Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Learning mixture models via component-wise parameter smoothing

Chandan K. Reddy[a,*], Bala Rajaratnam[b]

[a] *Department of Computer Science, Wayne State University, Detroit, MI, United States*

[b] *Department of Statistics, Stanford University, Stanford, CA, United States*

ARTICLE INFO

Article history:
Received 30 January 2008
Received in revised form 23 April 2009
Accepted 23 April 2009
Available online 3 May 2009

ABSTRACT

The task of obtaining an optimal set of parameters to fit a mixture model has many applications in science and engineering domains and is a computationally challenging problem. A novel algorithm using a convolution based smoothing approach to construct a hierarchy (or family) of smoothed log-likelihood surfaces is proposed. This approach smooths the likelihood function and applies the EM algorithm to obtain a promising solution on the smoothed surface. Using the most promising solutions as initial guesses, the EM algorithm is applied again on the original likelihood. Though the results are demonstrated using only two levels, the method can potentially be applied to any number of levels in the hierarchy. A theoretical insight demonstrates that the smoothing approach indeed reduces the overall gradient of a modified version of the likelihood surface. This optimization procedure effectively eliminates extensive searching in non-promising regions of the parameter space. Results on some benchmark datasets demonstrate significant improvements of the proposed algorithm compared to other approaches. Empirical results on the reduction in the number of local maxima and improvements in the initialization procedures are provided.

Published by Elsevier B.V.

1. Introduction

Finite mixture modeling is a well-studied model-based clustering algorithm in the field of statistical pattern recognition (McLachlan and Basford, 1988; Bhninga et al., 2007). One of the main problems in fitting mixture models to observed data is parameter estimation, for which the *Expectation-Maximization* (EM) algorithm provides a reasonable set of estimates (Demspter et al., 1977; Redner and Walker, 1984). Traditional optimization approaches such as steepest descent, conjugate gradient, or Newton–Raphson methods are too complicated for use in solving this problem (Xu and Jordan, 1996). The EM algorithm has become a popular method since it takes advantage of problem specific properties.

Given the number of components and an initial set of parameters, the EM algorithm computes the optimal estimates of the parameters that locally maximize the likelihood of the data. However, the main problem with the EM algorithm is that it is a '*greedy*' method which is very sensitive to the given initial set of parameter values (Biernacki et al., 2003). The main reasons that motivated the new algorithm presented in this paper are as follows (see also Reddy et al. (2008)):

- The EM algorithm converges to a local maximum of the likelihood function very quickly.
- There are often several other promising local optimal solutions in the vicinity of the solutions obtained from methods that provide good initial guesses of the solution.

[*] Corresponding address: Department of Computer Science, Wayne State University, 5143 Cass Avenue, 452 State Hall, Detroit, MI - 48202, United States. Tel.: +1 313 577 9005; fax: +1 313 577 6868.

E-mail address: reddy@cs.wayne.edu (C.K. Reddy).

0167-9473/$ – see front matter. Published by Elsevier B.V.
doi:10.1016/j.csda.2009.04.012

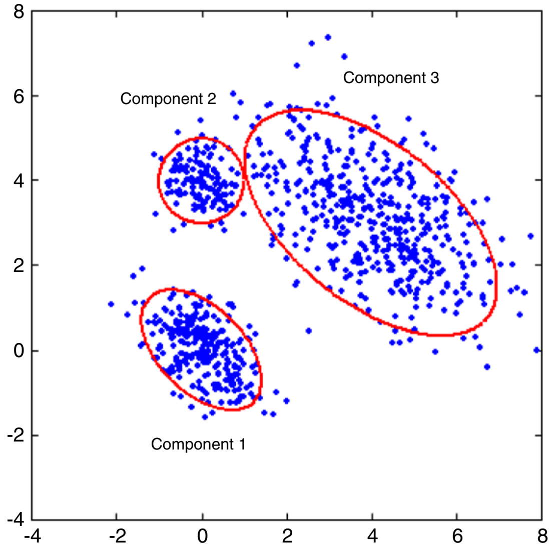


Fig. 1. Data generated by three Gaussian components. The problem of learning Gaussian mixture models is to obtain the parameters of these Gaussian components and the membership probabilities of each datapoint.

- Model selection criteria usually assumes that the global optimal solution of the log-likelihood function can be obtained. However, achieving this is computationally intractable.
- Some regions in the search space do not contain any promising solutions. The promising and non-promising regions co-exist, and it often becomes challenging to avoid wasting computational resources to search in non-promising regions.

Of all the concerns highlighted above, the fact that local maxima are not uniformly distributed makes it important for us to develop algorithms that help in avoiding search in non-promising regions (Reddy and Rajaratnam, 2008). Indeed, more focus needs to be given to searching promising subspaces and obtaining promising initial estimates. One way to achieve this is by smoothing the surface, obtaining promising regions and then gradually tracing back these solutions onto the original surface. In this work, we develop a hierarchical smoothing algorithm for the mixture modeling problem using a convolution-based approach. We consider the problem of learning parameters of Gaussian Mixture Models (GMM). Fig. 1 shows data generated by three Gaussian components with different means and variances. Note that every data point has a probabilistic (or soft) membership that gives the probability with which it belongs to each of the components. The data points that belong to component 1 will have high probability of membership for component 1. On the other hand, data points belonging to components 2 and 3 are not well separated. The problem of learning Gaussian mixture models involves not only estimating the parameters of these components but also finding the probabilities with which each data point belongs to these components.

In this paper, we demonstrate the use of "*hierarchical smoothing*" in the context of parameter estimation. Ideally, smoothing procedures should satisfy the following properties:

- Reduce the overall magnitude of the gradient of the surface.
- Reduce the number of local maxima (or minima).
- Smooth different regions of the search space adaptively.
- Avoid over smoothing which might make the surface too flat.

The rest of this 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 concepts of maximum likelihood estimation (or MLE) and the Expectation Maximization (EM) algorithm. Section 4 describes different smoothing strategies and gives the corresponding smooth-EM updates. Section 5 discusses the proposed smoothing framework and describes some of the implementation details. Section 6 shows experimental results of our algorithm on both synthetic and real datasets. Finally, Section 7 concludes our discussion with future research directions.

2. Relevant background

EM based methods have been successfully applied to solve a wide range of problems that arise in a wide range of fields such as pattern recognition (Baum et al., 1970; Bilmes, 1998), clustering (Banfield and Raftery, 1993), information retrieval (Nigam et al., 2000), computer vision (Carson et al., 2002), data mining (Shumway and Stoffer, 1982) etc. Many EM variants have been extensively used for learning mixture models and several researchers have proposed new techniques that provide good initialization values. Generic techniques like deterministic annealing (Rose, 1998; Ueda and Nakano, 1998), and genetic algorithms (Pernkopf and Bouchaffra, 2005; Martnez and Vitri, 2000) have been applied to obtain a good set of parameters. Though, these techniques have asymptotic guarantees, they are time consuming and hence cannot be used in most practical applications. Some problem specific algorithms like split and merge EM (Ueda et al., 2000), component-wise EM (Figueiredo and Jain, 2002), greedy learning (Verbeek et al., 2003), incremental version for sparse representations (Neal and Hinton, 1998), and parameter space grid (Li, 1999) have also been proposed in the literature. In spite of the high computational cost associated with some of these methods, very little effort has been taken to explore promising subspaces within the larger parameter space. Most of these algorithms eventually apply the EM algorithm to move to a locally maximal set of parameters on the likelihood surface. Simpler practical approaches like running EM from several random initializations, and then choosing the final estimate as the local maximum that yields the highest value of the likelihood have also been successful to a certain extent (Hastie and Tibshirani, 1996; Roberts et al., 1998). Though some of these methods apply other additional mechanisms (like perturbations (Elidan et al., 2002)) to escape out of local optimal solutions, more systematic methods are yet to be investigated for searching the subspace. Recently, a dynamical system based formulation has shown much promise (Reddy et al., 2008).

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 (Shumway and Stoffer, 1982; Beran and Mazzola, 1999). In the field of natural language processing, smoothing techniques are also used for adjusting maximum likelihood estimates to produce more accurate probabilities for language models (Chen and Goodman, 1996). Convolution based smoothing approaches are predominantly used in the field of digital image processing for image enhancement by noise removal (Blake and Zisserman, 1987; Chu et al., 1998). Other variants of smoothing techniques include continuation methods (Richter and DeCarlo, 1983; Dunlavy et al., 2005) which are used successfully in various applications. Different multi-level procedures other than smoothing and its variants are clearly illustrated in Teng (1999).

From the optimization perspective, smoothing procedures help in reducing the ruggedness of the surface and helps local methods in preventing the "local maxima problem". They have been used for the structure prediction of molecular clusters (Shao et al., 2000). The smoothing procedure proposed in this paper obtains a hierarchy of smooth surfaces with fewer and fewer local maxima. Promising initial points can be obtained by "tracing back" promising solutions at each level. This yields an initialization procedure that has the capability of avoiding searching in non-promising regions of the solution space. Our approach assumes that the number of components in our mixture model is known beforehand. In summary, the main contributions of this paper are:

- Develop a hierarchical parameter smoothing algorithm using convolution based techniques.
- Theoretically show that the expected value of the gradient is reduced for a modified (smoothed) version of the log-likelihood surface.
- Show that the "density based smoothing" is equivalent to "component-wise parameter smoothing" in the case of Gaussian Mixture Models.
- Demonstrate the reduction in the number of unique local maxima empirically.
- Show that smoothing helps in obtaining a promising set of initial parameter values.

Fig. 2 compares the conventional approach with the smoothing approach. In the traditional approach, a global optimization 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 the optimal parameter set on the original log-likelihood surface. Since the smoothed 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, the latter being often computationally expensive.

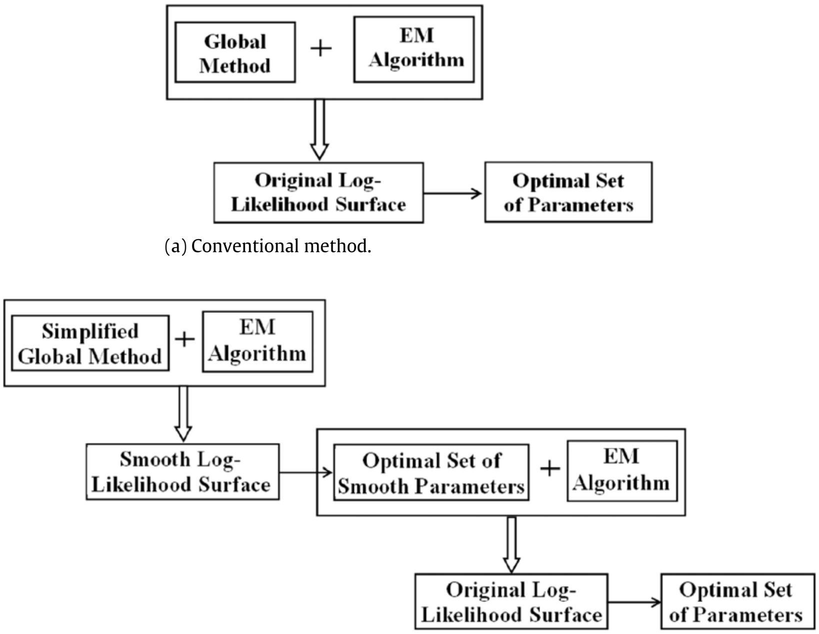
3. Preliminaries

We will now introduce some necessary preliminaries on mixture models, EM algorithm and convolution kernels. Table 1 gives the notations used in this paper.

3.1. Mixture models

Let us assume that there are k Gaussian components in the mixture model. The form of the probability density function is as follows:

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


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.

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 set
θ_i	Parameters of *i*th component
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

where $x = [x_1, x_2, \ldots, x_d]^{\mathrm{T}}$ is the feature vector of d dimensions. The α_i's represent the *mixing weights*. The symbol Θ represents the parameter set $(\alpha_1, \alpha_2, \ldots \alpha_k, \theta_1, \theta_2, \ldots, \theta_k)$ and p is a d-variate Gaussian density parameterized by θ_i (i.e. μ_i and Σ_i):

$$p(x|\theta_i) = \frac{|\Sigma_i|^{-\frac{1}{2}}}{(2\pi)^{d/2}} \mathrm{e}^{-\frac{1}{2}(x-\mu_i)^{\mathrm{T}}\Sigma^{-1}(x-\mu_i)}. \tag{2}$$

Also, it should be noticed that being probabilities α_i must satisfy

$$0 \le \alpha_i \le 1, \quad \forall i = 1, \ldots, k, \quad \text{and} \quad \sum_{i=1}^{k} \alpha_i = 1. \tag{3}$$

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

$$\log\ p(\mathcal{X}|\Theta) = \log \prod_{j=1}^{n} p(x^{(j)}|\Theta) = \sum_{j=1}^{n} \log \sum_{i=1}^{k} \alpha_i\, p(x^{(j)}|\theta_i). \tag{4}$$

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

$$\widehat{\Theta}_{MLE} = \arg\max_{\Theta}\ \{\log\ p(\mathcal{X}|\Theta)\}. \tag{5}$$

where $\tilde{\Theta}$ indicates the entire parameter space. Since, this MLE cannot be found analytically for mixture models, one has to rely on iterative procedures that can find the global maximum of $\log\ p(\mathcal{X}|\Theta)$. The EM algorithm described in the next section has been used successfully to find the local maximum of such a function (McLachlan and Krishnan, 1997).

3.2. Expectation maximization

The EM algorithm assumes $\mathcal{X}$ to be the *observed* data. The missing part, termed as *hidden* data, is a set of n labels $\mathcal{Z} = \{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(n)}\}$ associated with the n samples, indicating which component produced each sample (McLachlan and Krishnan, 1997). Each label $\mathbf{z}^{(j)} = [z_1^{(j)}, z_2^{(j)}, \ldots, z_k^{(j)}]$ 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 ith 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 known is given as,

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

$$\log\ p(\mathcal{Y}|\Theta) = \sum_{j=1}^{n}\sum_{i=1}^{k} z_i^{(j)} \log\ [\alpha_i\, p(x^{(j)}|\theta_i)]. \tag{6}$$

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

- **E-Step:** Compute the conditional expectation of the hidden data, 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:

$$Q(\Theta|\widehat{\Theta}(t)) \equiv E_Z[\log\ p(\mathcal{X}, \mathcal{Z})|\mathcal{X}, \widehat{\Theta}(t)]. \tag{7}$$

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

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

where the last equation follows from Bayes law (α_i is the a priori probability that $z_i^{(j)} = 1$, while $w_i^{(j)}$ is the 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:

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

3.3. EM for GMMs

Several variants of the EM algorithm have been extensively used to solve the above problem. The convergence properties of the EM algorithm for Gaussian mixtures are thoroughly discussed in Xu and Jordan (1996). The Q-*function* for GMM is given by:

$$Q(\Theta|\widehat{\Theta}(t)) = \sum_{j=1}^{n}\sum_{i=1}^{k} w_i^{(j)} \left[\log\left(\frac{|\Sigma|^{-\frac{1}{2}}}{(2\pi)^{\frac{d}{2}}}\right) - \frac{1}{2}(x^{(j)} - \mu_i)^{\mathrm{T}}\Sigma^{-1}(x^{(j)} - \mu_i) + \log\ \alpha_i\right], \tag{9}$$

where

$$w_i^{(j)} = \frac{\widehat{\alpha_i}(t)|\widehat{\Sigma_i}(t)|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(x^{(j)}-\widehat{\mu_i}(t))^{\mathrm{T}}|\widehat{\Sigma_i}(t)|^{-1}(x^{(j)}-\widehat{\mu_i}(t))}}{\sum_{i=1}^{k}\widehat{\alpha_i}(t)|\widehat{\Sigma_i}(t)|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(x^{(j)}-\widehat{\mu_i}(t))^{\mathrm{T}}|\widehat{\Sigma_i}(t)|^{-1}(x^{(j)}-\widehat{\mu_i}(t))}}. \tag{10}$$

The maximization step is given by the following equation:

$$\frac{\partial}{\partial\Theta_k} Q(\Theta|\widehat{\Theta}(t)) = 0 \tag{11}$$

where Θ_k is the parameter set for the kth component. Since the posterior probabilities in the E-step now appear in the Q-function as given constants, and therefore resembling the Gaussian likelihood when the components are pre-specified, maximizing this function in the M-step becomes trivial. The updates for the maximization step in the case of GMMs are given as follows:

$$\begin{aligned}
\mu_i(t+1) &= \frac{\sum_{j=1}^{n} w_i^{(j)} x^{(j)}}{\sum_{j=1}^{n} w_i^{(j)}}, \\
\Sigma_i(t+1) &= \frac{\sum_{j=1}^{n} w_i^{(j)} (x^{(j)} - \mu_i(t+1))(x^{(j)} - \mu_i(t+1))^{\mathrm{T}}}{\sum_{j=1}^{n} w_i^{(j)}}, \\
\alpha_i(t+1) &= \frac{1}{n} \sum_{j=1}^{n} w_i^{(j)}.
\end{aligned} \tag{12}$$

The convergence properties of the EM algorithm for Gaussian mixtures are thoroughly discussed in Xu and Jordan (1996). One of the main challenges of using the EM algorithm is the initialization step. The final result obtained using the EM algorithm 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 caused by initial parameters. The approach taken is termed "*convolution based smoothing*" and is described in the next section.

3.4. Convolution kernels

We will now introduce some preliminaries on convolution kernels. A convolution kernel that yields closed form solutions in both the E and M steps are useful for smoothing a mixture model. Three widely used kernels are shown in Fig. 3. We choose to use a Gaussian kernel for smoothing the original log-likelihood function for the following inter-related reasons:

- When the underlying distribution is assumed to be generated from Gaussian components, Gaussian kernels are more effective and easy to handle.
- 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 according to scale will be much easier.
- The Gaussian kernel is smooth compared to the triangular or step kernels.

We will now proceed to formally define convolution kernels, and Gaussian convolution kernels in particular. Consider a density $p(x|\theta_1)$ and a kernel $g(x)$, the density function $p(x|\theta_1)$ convolved with the kernel $g(x)$ is defined as follows:

$$p^*(x|\theta) \equiv p(x|\theta_1) \otimes g(x) \equiv \int p(x-\tau|\theta_1) g(\tau) \mathrm{d}\tau. \tag{13}$$

When $p(x|\theta_1)$ and $g(x)$ are both Gaussian with parameters (μ_1, σ_1^2) and (μ_0, σ_0^2) respectively, i.e.,

$$p(x|\theta_1) = \frac{1}{\sigma_1\sqrt{2\pi}} \mathrm{e}^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} \tag{14}$$

$$g(x) = \frac{1}{\sigma_0\sqrt{2\pi}} \mathrm{e}^{-\frac{(x-\mu_0)^2}{2\sigma_0^2}} \tag{15}$$

then $p^*(x|\theta)$ is Gaussian with mean parameter $\mu_1 + \mu_0$ and variance parameter $\sigma_1^2 + \sigma_0^2$. We state this formally in the following lemma.

Lemma 1 (*Convolution of Gaussians*). *For $p(x|\theta_1)$ and $g(x|\mu_0, \sigma_0)$ given as in Eqs. (14) and (15), then*

$$p^*(x|\theta) = \frac{1}{\sqrt{2\pi(\sigma_1^2 + \sigma_0^2)}} \mathrm{e}^{-\frac{(x-(\mu_1+\mu_0))^2}{2(\sigma_1^2+\sigma_0^2)}}. \tag{16}$$

The proof for this lemma is given in Appendix.

From the above result, one can see that when the original Gaussian density function is convolved with a Gaussian kernel of zero mean, only the variance parameter changes. In the context of smoothing based nonlinear optimization, reducing

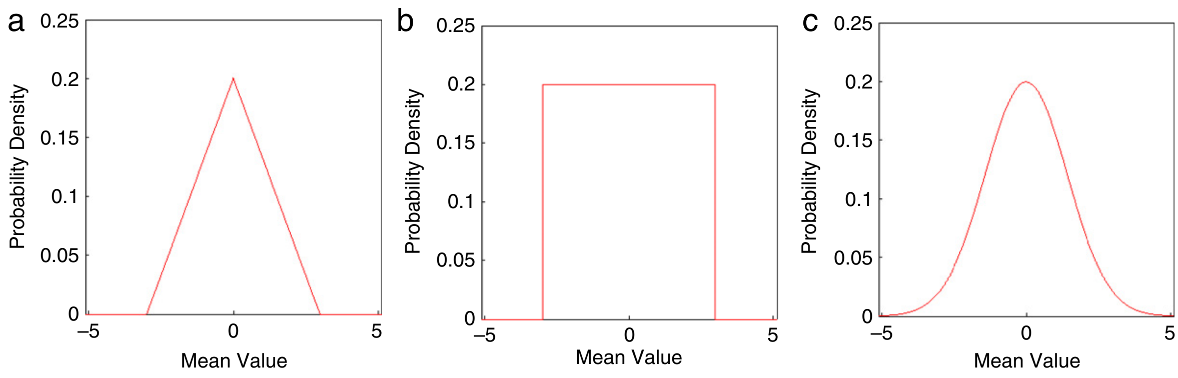


Fig. 3. Different convolution kernels. (a) Triangular function. (b) Step function and (c) Gaussian function.

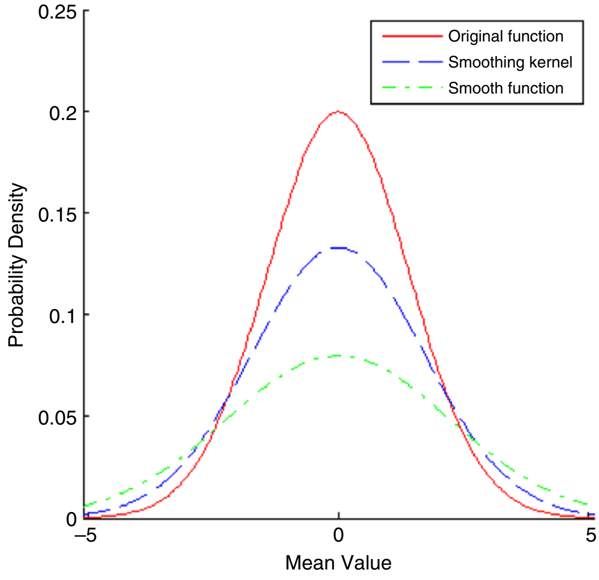


Fig. 4. The effects of smoothing a Gaussian density function with a Gaussian kernel.

the peaks is a very useful property which can be achieved by changing the variance parameter. Also, shifting the mean is not desired and hence we will always prefer to use a kernel with $\mu_0 = 0$ to perform our convolutions. The quantity σ_0 is called a "kernel parameter" and the the choice of σ_0 in practical situations will be discussed later. We will denote $p^*(x|\theta_1)$ as $c(\breve{x}, \theta_1)$ to emphasize the fact that the convolution affects the argument of the density function as defined in Eq. (13). We will sometimes refer to this as "data smoothing" or "density smoothing" as the convolution smoothes the density of the data (given by x). Convolving two Gaussians to obtain another Gaussian is shown graphically in Fig. 4.

4. Smoothing the log-likelihood surface

In principle, the overall log-likelihood surface can be convolved using a Gaussian kernel directly. Doing so directly however is not a feasible approach because of the following reasons:

- It results in an analytic expression that is not easy to work with, and computing the EM updates become rather cumbersome.
- It is computationally very expensive.
- Different regions of search space must be smoothed differently. Choosing parameters to do this task is difficult using this approach.

To avoid the first issue, we exploit the structure of the problem. Since the log-likelihood surface is obtained from individual densities, smoothing each component's individual density function will smooth the overall log-likelihood surface. This will also give flexibility in choosing the kernel parameters (to be discussed in following subsection). Fig. 5 shows the block diagram of the smoothing procedure. The kernel parameters are chosen from the initial set of parameters and the

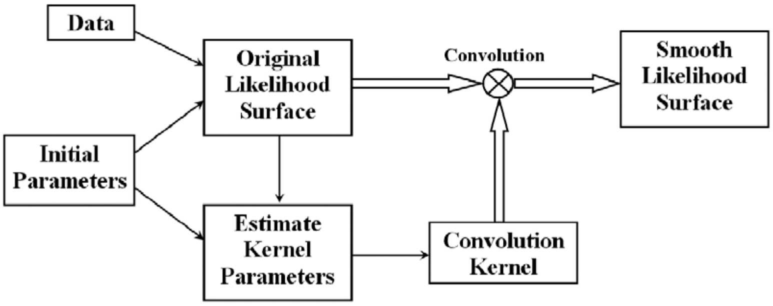

Data
Initial Parameters
Original Likelihood Surface
Estimate Kernel Parameters
Convolution
Convolution Kernel
Smooth Likelihood Surface

Fig. 5. Block diagram of the smoothing approach. A 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.

original log-likelihood surface. The kernel is then convolved with the original log-likelihood surface to obtain the smooth log-likelihood surface.

The new component-wise smoothed density (p^*), is defined as follows:

$$p^*(x|\Theta) = \sum_{i=1}^{k} \alpha_i \, p^*(x|\theta_i). \tag{17}$$

The corresponding smooth log-likelihood function is given by:

$$f^*(\mathcal{X}, \Theta) = \sum_{j=1}^{n} \log \sum_{i=1}^{k} \alpha_i \, p^*(x^{(j)}|\theta_i). \tag{18}$$

4.1. Kernel parameters

Obtaining the parameters for the smoothing kernel $g(x)$ is a non-trivial task. The parameters of the smoothing kernel can be chosen to be static, i.e., independent of the parameters of the individual components. Such fixed kernels will be effective when the underlying distribution is from similar components. One of the main problems of using a fixed kernel is that, some of the components may not be smoothed while others may be excessively smoothed. Since, the Gaussian kernel has the property that the convolution leads to addition in the parameters, the convolution described in Eq. (16) can also be treated as *additive smoothing*. To avoid the problems of fixed kernel smoothing, we introduce the concept of "variable kernel smoothing". Each component will be treated differently and smoothed according to the existing parameter values of each component. This smoothing strategy is much more flexible and works well in practice. Since, 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 is a function of σ_i. Both these approaches do not allow for smoothing the mixing weight parameters (α_i's). From a practical point of view, we chose to have a single parameter for the variable kernel case. For each of the components, the current σ_i value is multiplied with this single parameter. This will make it easy to run the corresponding algorithm because there is only one smoothing parameter that needs to be optimized. Having different parameters for different components might yield better results experimentally, but will be a cumbersome task to the user. Hence, we decided to use a single parameter. Simple experiments are conducted to decide upon this single smoothing parameter for any given data set and hence our algorithm is computationally efficient even for large data sets.

The introduction of the smoothing does not lead to any additional identifiability problems in the Gaussian mixture model as the resulting effect is an addition to the variance parameter by a known constant hence there is a one-to-one mapping from the parameter space of the smoothed model to that of the unsmoothed or original model.

4.2. EM updates

For both of the above mentioned (additive and multiplicative) smoothing kernels, we state the corresponding version of the EM updates below. The complete derivations of these EM equations for the case of "fixed kernel smoothing" is given in Appendix. The Q-*function* of the EM algorithm applied to the smoothed log-likelihood surface in the univariate case is given by:

$$Q(\Theta|\widehat{\Theta}(t)) = \sum_{j=1}^{n}\sum_{i=1}^{k} w_i^{(j)} \left[\log \frac{1}{\sqrt{2\pi(\tilde{\sigma}_i^2)}} - \frac{(x^{(j)} - \tilde{\mu}_i)^2}{2\tilde{\sigma}_i^2} + \log \alpha_i \right], \tag{19}$$

where

$$w_i^{(j)} = \frac{\frac{\alpha_i(t)}{\tilde{\sigma}_i} \mathrm{e}^{-\frac{1}{2\tilde{\sigma}_i^2}(x^{(j)} - \tilde{\mu}_i(t))^2}}{\sum_{i=1}^{k} \frac{\alpha_i(t)}{\tilde{\sigma}_i} \mathrm{e}^{-\frac{1}{2\tilde{\sigma}_i^2}(x^{(j)} - \tilde{\mu}_i(t))^2}} \tag{20}$$

and $\tilde{\Theta}$ represents the smoothed parameters. The updates for the maximization step in the case of GMMs are once more given as follows:

$$\tilde{\mu}_i(t+1) = \frac{\sum_{j=1}^{n} w_i^{(j)} x^{(j)}}{\sum_{j=1}^{n} w_i^{(j)}}, \tag{21}$$

$$\tilde{\sigma}_i^2(t+1) = \frac{\sum_{j=1}^{n} w_i^{(j)} (x^{(j)} - (\tilde{\mu}_i(t+1)))^2}{\sum_{j=1}^{n} w_i^{(j)}}, \tag{22}$$

$$\tilde{\alpha}_i(t+1) = \frac{1}{n} \sum_{j=1}^{n} w_i^{(j)}. \tag{23}$$

It is easily observed that we retain the same form of the updates as in the non-smoothed version (see Eq. (12)).

4.3. Properties of smoothing approach

Our main contribution regarding convolution based smoothing techniques are discussed in this section. First, we will show that in the case of Gaussian mixture models, component-wise smoothing with respect to the argument of the density (i.e. data smoothing) is equivalent to smoothing with respect to the parameters. Second, we will show that the component-wise smoothing indeed reduces two measures of the overall gradient of the likelihood surface and hence smooths it.

Lemma 2 (*Density Smoothing*). *Convolution of a Gaussian density function with parameters μ_1 and σ_1 with respect to the argument of the density function, with a Gaussian kernel with parameters $\mu_0 = 0$ and σ_0 is equivalent to convolving the function with respect to μ_1. i.e.*

$$c(x, \breve{\mu}_1) = c(\breve{x}, \mu_1) \quad \textit{when}\, g(\tau) = \frac{1}{\sqrt{2\pi}\sigma_0} \mathrm{e}^{-\frac{\tau^2}{2\sigma_0^2}}.$$

Proof. See Appendix. □

We now proceed to demonstrate that a modified version of the component-wise smoothed log-likelihood surface has an overall gradient that is lower than that of the original likelihood for the same mixture. Our claim is that the convolution induces a less rugged log-likelihood surface. One possible way to measure the ruggedness of the log-likelihood surface is to consider the gradient or rather the magnitude of the gradient of the log-likelihood function. Two measures of this are

$$\left| \frac{\partial \log\, p(x|\theta)}{\partial \theta} \right| \quad \text{or} \quad \left(\frac{\partial \log\, p(x|\theta)}{\partial \theta} \right)^2.$$

These are of course random quantities and will differ from sample to sample. Hence, two measures of the ruggedness of the surface can be described by either

$$E\left[\left| \frac{\partial \log\, p(x|\theta)}{\partial \theta} \right| \right] \quad \text{or} \quad E\left[\left(\frac{\partial \log\, p(x|\theta)}{\partial \theta} \right)^2 \right].$$

We will work with the latter as this quantity is firstly analytically more tractable and secondly has a special meaning in statistics, namely the Fisher Information matrix. In general, The Fisher Information matrix cannot be obtained analytically for GMMs (Figueiredo and Jain, 2002).

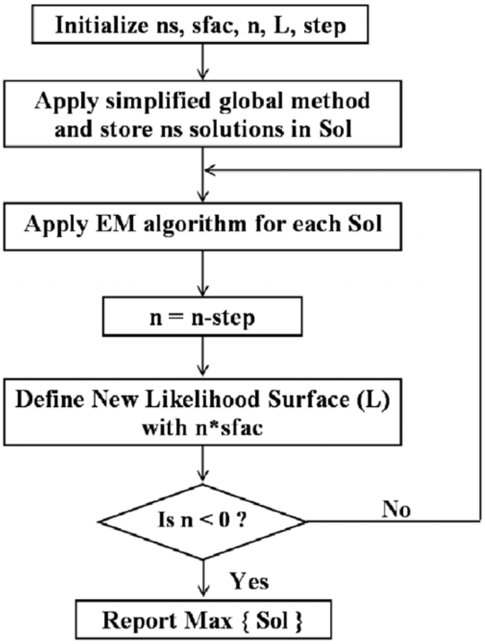

Initialize ns, sfac, n, L, step
Apply simplified global method and store ns solutions in Sol
Apply EM algorithm for each Sol
n = n-step
Define New Likelihood Surface (L) with n*sfac
Is n < 0 ?
No
Yes
Report Max { Sol }

Fig. 6. Flowchart of the smoothing algorithm.

Theorem 3. *The expected value of the squared gradient of the smoothed complete log-likelihood surface is lower than the original complete log-likelihood surface. More precisely,*

$$E\left[\left(\frac{\partial \log\, p(x,z|\theta)}{\partial \mu_i}\right)^2\right] \geq E\left[\left(\frac{\partial \log\, p^*(x,z|\theta)}{\partial \mu_i}\right)^2\right],$$
$$E\left[\left(\frac{\partial \log\, p(x,z|\theta)}{\partial \sigma_i^2}\right)^2\right] \geq E\left[\left(\frac{\partial \log\, p^*(x,z|\theta)}{\partial \sigma_i^2}\right)^2\right].$$
$$\forall i = 1, 2, \ldots, k.$$

Proof. See Appendix. □

We also show that the Theorem 3 above holds true under a different measure of ruggedness of the log-likelihood surface. The above theorem demonstrates the power of component-wise smoothing. It does not however say much about the number of local maxima. Later, we empirically demonstrate that our component-wise smoothing approach reduces the number of local maxima. We also observe that the gradient with respect to α_i $(i = 1, 2, \ldots, k)$ is the same for both the models.

5. Algorithm and implementation

This section describes the smoothing algorithm proposed above in detail and elaborates on some of the implementation details. The basic advantage of the smoothing approach is that a simplified version of the global method can be used to explore fewer promising local maxima on the smoothed surface. These solutions are used as initial guesses for the EM algorithm which is again applied to the next level of smoothing. Smoothing is targeted to help avoid searching in non-promising areas of the parameter space. Fig. 6 gives the flow chart of our smoothing algorithm which is discussed below.

First we introduce certain variables that are used in our algorithm. 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 smoothed (which is usually chosen by trial-and-error). The symbol *ns* denotes the number of solutions that will be traced. The symbol *n* specifies 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 this method. Having many levels might increase the accuracy of the solutions, but it is computationally expensive. On the other hand, having fewer levels is computationally very cheap, but one might have to forgo the quality of the final solution. Deciding these parameters is not only user-specific but also significantly dependent on the data that is being modeled. Algorithm 1 describes the smoothing approach.

The algorithm takes the smoothing factor, number of levels, number of solutions, parameters set and the data as input and computes the global maximum on the log-likelihood surface. The procedure *Smooth* returns the likelihood surface

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:
$step = 1/nl \quad Sfac = Sfac/nl$
$L =$ Smooth($\mathcal{X}$, $\Theta$,$nl$*$Sfac$)
$Sol =$ Global($\mathcal{X}$, $\Theta$, $L$,$ns$)
while $n \geq 0$ do
    $L =$ Smooth($\mathcal{X}$, $\Theta$,$nl$*$Sfac$)
    for i = 1:$ns$ do
        Sol(i) = EM(Sol(i),$\mathcal{X}$,$L$,$\tau$)
    end for
    $nl = nl\text{-}step$
end while
$\widehat{\Theta}_{MLE} = \max\{\text{Sol}\}$
```

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 the EM algorithm on the likelihood surface corresponding to the next level smoothed surface. The EM algorithm also returns *ns* number of solutions corresponding to the *ns* number of initial estimates. At every iteration, a new likelihood surface is constructed with a reduced smoothing factor. This process is repeated until the smoothing factor becomes zero which in turn 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 the multi-level methods proposed in Karypis and Kumar (1999); Dasgupta and Schulman (2000). The main difference is that the dimensionality of the parameter space is not changed during the smoothing (or coarsening) process.

6. Results and discussion

Our algorithm has been tested on three different datasets. The initial values for the centers and the covariances were chosen to be uniformly random. Uniform priors were chosen 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 is generated from a two-dimensional, three-component Gaussian mixture distribution (Ueda and Nakano, 1998) 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. All the three mixtures have uniform priors. The true mixtures with data generated from these three components are shown in Fig. 8. In the third synthetic dataset, more complicated overlapping Gaussian mixtures are considered (Figueiredo and Jain, 2002). It has four components with 1000 data samples (see Fig. 9). 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 $\alpha_4 = 0.1$.

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

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

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 general theoretical way of estimating the amount of reduction in the number of unique local maximum on the likelihood surface and is the subject of ongoing work. 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.

Fig. 10 shows the variation of the number of local maxima with respect to the smoothing factor for different datasets. One can see that if the smoothing factor is increased beyond a certain threshold value (σ_{opt}), the number of local maxima increases rapidly. This might be due to the fact that over-smoothing the surface will make the surface flat, thus making it

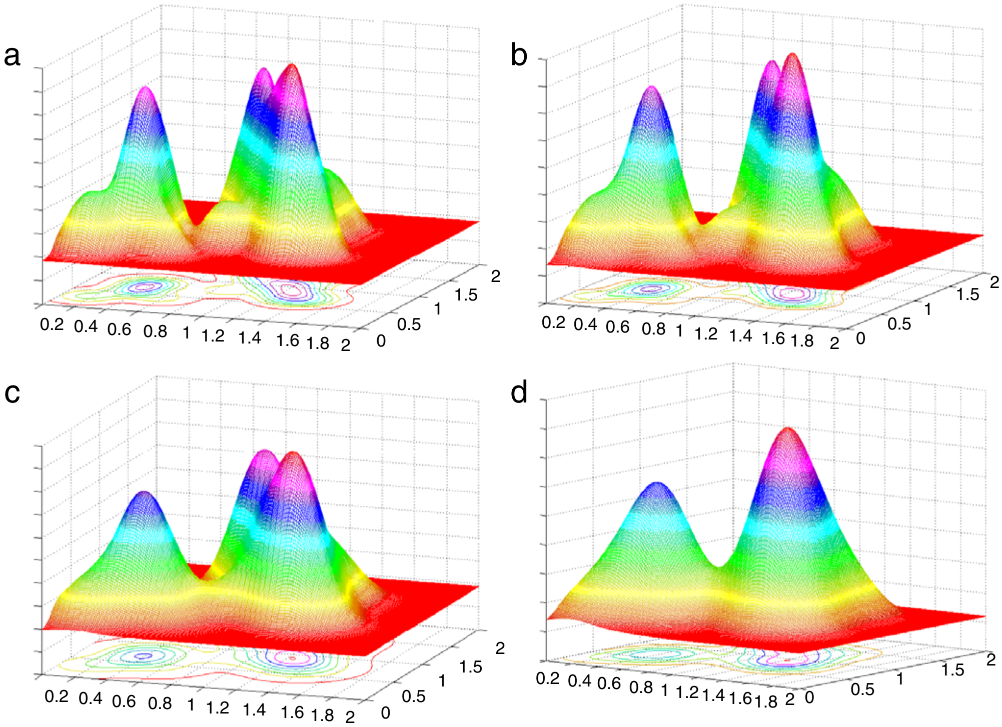


Fig. 7. Various stages during the smoothing process. (a) The original log-likelihood surface which is very rugged. (b)–(c) Intermediate smoothed surfaces. (d) Final smoothed surface with only two local maxima.

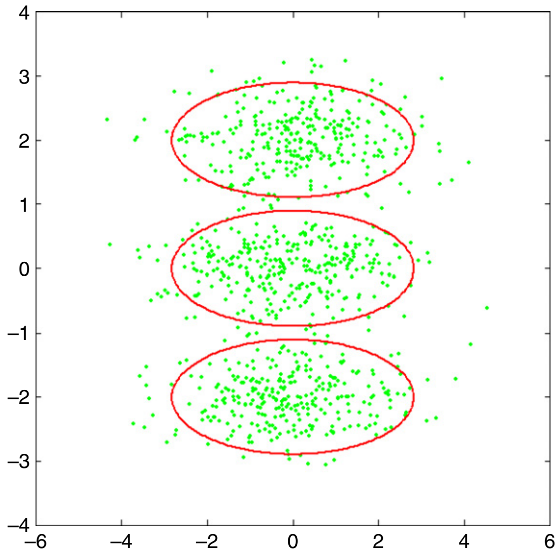


Fig. 8. True mixture of the three Gaussian components with 900 samples.

difficult for the EM algorithm to converge. Experiments were conducted using 1000 random starts and the number of unique local maxima were stored.

6.2. Smoothing for initialization

From an optimization perspective, smoothing the log-likelihood surface also helps in identifying promising solutions. Experiments were conducted using 100 random starts. The average across all the starts is reported (RS + EM). For comparing with the smoothing results, we use the same 100 random starts on the smoothed surface first and use the EM solution obtained as the initial value for the EM algorithm applied on the original surface (Smooth + EM). Table 2 summarizes

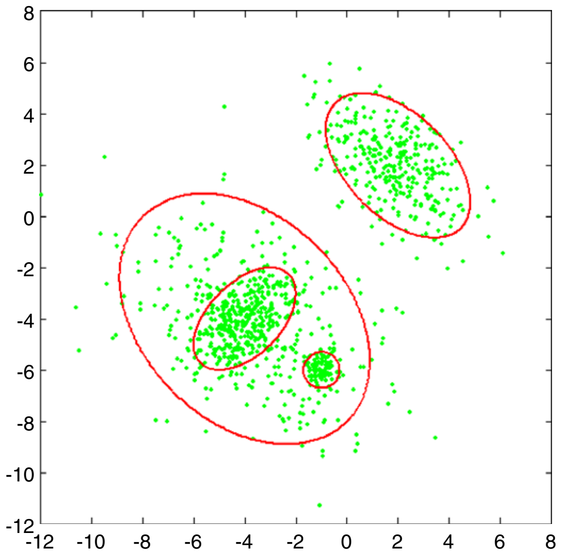
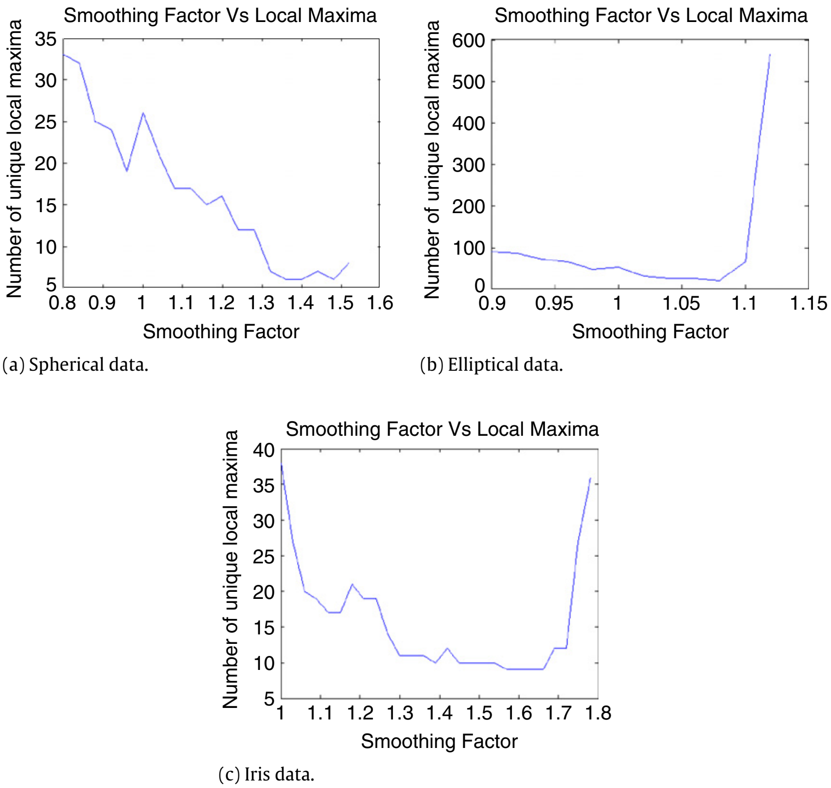


Fig. 9. True mixtures of more complex overlapping Gaussian mixture model with 1000 samples.



(a) Spherical data.

(b) Elliptical data.

(c) Iris data.

Fig. 10. Reduction in the number of local maxima for various datasets.

these results. We used only two levels and track the solution for each level. Hence, from the computational standpoint, the smoothing procedure basically needs twice the amount of time to run the experiments, because we use the EM algorithm twice for each random start. However, based on the mean and standard deviation values obtained with RS + EM using 100 random starts, one can confirm that even if we use twice the number of random starts (let us say 200), we will not be able to achieve promising solutions compared to the solutions obtained using the smoothing procedures. Hence, for every run, the smoothing algorithm tends to converge to more promising set of solutions because of the fact that the smoothed log-likelihood surface helps in reducing the chance of getting stuck in poor local optimal solutions. The two main claims

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

(reduction in the number of local maximum and better initial estimates) about the contributions have thus been justified. More sophisticated global methods like genetic algorithms, simulated annealing, adaptive partitioning (Tang, 1994) etc. and their simplified versions can also be used in combination with our approach. Since the main focus of our work 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 smoothed likelihood surface. A hierarchy of smooth surfaces is constructed and an 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 searching in non-promising areas of the parameter space. Benchmark results demonstrate a significant improvement of the proposed algorithm compared to other existing methods.

The method presented in this paper is one of the basic smoothing approaches on the well studied Gaussian mixture case. There is a plethora of future research directions that can be pursued using this basic concept. The effects of convolving Gaussian components with other kernels and efficient algorithms for choosing the smoothing parameter automatically given the multivariate data are yet to be studied. The idea of combining the hierarchical smoothing procedure with model selection criteria also needs further investigation. An optimal number of levels and the degree of smoothing at each level can be chosen adaptively so that significant distortions of the likelihood surface does not occur. Generic convolution based smoothing strategies can be treated as powerful optimization tools that can enhance search capability significantly and can be applied in the context of other finite mixture models (i.e., beyond the Gaussian mixtures). The use of the above approach in the presence of missing information (Hunt and Jorgensen, 2003) is to be studied in the near future.

Acknowledgements

We would like to thank Prof. Hsiao-Dong Chiang for some useful discussions. We would also like to thank the anonymous reviewers for providing useful comments.

Appendix

Proof of Lemma 1 (*Convolution of Two Gaussians*).

$$\begin{aligned}
p^*(x|\theta) &\equiv p(x|\theta_1) \otimes g(x) \\
&\equiv \int p(x-\tau|\theta_1) g(\tau) \mathrm{d}\tau \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} \mathrm{e}^{-\frac{(x-\tau-\mu_1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_0} \mathrm{e}^{-\frac{(\tau-\mu_0)^2}{2\sigma_0^2}} \mathrm{d}\tau \\
&= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} \mathrm{e}^{-\frac{((x-\tau)-\mu_1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_0} \mathrm{e}^{-\frac{(\tau-\mu_0)^2}{2\sigma_0^2}} \mathrm{d}\tau \\
&= \frac{1}{\sqrt{2\pi}\sigma_1} \frac{1}{\sqrt{2\pi}\sigma_0} \int_{-\infty}^{\infty} \mathrm{e}^{-\frac{\sigma_0^2(\tau-x+\mu_1)^2+\sigma_1^2(\tau-\mu_0)^2}{2\sigma_1^2\sigma_0^2}} \mathrm{d}\tau .
\end{aligned} \tag{A.1}$$

After completing the square in terms of τ and further simplification, we obtain

$$p^*(x|\theta) = \frac{\mathrm{e}^{-\frac{(x-(\mu_1+\mu_0))^2}{2(\sigma_1^2+\sigma_0^2)}}}{\sqrt{2\pi}\sigma_1 \sqrt{2\pi}\sigma_0} \int_{-\infty}^{\infty} \mathrm{e}^{-\frac{(\tau-\mu_\tau)^2}{2\sigma_\tau^2}} \mathrm{d}\tau \tag{A.2}$$

where

$$\mu_\tau = \frac{\mu_0\sigma_1^2 + (x-\mu_1)\sigma_0^2}{(\sigma_1^2+\sigma_0^2)} \quad \text{and} \quad \sigma_\tau^2 = \frac{\sigma_1^2\sigma_0^2}{(\sigma_1^2+\sigma_0^2)}. \tag{A.3}$$

Hence, we have

$$\begin{aligned} p^*(x|\theta) &= \frac{\mathrm{e}^{-\frac{(x-(\mu_1+\mu_0))^2}{2(\sigma_1^2+\sigma_0^2)}}}{\sqrt{2\pi(\sigma_1^2+\sigma_0^2)}} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\,\sigma_\tau}\,\mathrm{e}^{-\frac{(\tau-\mu_\tau)^2}{2\sigma_\tau^2}}\,\mathrm{d}\tau \\ &= \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_0^2)}}\mathrm{e}^{-\frac{(x-(\mu_1+\mu_0))^2}{2(\sigma_1^2+\sigma_0^2)}} \end{aligned} \tag{A.4}$$

(because the quantity inside the integral is 1 as it is a probability density function).

Derivations for EM updates. For simplicity, we show the derivations for EM updates in the fixed kernel case. The approach is similar to the case when there is no smoothing. Let us 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.

Convolving each component of the mixture model using results from Lemma 1, we obtain

$$\begin{aligned} \log p^*(\mathcal{X},\mathcal{Z}|\Theta) &= \sum_{j=1}^{n}\sum_{i=1}^{k}\log\left[\frac{1}{\sqrt{2\pi(\sigma_i^2+\sigma_0^2)}}\mathrm{e}^{-\frac{(x-(\mu_i+\mu_0))^2}{2(\sigma_i^2+\sigma_0^2)}}\,p(z_i^{(j)}=1)\right]^{z_i^{(j)}} \\ &= \sum_{j=1}^{n}\sum_{i=1}^{k} z_i^{(j)}\left[-\log\left(\sqrt{2\pi(\sigma_i^2+\sigma_0^2)}\right) - \frac{(x-(\mu_i+\mu_0))^2}{2(\sigma_i^2+\sigma_0^2)} + \log\alpha_i\right]. \end{aligned} \tag{A.5}$$

Expectation Step: For this step, we need to compute the Q-function which is the expected value of Eq. (A.5) with respect to the hidden variables.

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

Computing the expected value of the hidden variables ($z_i^{(j)}$), we obtain,

$$\begin{aligned} w_i^{(j)} &= E_z[z_i^{(j)}] = \sum_{c=0}^{1} c * p(z_i^{(j)} = c|\Theta^{(t)}, x^{(j)}) \\ &= \frac{p(x^{(j)}|\Theta^{(t)}, z_i^{(j)}=1)\,p(z_i^{(j)}=1|\Theta^{(t)})}{p(x^{(j)}|\Theta^{(t)})} \\ &= \frac{\frac{1}{\sqrt{(\sigma_i^2+\sigma_0^2)}}\mathrm{e}^{-\frac{(x-(\mu_i+\mu_0))^2}{2(\sigma_i^2+\sigma_0^2)}}\,\alpha_i^{(t)}}{\sum_{m=1}^{k}\frac{1}{\sqrt{(\sigma_m^2+\sigma_0^2)}}\mathrm{e}^{-\frac{(x-(\mu_m+\mu_0))^2}{2(\sigma_m^2+\sigma_0^2)}}\,\alpha_m^{(t)}}. \end{aligned} \tag{A.7}$$

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

$$\frac{\partial}{\partial\Theta_i}Q(\Theta|\widehat{\Theta}(t)) = 0 \tag{A.8}$$

where Θ_i are the parameters of the ith component. Due to the assumption made that each data point comes from a single component, solving the above equation once more becomes trivial. The updates for the maximization step in the case of

GMMs are given as follows:

$$(\mu_i + \mu_0) = \frac{\sum_{j=1}^{n} w_i^{(j)} x^{(j)}}{\sum_{j=1}^{n} w_i^{(j)}}, \tag{A.9}$$

$$(\sigma_i^2 + \sigma_0^2) = \frac{\sum_{j=1}^{n} w_i^{(j)} (x^{(j)} - (\mu_i + \mu_0))^2}{\sum_{j=1}^{n} w_i^{(j)}} \quad \text{and} \tag{A.10}$$

$$\alpha_i = \frac{1}{n} \sum_{j=1}^{n} w_i^{(j)}. \quad \square \tag{A.11}$$

Proof of Lemma 2. Lets $c(\breve{x}, \mu_1)$ denote smoothing with respect to the density and $c(x, \breve{\mu}_1)$ denote smoothing with respect to the parameter. Now, the convolution of Gaussian density with respect to the mean is shown below:

$$c(x, \breve{\mu}_1) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-(\mu_1-\tau))^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(\tau-\mu_0)^2}{2\sigma_0^2}} \mathrm{d}\tau$$

since $\mu_0 = 0$, we have

$$c(x, \breve{\mu}_1) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x+\tau-\mu_1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{\tau^2}{2\sigma_0^2}} \mathrm{d}\tau. \tag{A.12}$$

Making the change of variable, $y = -\tau$, we have

$$\begin{aligned} c(x, \breve{\mu}_1) &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{((x-y)-\mu_1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{y^2}{2\sigma_0^2}} \mathrm{d}y \\ &= c(\breve{x}, \mu_1). \end{aligned} \tag{A.13}$$

Hence, convolution of a Gaussian density function with a Gaussian density with zero mean is equivalent to convolving the function with respect to its mean. □

Proof for Theorem 3. The expected gradient value of the smoothed complete log-likelihood surface is lower than the original complete log-likelihood surface, i.e.,

$$E\left[\left(\frac{\partial \log\ p(x|\theta)}{\partial \theta_i}\right)^2\right] \geq E\left[\left(\frac{\partial \log\ p^*(x|\theta)}{\partial \theta_i}\right)^2\right]$$
$$\forall i = 1, 2, \ldots, k.$$

Let us consider the expectation of the squared-gradient for the one component model. (i.e. its associated Fisher information (FI)):

$$l(\theta) = -n \log \sigma - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2.$$

Now,

$$FI = E\left[-\frac{\partial^2 \log f(x|\theta)}{\partial \theta^2}\right] = E\left[\left(\frac{\partial \log f(x|\theta)}{\partial \theta}\right)^2\right].$$

$$\frac{\partial \log L(\theta)}{\partial \mu} = \frac{1}{\sigma^2} \sum_{i=1}^{n} (x_i - \mu),$$

$$\frac{\partial \log L(\theta)}{\partial \sigma^2} = \frac{n}{2\sigma^4} \left(\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{n} - \sigma^2\right),$$

$$\frac{\partial^2 \log L(\theta)}{\partial\mu^2} = \frac{1}{\sigma^2}\sum_{i=1}^{n} -1 = -\frac{n}{\sigma^2},$$
$$\frac{\partial^2 \log L}{\partial\mu\partial\sigma^2} = -\frac{1}{\sigma^4}\sum_{i=1}^{n}(x_i-\mu) = \frac{\partial^2 \log L}{\partial\sigma^2\partial\mu},$$
$$\frac{\partial^2 \log L}{\partial\sigma^2\partial\sigma^2} = n\left[\frac{1}{2\sigma^4}\right] - \frac{1}{\sigma^6}\sum_{i=1}^{n}(x_i-\mu)^2.$$

Hence,

$$FI = \left[-\frac{\partial^2 \log f(x|\theta)}{\partial\theta^2}\right] = \begin{bmatrix} \frac{n}{\sigma^2} & 0 \\ 0 & \frac{n}{2\sigma^4} \end{bmatrix}.$$

Now, convolving with a Gaussian kernel with standard deviation σ_0 gives the FI for the smooth one-component model as follows:

$$FI^{smooth} = \begin{bmatrix} \frac{n}{\sigma^2+\sigma_0^2} & 0 \\ 0 & \frac{n}{2(\sigma^2+\sigma_0^2)^2} \end{bmatrix}.$$

It is now straightforward to see that

$$E\left[\left(\frac{\partial \log\ p^*(x,z|\theta)}{\partial\mu_i}\right)^2\right] \le E\left[\left(\frac{\partial \log\ p(x,z|\theta)}{\partial\mu_i}\right)^2\right]$$
$$\forall i = 1, 2, \ldots, k$$

where

$$\log\ p(x,z|\theta) = \sum_{i=1}^{k} z_i\ \log[\alpha_i \cdot p(x|\theta_i)].$$

Therefore,

$$\begin{aligned} E\left[\left(\frac{\partial \log\ p^*(x,z|\theta)}{\partial\mu_i}\right)^2\right] &= E\left[\left(\frac{\partial \sum_{i=1}^{k} z_i\ \log[\alpha_i \cdot p^*(x|\theta_i)]}{\partial\mu_i}\right)^2\right] \\ &= E\left[\left(\frac{z_i \cdot \partial \log[p^*(x|\theta_i)]}{\partial\mu_i}\right)^2\right] \\ &= \alpha_i \cdot I^*(\mu_i) \\ &\le \alpha_i \cdot I(\mu_i) \\ &= E\left[\left(\frac{\partial \log\ p(x,z|\theta)}{\partial\mu_i}\right)^2\right] \end{aligned}$$

where $I^*(\mu_i)$ and $I(\mu_i)$ are the FI for the ith component of the complete versions of the smoothed and original models respectively.

When we instead consider the expected square of the second derivatives (as an alternative measure of roughness), we obtain the following:

$$\left[-\frac{\partial^2 \log f(x|\theta)}{\partial\mu_i \cdot \partial\sigma_i^2}\right] = \begin{bmatrix} \frac{1}{\sigma^2} & \frac{1}{\sigma^4}(x-\mu) \\ \frac{1}{\sigma^4}(x-\mu) & \frac{1}{2\sigma^4} - \frac{(x-\mu)^2}{\sigma^6} \end{bmatrix}. \tag{A.14}$$

Squaring each term and taking expectation, we obtain

$$R = E\left[\left(\frac{\partial^2 \log f(x|\theta)}{\partial\theta^2}\right)^2\right] = \begin{bmatrix} \dfrac{1}{\sigma^4} & \dfrac{1}{\sigma^6} \\ \dfrac{1}{\sigma^6} & \dfrac{11}{4\sigma^8} \end{bmatrix}. \tag{A.15}$$

It is clear that $\frac{1}{\sigma^2}$, $\frac{1}{\sigma^4}$ and $\frac{1}{\sigma^8}$ are decreasing functions of σ^2 hence the roughness measure is lowered by the introduction of the smoothing parameter. □

References

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

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

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

Bhninga, D., Seidelb, W., Alfc, M., Patileae, B.G.V., Walther, G., 2007. Advances in mixture models. Computational Statistics and Data Analysis 51 (11), 5205–5210.

Biernacki, C., Celeux, G., Govaert, G., 2003. Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. Computational Statistics and Data Analysis 41 (3–4), 561–575.

Bilmes, J.A., 1998. 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).

Blake, A., Zisserman, A., 1987. Visual Reconstruction. MIT Press, Cambridge, MA.

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

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

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

Dasgupta, S., Schulman, L.J., 2000. A two-round variant of em for gaussian mixtures. In: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence. pp. 152–159.

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

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

Elidan, G., Ninio, M., Friedman, N., Schuurmans, D., 2002. Data perturbation for escaping local maxima in learning. In: Proceedings of the Eighteenth National Conference on Artificial Intelligence. pp. 132–139.

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

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

Hunt, L., Jorgensen, M., 2003. Mixture model clustering for mixed data with missing information. Computational Statistics and Data Analysis 41 (3–4), 429–440.

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

Li, J.Q., 1999. Estimation of mixture models. Ph.D. thesis. Department of Statistics, Yale University.

Martnez, A.M., Vitri, J., 2000. Learning mixture models using a genetic version of the EM algorithm. Pattern Recognition Letters 21 (8), 759–769.

McLachlan, G.J., Basford, K.E., 1988. Mixture Models: Inference and Applications to Clustering. Marcel Dekker, New York.

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

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

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

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

Reddy, C.K., Chiang, H.D., Rajaratnam, B., 2008. TRUST-TECH based expectation maximization for learning finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (7), 1146–1157.

Reddy, C.K., Rajaratnam, B., 2008. Component-wise parameter smoothing for learning mixture models. In: Proceedings of the 19th International Conference on Pattern Recognition, ICPR, Tampa, FL, USA, pp. 1–4.

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

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

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

Rose, K., 1998. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. Proceedings of the IEEE 80, 2210–2239.

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

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

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

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

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

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

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

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