. The original likelihood surface is obtained from the initial set of parameters and the given dataset. The kernel parameters are chosen from the initial set of parameters and the original log-likelihood surface. The kernel is then convolved with the original log-likelihood surface to obtain smooth log-likelihood surface.

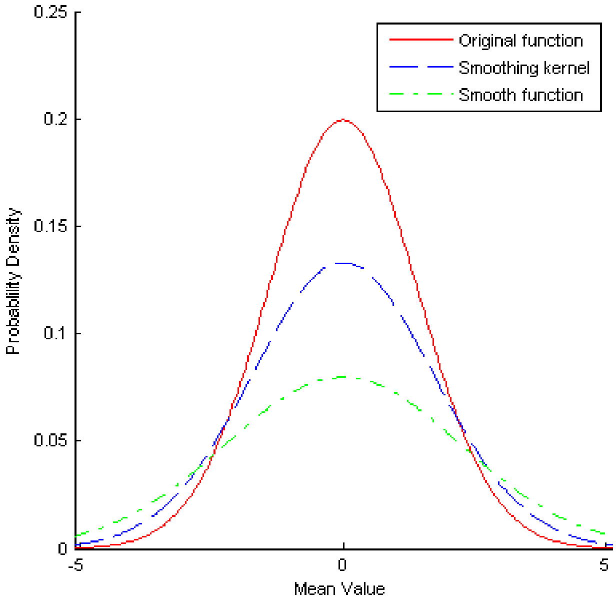


Fig. 5. The effects of smoothing a Gaussian density function with a Gaussian kernel.

4.1 Kernel Parameters

The parameters of the smoothing kernel can be chosen to be fixed so that they need not depend on the parameters of individual components. Fixed kernels will be effective when the underlying distribution comes from similar components. The main disadvantage of choosing a fixed kernel is that some of the components might not be smoothened while others might be over smoothened. Since, the Gaussian kernel has the property that the convolution sums up the parameters, this can also be treated as *Additive smoothing*. To avoid the problems of fixed kernel smoothing, we introduce the concept of variable kernel smoothing in this paper. Each component will be treated differently and smoothed according to the existing parameter values. This smoothing strategy is much more flexible and works effectively in practice. Since, the kernel parameters are effectively multiplied, this smoothing can be considered as *Multiplicative Smoothing*. In other words, σ_0 must be chosen individually for different components and it must be a function of σ_i. Both these approaches don't allow for smoothing the mixing weights parameters (α's).

4.2 EM Updates

For both of the above mentioned smoothing kernels, the following equations are valid. The complete derivations of these EM equations for the case of fixed kernel smoothing is given in Appendix C. The $Q - function$ of the EM algorithm applied to the smoothened log-likelihood surface is given by:

$$\begin{aligned} Q(\Theta|\widehat{\Theta}(t)) = \sum_{j=1}^{n}\sum_{i=1}^{k} w_i^{(j)}[log \frac{1}{\sqrt{2\pi(\tilde{\sigma}_i^2)}} \\ - \frac{(x^{(j)} - \tilde{\mu}_i)^2}{2\tilde{\sigma}_i^2} + log\ \alpha_i] \end{aligned} \tag{14}$$

where

$$w_i^{(j)} = \frac{\frac{\alpha_i(t)}{\tilde{\sigma}_i} e^{-\frac{1}{2\tilde{\sigma}_i^2}(x^{(j)} - \tilde{\mu}_i(t))^2}}{\sum_{i=1}^{k} \frac{\alpha_i(t)}{\tilde{\sigma}_i} e^{-\frac{1}{2\tilde{\sigma}_i^2}(x^{(j)} - \tilde{\mu}_i(t))^2}} \tag{15}$$

The updates for the maximization step in the case of GMMs are given as follows:

$$
\begin{aligned}
\tilde{\mu}_i(t+1) &= \frac{\sum_{j=1}^{n} w_i^{(j)} x^{(j)}}{\sum_{j=1}^{n} w_i^{(j)}} \\
\tilde{\sigma}_i^2(t+1) &= \frac{\sum_{j=1}^{n} w_i^{(j)} (x^{(j)} - (\tilde{\mu}_i(t+1))^2}{\sum_{j=1}^{n} w_i^{(j)}} \\
\tilde{\alpha}_i(t+1) &= \frac{1}{n} \sum_{j=1}^{n} w_i^{(j)}
\end{aligned}
\tag{16}
$$

5 Algorithm and its implementation

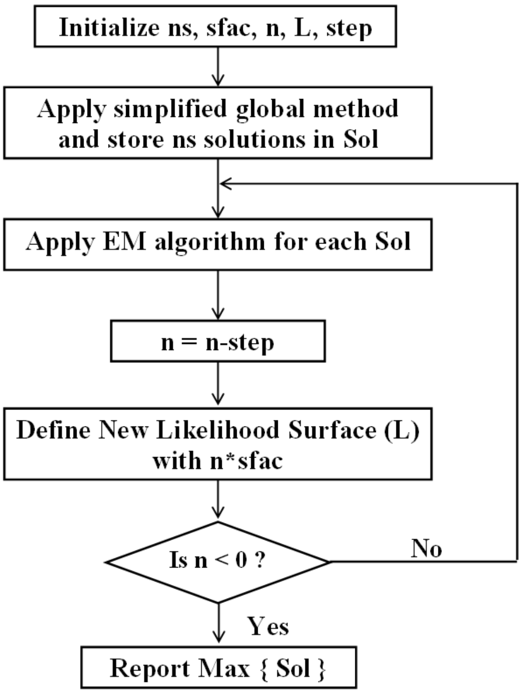

Initialize ns, sfac, n, L, step
Apply simplified global method and store ns solutions in Sol
Apply EM algorithm for each Sol
n = n-step
Define New Likelihood Surface (L) with n*sfac
Is n < 0 ?
No
Yes
Report Max { Sol }

Fig. 6. Flowchart of the smoothing algorithm

This section describes the smoothing algorithm in detail and explains the implementation details. The basic advantage of the smoothing approach is that a simplified global method can be used to explore fewer promising local maximum on the smoothened surface. These solutions are used as initial guesses for the EM algorithm which is again applied to the next level of smoothing. Smoothing will help to avoid search in unwanted non-promising areas of the parameter space. Fig. 6 gives the flow chart of our smoothing algorithm.

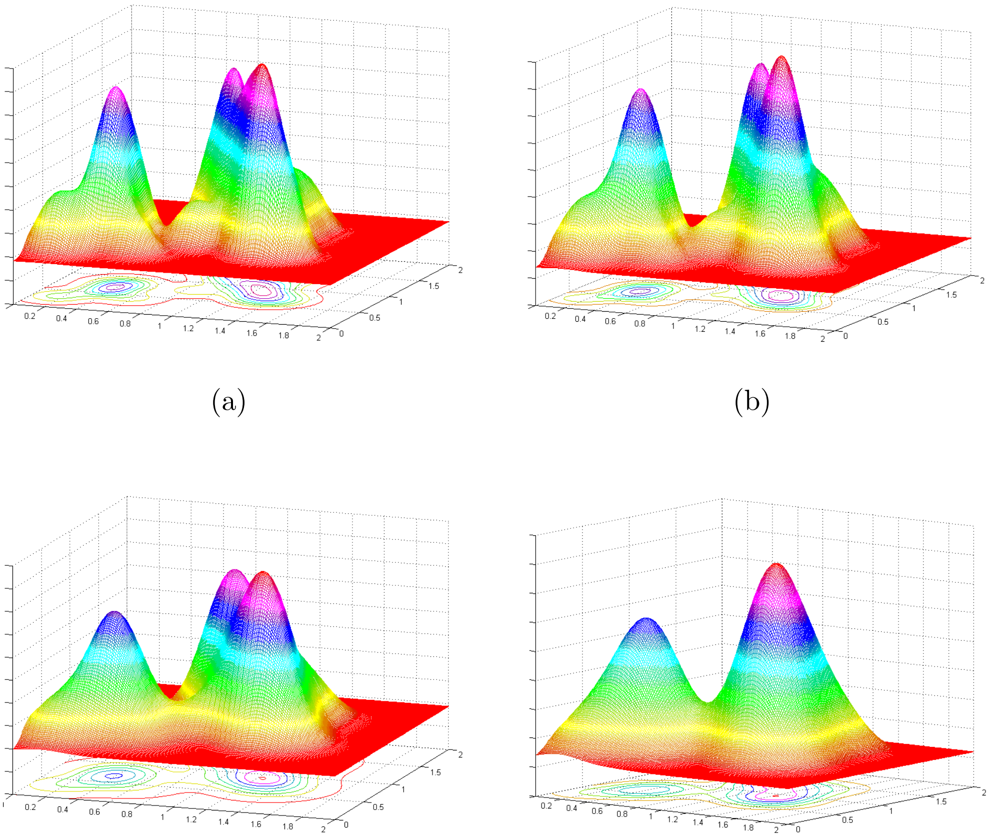
Before describing the algorithm, we first introduce certain variables that are used. The likelihood surface (defined by L) depends on the parameters and the available data. The smoothing factor ($sfac$) determines the extent to which the likelihood surface needs to be smoothened (which is usually chosen by trial-and-error). If the smoothing factor exceeds certain threshold, the number local maxima will increase tremendously which is clearly demonstrated in the next section. ns denotes the number of solutions that will be traced. n determines number of levels in the smoothing hierarchy. It is clear that there is a trade-off between the number of levels and the accuracy of the continuation method. Having many levels might increase the accuracy of the solutions, but it is computationally expensive. On the other hand, having few levels is computationally very cheap, but we might have to forgo the quality of the final solution. Deciding these parameters is not only user-specific but also depends significantly on the data that is being modeled. Algorithm 1 describes the smoothing approach.

Algorithm 1 Smooth

```
Input: Parameters $\Theta$, Data $\mathcal{X}$, Tolerance $\tau$, Smooth factor $Sfac$, number of levels $nl$, number of solutions $ns$
Output: $\widehat{\Theta}_{MLE}$
Algorithm:
step=1/nl      Sfac=Sfac/nl
L=Smooth($\mathcal{X}$, $\Theta$,nl*Sfac)
Sol=Global($\mathcal{X}$, $\Theta$, L,ns)
while n $\geq$ 0 do
    nl=nl-step
    L=Smooth($\mathcal{X}$, $\Theta$,nl*Sfac)
    for i=1:ns do
        Sol(i)=EM(Sol(i),$\mathcal{X}$,L,$\tau$)
    end for
end while
$\widehat{\Theta}_{MLE}$ =max{Sol}
```

The algorithm takes smoothing factor, number of levels, number of solutions, parameters set and the data. Smooth function returns the likelihood surface corresponding to smoothing factor at each level. Initially, a simple global method is used to identify promising solutions (ns) on the smooth likelihood surface which are stored in Sol. With these solutions as initial estimates, we then apply EM algorithm on the likelihood surface corresponding to the next level smooth surface. The EM algorithm also returns ns number of solutions corresponding to the ns number of initial estimates. At every iteration, new likelihood surface is constructed with a reduced smoothing factor. This process is repeated until the smoothing factor becomes zero which corresponds to the original likelihood surface. Though, it appears to be a daunting task, it can be easily implemented in practice. The main idea is to construct a family or hierarchy of surfaces and carefully trace the promising solutions from the top most surface to the bottom most one. In terms of tracing back the solutions to uncoarsened models, our method resembles other multi-level methods proposed in [31,32]. The main difference is that the dimensionality of the parameter space is not changed during the smoothing (or coarsening) process.

6 Results and Discussion



(a)

(b)

(c)

(d)

Fig. 7. Various stages during the smoothing process. (a) The original log-likelihood surface which is very rugged (b)-(c) Intermediate smoothened surfaces (d) Final smoothened surface with only two local maxima.

Our algorithm has been tested on four different datasets. The initial values for the centers were chosen from the available data points randomly. The covariances were chosen randomly and uniform prior is assumed for initializing the components.

A simple synthetic data with 40 samples and 5 spherical Gaussian components was generated and tested with our algorithm. Priors were uniform and the standard deviation was 0.01. The centers for the five components are given as follows: $\mu_1 = [0.3\ 0.3]^T$, $\mu_2 = [0.5\ 0.5]^T$, $\mu_3 = [0.7\ 0.7]^T$, $\mu_4 = [0.3\ 0.7]^T$ and $\mu_5 = [0.7\ 0.3]^T$.

The second dataset was that of a diagonal covariance case. The data generated from a two-dimensional, three-component Gaussian mixture distribution with mean vectors at $[0\ \ -2]^T, [0\ 0]^T, [0\ 2]^T$ and same diagonal covariance matrix with values 2 and 0.2 along the diagonal [12]. All the three mixtures have uniform priors. In the third synthetic dataset, a more complicated overlapping Gaussian mixtures are considered [15]. It has four components with 1000 data samples. The parameters are as follows: $\mu_1 = \mu_2 = [-4\ \ -4]^T$, $\mu_3 = [2\ 2]^T$ and $\mu_4 = [-1\ \ -6]^T$. $\alpha_1 = \alpha_2 = \alpha_3 = 0.3$ and $\alpha_4 = 0.1$.

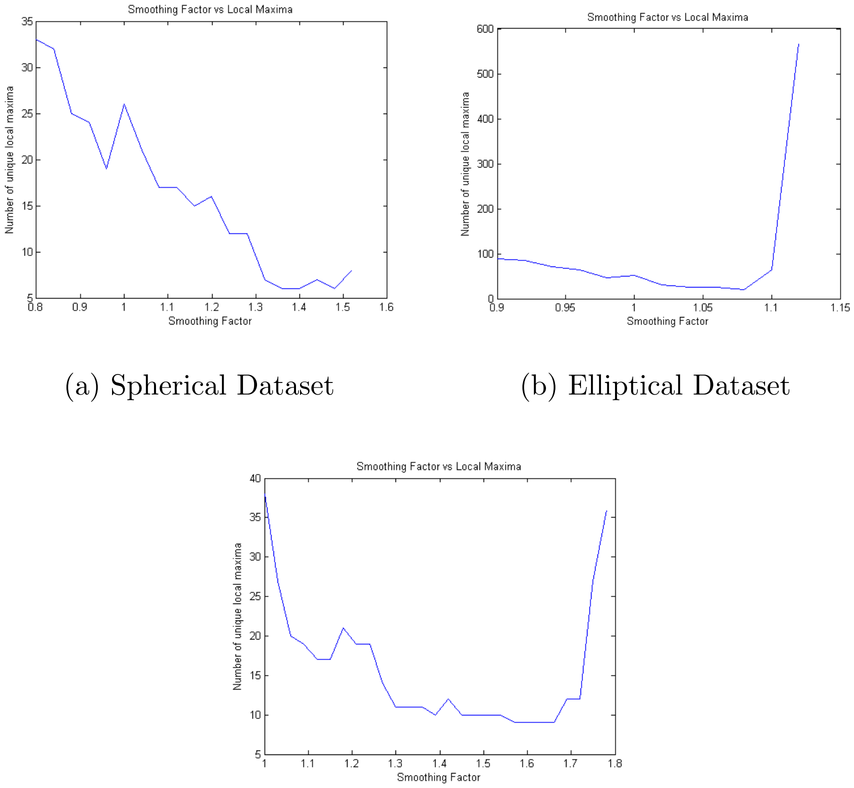
$$C_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad C_2 = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix}$$

$$C_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \qquad C_4 = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.125 \end{bmatrix}$$

Finally, we also used the widely used Iris real world data set. It contains 150 samples with 3 classes and 4 attributes. We have shown for all these datasets that the number of local minima gets reduced and the initial set of parameters chosen using the smoothing algorithm are more promising.

6.1 Reduction in the number of local maxima

One of the main advantages of the proposed smoothing algorithm is to ensure that the number of local maxima on the likelihood surface has been reduced. To the best of our knowledge, there is no theoretical way of estimating the amount of reduction in the number of unique local maxima on the likelihood surface. Hence, we use empirical simulations to justify the fact that the procedure indeed reduces the number of local maxima. Fig. 7 demonstrates the capability of our algorithm to reduce the number of local maxima. In this simple case, there were six local maxima originally, which were reduced to two local maxima after smoothing. Other stages during the transformation are also shown. Also, the fact that *the convergence regions of global optimal and other promising sub-optimal regions is enlarged* is clearly illustrated in the figure. The convergence regions of the two most promising areas are broadened in the smoothened likelihood surface. Fig. 8 shows the variation of the number of local maxima with respect to the smoothing factor for different datasets. Experiments were conducted using 100 random starts and the number of unique local maximum were stored (1000 in the case of elliptical dataset). For all these datasets, we have used the same random starts for the smoothened surfaces. These is a gradual reduction in the number of local maxima as the smooth factor is increased. One can see that if the smoothing factor is increased beyond



(a) Spherical Dataset

(b) Elliptical Dataset

(c) Iris Dataset

Fig. 8. Reduction in the number of unique local maximum for various datasets.

a certain threshold value (σ_{opt}), the number of local maxima increase rapidly. This might be due to the fact that over-smoothing the surface will make the surface flat, thus making it difficult for the EM to converge.

6.2 Smoothing for better Initialization

Smoothing the likelihood surface also helps in the optimization procedure. The average across all the starts is reported. The surface is then smoothened and the some promising solutions are used to trace the local optimal solutions and the average across all these starts are reported. For the smoothened version, only two levels were used. In other words, the optimal smoothing parameter is chosen and the EM algorithm is applied on the smoothened likelihood surface. The solutions obtained here are used as initial guesses on the original likelihood surface. Table 2 summarizes the results obtained directly with the original likelihood and the smoothened likelihood. We have used only two level and tracked three solutions for each level.

The two main claims (reduction in the number of local maximum and better

Table 2
Comparison of smoothing algorithm with the random starts. Mean and standard deviations across 100 random starts are reported.

Dataset	RS+EM	Smooth+EM
Spherical	36.3 ± 2.33	41.22 ± 0.79
Elliptical	-3219 ± 0.7	-3106 ± 12
Full covariance	-2391.3 ± 35.3	-2164.3 ± 18.56
Iris	-196.34 ± 15.43	-183.51 ± 2.12

initial estimates) about the contributions of this paper have been justified. More sophisticated global methods like Genetic algorithms, simulated annealing, adaptive sampling [33] etc. and their simplified versions can also be used in combination with our approach. Since the main focus of our paper is to demonstrate the smoothing capability, we used multiple random restarts as our global method.

7 Conclusion and Future Work

This paper introduces a smoothing approach for learning finite mixture models from multivariate data. Our algorithm is based on the conventional Expectation Maximization (EM) approach applied to a smoothened likelihood surface. A hierarchy of smooth surfaces is constructed and optimal set of parameters are obtained by applying a discrete version of continuation method to the promising solutions of the smooth surface. This smoothing process not only reduces the overall gradient of the surface but also reduces the number of local maxima. This is an effective optimization procedure that eliminates extensive search in the non-promising areas of the parameter space. Benchmark results demonstrate a significant improvement of the proposed algorithm compared to other existing methods.

As a continuation of this work, the effects of convolving Gaussian components with other kernels will be studied. Efficient algorithms for choosing the smoothing parameter automatically based on the available data will be developed. The idea of combining this hierarchical smoothing process with the model selection criteria will be explored. Optimal number of levels and the degree of smoothing at each level can be chosen adaptively so that significant amount of distortion does not occur. Though applied for GMMs in this paper, convolution based smoothing strategies can be treated as powerful optimization tools that can enhance the search capability significantly.

APPENDIX-A: Convolution of two Gaussians

Lets consider two gaussian density functions with parameters θ_1 and θ_0.

$$P(x|\theta_1) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}$$
$$P(x|\theta_0) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}}$$

By definition of convolution, we have

$$g(t) \otimes h(t) = \int_{-\infty}^{\infty} g(\tau)h(t-\tau)d\tau$$

$$\begin{aligned} c(x|\theta_1, \theta_0) &= p(x|\theta_1) \otimes p(x|\theta_0) \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{((x-\tau)-\mu_1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(\tau-\mu_0)^2}{2\sigma_0^2}} d\tau \\ &= \frac{1}{\sqrt{2\pi}\sigma_1} \frac{1}{\sqrt{2\pi}\sigma_0} \int_{-\infty}^{\infty} e^{-\frac{\sigma_0^2(\tau-x+\mu_1)^2+\sigma_1^2(\tau-\mu_0)^2}{2\sigma_1^2\sigma_0^2}} d\tau \end{aligned} \tag{.1}$$

After rearranging the terms that are independent of τ and further simplification, we get

$$c(x|\theta_1, \theta_0) = \frac{e^{-\frac{(x-(\mu_1+\mu_0))^2}{2(\sigma_1^2+\sigma_0^2)}}}{\sqrt{2\pi}\sigma_1 \sqrt{2\pi}\sigma_0} \int_{-\infty}^{\infty} e^{-\frac{(\tau-\mu_\tau)^2}{2\sigma_\tau^2}} d\tau$$

where

$$\mu_\tau = \frac{\mu_0\sigma_1^2 + (x-\mu_1)\sigma_0^2}{(\sigma_1^2+\sigma_0^2)}$$
$$\sigma_\tau = \frac{\sigma_1^2\sigma_0^2}{(\sigma_1^2+\sigma_0^2)}$$

Hence, we have

$$
\begin{aligned}
c(x|\theta_1, \theta_0) &= \frac{e^{-\frac{(x-(\mu_1+\mu_0))^2}{2(\sigma_1^2+\sigma_0^2)}}}{\sqrt{2\pi(\sigma_1^2+\sigma_0^2)}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\ \sigma_\tau}\ e^{-\frac{(\tau-\mu_\tau)^2}{2\sigma_\tau^2}}\ d\tau \\
&= \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_0^2)}} e^{-\frac{(x-(\mu_1+\mu_0))^2}{2(\sigma_1^2+\sigma_0^2)}}
\end{aligned}
$$

(because the quantity inside the integral is 1 for a Gaussian density function.)

APPENDIX-B: Proof of Theorem 1

Convolution of Gaussian density with respect to the mean is shown below :

$$
\tilde{c}(x, \theta_0) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-(\mu_1-\tau)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{\tau^2}{2\sigma_0^2}} d\tau \tag{.2}
$$

Now, consider Eq. (.1). Substituting $\mu_0 = 0$ and replacing τ with $-\tau$, we get

$$
c(x, \theta_0) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x+\tau-\mu_1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{\tau^2}{2\sigma_0^2}} d\tau \tag{.3}
$$

From Eqs. (.2) and (.3), we can see that convolution of a Gaussian density function with a Gaussian density with zero mean is equivalent to convolving the function with respect to mean.

APPENDIX-C: Derivations for EM updates

For simplicity, we show the derivations for EM updates in the fixed kernel case. Lets consider the case where a fixed Gaussian kernel with parameters μ_0 and σ_0 which will be used to convolve each component of the GMM. We know that

$$log\ p(\mathcal{X}, \mathcal{Z}|\Theta) = log \prod_{j=1}^{n}\ p(x^{(j)}|z^{(j)}, \Theta)\ \cdot\ p(z^{(j)}) \tag{.4}$$

For the j^{th} data point, we have

$$\begin{aligned} & p(x^{(j)}|z^{(j)}, \Theta)\ \cdot\ p(z^{(j)}) \\ & = \prod_{i=1}^{k} \left[\frac{1}{\sqrt{2\pi(\sigma_i^2 + \sigma_0^2)}} e^{-\frac{(x-(\mu_i+\mu_0))^2}{2(\sigma_i^2+\sigma_0^2)}}\ p(z_i^{(j)} = 1) \right]^{z_i^{(j)}} \end{aligned} \tag{.5}$$

Hence,

$$\begin{aligned} log\ p(\mathcal{X}, \mathcal{Z}|\Theta) &= \sum_{j=1}^{n} log\ p(x^{(j)}|z^{(j)}, \Theta) \cdot p(z^{(j)}) \\ = \sum_{j=1}^{n}\sum_{i=1}^{k} log & \left[\frac{1}{\sqrt{2\pi(\sigma_i^2 + \sigma_0^2)}} e^{-\frac{(x-(\mu_i+\mu_0))^2}{2(\sigma_i^2+\sigma_0^2)}}\ p(z_i^{(j)} = 1) \right]^{z_i^{(j)}} \\ = \sum_{j=1}^{n}\sum_{i=1}^{k} z_i^{(j)} & [-log(\sqrt{2\pi(\sigma_i^2 + \sigma_0^2)}) \\ & \qquad - \frac{(x - (\mu_i + \mu_0))^2}{2(\sigma_i^2 + \sigma_0^2)} + log\ \alpha_i] \end{aligned} \tag{.6}$$

Expectation Step
For this step, we need to compute the Q-function which is the expected value of Eq. (.6) with respect to the hidden variables.

$$\begin{aligned} Q(\Theta|\Theta^{(t)}) &= E_z \left[log\ p(\mathcal{X}, \mathcal{Z}|\Theta)|\mathcal{X}, \Theta^{(t)} \right] \\ &= \sum_{j=1}^{n}\sum_{i=1}^{k} E_z[z_i^{(j)}][-log(\sqrt{2\pi(\sigma_i^2 + \sigma_0^2)}) \\ & \qquad - \frac{(x - (\mu_i + \mu_0))^2}{2(\sigma_i^2 + \sigma_0^2)} + log\ \alpha_i] \end{aligned} \tag{.7}$$

To compute the Expected value of the hidden variables ($w_i^{(j)}$),

$$
\begin{aligned}
w_i^{(j)} &= E_z[z_i^{(j)}] = \sum_{c=0}^{1} c * p(z_i^{(j)} = c|\Theta^{(t)}, x^{(j)}) \\
&= \frac{p(x^{(j)}|\Theta^{(t)}, z_i^{(j)} = 1)\ p(z_i^{(j)} = 1|\Theta^{(t)})}{p(x^{(j)}|\Theta^{(t)})} \\
&= \frac{\frac{1}{\sqrt{(\sigma_i^2+\sigma_0^2)}} e^{-\frac{(x-(\mu_i+\mu_0))^2}{2(\sigma_i^2+\sigma_0^2)}} \alpha_i^{(t)}}{\sum_{m=1}^{k} \frac{1}{\sqrt{(\sigma_m^2+\sigma_0^2)}} e^{-\frac{(x-(\mu_m+\mu_0))^2}{2(\sigma_m^2+\sigma_0^2)}} \alpha_m^{(t)}}
\end{aligned}
$$

Maximization Step
The maximization step is given by the following equation :

$$\frac{\partial}{\partial \Theta_i} Q(\Theta|\widehat{\Theta}(t)) = 0$$

where Θ_i are the parameters of the i^{th} component. Due to the assumption made that each data point comes from a single component, solving the above equation becomes trivial. The updates for the maximization step in the case of GMMs are given as follows:

$$
\begin{aligned}
(\mu_i + \mu_0) &= \frac{\sum_{j=1}^{n} w_i^{(j)} x^{(j)}}{\sum_{j=1}^{n} w_i^{(j)}} \\
(\sigma_i^2 + \sigma_0^2) &= \frac{\sum_{j=1}^{n} w_i^{(j)} (x^{(j)} - (\mu_i + \mu_0))^2}{\sum_{j=1}^{n} w_i^{(j)}} \qquad (.8) \\
\alpha_i &= \frac{1}{n} \sum_{j=1}^{n} w_i^{(j)}
\end{aligned}
$$

References

[1] G. J. McLachlan, K. E. Basford, Mixture models: Inference and applications to clustering, Marcel Dekker, New York, 1988.

[2] A. P. Demspter, N. A. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society Series B 39 (1977) 1–38.

[3] R. A. Redner, H. F. Walker, Mixture densities, maximum likelihood and the EM algorithm, SIAM Review 26 (1984) 195–239.

[4] L. Xu, M. I. Jordan, On convergence properties of the EM algorithm for gaussian mixtures, Neural Computation 8 (1) (1996) 129–151.

[5] L. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains, Annals of Mathematical Statistics 41 (1970) 164–171.

[6] J. A. Bilmes, A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models, Tech. rep., U. C. Berkeley (April 1998).

[7] J. D. Banfield, A. E. Raftery, Model-based gaussian and non-gaussian clustering, Biometrics 49 (1993) 803– 821.

[8] K. Nigam, A. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Machine Learning 39 (2-3) (2000) 103 – 134.

[9] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobworld: Image segmentation using expectation-maximization and its application to image querying, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (8) (2002) 1026–1038.

[10] R. Shumway, D. Stoffer, An approach to time series smoothing and forecasting using the EM algorithm, Journal of Time Series Analysis 3 (4) (1982) 253–264.

[11] G. McLachlan, D. Peel, Finite Mixture Models, John Wiley and Sons, 2000.

[12] N. Ueda, R. Nakano, Deterministic annealing EM algorithm, Neural Networks 11 (2) (1998) 271–282.

[13] F. Pernkopf, D. Bouchaffra, Genetic-based EM algorithm for learning gaussian mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1344–1348.

[14] N. Ueda, R. Nakano, Z. Ghahramani, G. Hinton, SMEM algorithm for mixture models, Neural Computation 12 (9) (2000) 2109–2128.

[15] M. Figueiredo, A. Jain, Unsupervised learning of finite mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (3) (2002) 381–396.

[16] J. J. Verbeek, N. Vlassis, B. Krose, Efficient greedy learning of gaussian mixture models, Neural Computation 15 (2) (2003) 469–485.

[17] R. M. Neal, G. E. Hinton, A new view of the EM algorithm that justifies incremental, sparse and other variants, in: M. I. Jordan (Ed.), Learning in Graphical Models, Kluwer Academic Publishers, 1998, pp. 355–368.

[18] J. Q. Li, Estimation of mixture models, Ph.D. thesis, Department of Statistics,Yale University (1999).

[19] T. Hastie, R. Tibshirani, Discriminant analysis by Gaussian mixtures, Journal of the Royal Statistical Society series B 58 (1996) 158–176.

[20] S. J. Roberts, D. Husmeier, I. Rezek, W. Penny, Bayesian approaches to gaussian mixture modeling, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (11) (1998) 1133 – 1142.

[21] B. Zhang, C. Zhang, X. Yi, Competitive EM algorithm for finite mixture models, Pattern Recognition 37 (2004) 131–144.

[22] J. Beran, G. Mazzola, Visualizing the relationship between two time series by hierarchical smoothing, Journal of Computational and Graphical Statistics 8 (2) (1999) 213–238.

[23] S. F. Chen, J. T. Goodman, An empirical study of smoothing techniques for language modeling, in: In Proceedings of the 34th Annual Meeting of the ACL, 1996, pp. 310–318.

[24] A. Blake, A. Zisserman, Visual reconstruction, MIT Press,Cambridge,MA., 1987.

[25] C. Chu, I. Glad, F. Godtliebsen, J. Marron, Edge-preserving smoothers for image processing, Journal of the American Statistical Association 93 (442) (1998) 526–541.

[26] S. Richter, R. DeCarlo, Continuation methods: Theory and applications, IEEE Transactions on Circuits and Systems 30 (6) (1983) 347–352.

[27] D. M. Dunlavy, D. P. O'leary, D. Klimov, D. Thirumalai, Hope: A homotopy optimization method for protein structure prediction, Journal of Computational Biology 12 (10) (2005) 1275–1288.

[28] S. H. Teng, Coarsening, sampling, and smoothing: Elements of the multilevel method, Algorithms for Parallel Processing, IMA Volumes in Mathematics and its Applications 105, Springer Verlag (1999) 247–276.

[29] C. Shao, R. Byrd, E. Eskow, R. Schnabel, Global optimization for molecular clusters using a new smoothing approach, Journal of Global Optimization 16 (2) (2000) 167–196.

[30] G. McLachlan, T. Krishnan, The EM Algorithm and Extensions, John Wiley and Sons, New York, 1997.

[31] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM Journal on Scientific Computing 20 (1) (1999) 359–392.

[32] S. Dasgupta, L. J. Schulman, A two-round variant of em for gaussian mixtures, Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence (2000) 152–159.

[33] Z. B. Tang, Adaptive partitioned random search to global optimization, IEEE Transactions on Automatic Control 39 (11) (1994) 2235–2244.