Stability Region based Expectation Maximization for Model-based Clustering

Chandan K. Reddy*, Hsiao-Dong Chiang
School of Electrical and Computer Engineering,
Cornell University,
Ithaca, NY - 14853.

Bala Rajaratnam
Department of Statistical Science
Cornell University,
Ithaca, NY - 14853.

Abstract

In spite of the initialization problem, the Expectation-Maximization (EM) algorithm is widely used for estimating the parameters in several data mining related tasks. Most popular model-based clustering techniques might yield poor clusters if the parameters are not initialized properly. To reduce the sensitivity of initial points, a novel algorithm for learning mixture models from multivariate data is introduced in this paper. The proposed algorithm takes advantage of TRUST-TECH (TRansformation Under STability-reTaining Equilibra CHaracterization) to compute neighborhood local maxima on likelihood surface using stability regions. Basically, our method coalesces the advantages of the traditional EM with that of the dynamic and geometric characteristics of the stability regions of the corresponding nonlinear dynamical system of the log-likelihood function. Two phases namely, the EM phase and the stability region phase, are repeated alternatively in the parameter space to achieve improvements in the maximum likelihood. Though applied to Gaussian mixtures in this paper, our technique can be easily generalized to any other parametric finite mixture model. The algorithm has been tested on both synthetic and real datasets and the improvements in the performance compared to other approaches are demonstrated. The robustness with respect to initialization is also illustrated experimentally.

1 Introduction

Finite mixtures allow a probabilistic model-based approach to unsupervised learning [10] which plays an important role in predictive data mining applications. 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 [4, 6]. The usual steepest descent, conjugate gradient, or Newton-Raphson methods are too complicated for use in solving this problem [19]. EM has become a popular method since it takes advantages of problem specific properties. EM based approaches have been successfully used to solve problems that arise in various other applications [12, 2].

In this paper, we consider the problem of learning parameters of Gaussian Mixture Models (GMM). Fig 1 shows data generated by three Gaussian components with different mean and variance. Note that every data point has a probabilistic (or soft) membership that gives the probability with which it belongs to each of the components. 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 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. Given the number of components and an initial set of parameters, EM algorithm can be applied to compute the optimal estimates of the parameters that maximize the likelihood of the data given the estimates of these components. 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 parameters. To overcome this problem, a novel two phase algorithm based on stability region analysis is proposed. The main research concerns that motivated the new algorithm presented in this paper are :

- EM algorithm for mixture modeling converges to a local maximum of the likelihood function very quickly.
- There are many other promising local optimal solutions in the close vicinity of the solutions obtained from the methods that provide good initial guesses of the solution.
- Model selection criteria usually assumes that the global optimal solution of the log-likelihood function can be obtained. However, achieving this is computationally intractable.

*Corresponding author : email - ckr6@cornell.edu

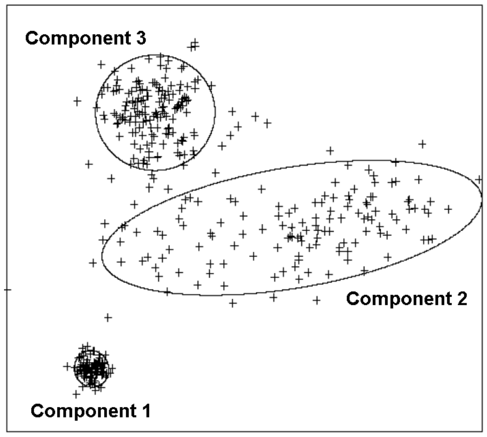


Figure 1. Data generated by three Gaussian components. The problem of learning mixture models is to obtain the parameters of these Gaussian components and the membership probabilities of each datapoint.

- Some regions in the search space do not contain any promising solutions. The promising and non-promising regions coexist and it becomes challenging to avoid wasting computational resources to search in non-promising regions.

Of all the concerns mentioned above, the fact that most of the local maxima are not distributed uniformly [16] makes it important for us to develop algorithms that not only help us to avoid searching in the low-likelihood regions but also emphasize the importance of exploring promising subspaces more thoroughly. This subspace search will also be useful for making the solution less sensitive to the initial set of parameters. In this paper, we propose a novel two phase algorithm for estimating the parameters of mixture models. Using concepts of dynamical systems and EM algorithm simultaneously to exploit the problem specific features of the mixture models, our algorithm obtains the optimal set of parameters by searching for the global maximum on the likelihood surface in a systematic manner.

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 problem of learning mixture models. Section 3 discusses some preliminaries about mixture models, EM algorithm and stability regions. Section 4 discusses our new framework and the details of our implementation are given in Section 5. Section 6 shows the experimental results of our algorithm on synthetic and real datasets. Finally, Section 7 concludes our discussion with future research directions.

2 Relevant Background

Although EM and its variants have been extensively used for learning mixture models, several researchers have approached the problem by identifying new techniques that give good initialization. More generic techniques like deterministic annealing [16], genetic algorithms [13] have been applied to obtain a good set of parameters. Though, these techniques have asymptotic guarantees, they are very time consuming and hence cannot be used for most of the practical applications. Some problem specific algorithms like split and merge EM [17], component-wise EM [6], greedy learning [18], incremental version for sparse representations[11], parameter space grid [8] are also proposed in the literature. Some of these algorithms are either computationally very expensive or infeasible when learning mixtures in high dimensional spaces [8]. Inspite of all the expense in 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 that leads to the local maximum with higher value of the likelihood are also successful to certain extent [15].

Though some of these methods apply other additional mechanisms (like perturbations [5]) to escape out of the local optimal solutions, systematic methods are yet to be developed for searching the subspace. The dynamical system of the log-likelihood function reveals more information on the neighborhood stability regions and their corresponding local maxima [3]. Hence, the difficulties of finding good solutions when the error surface is very rugged can be overcome by adding stability region based mechanisms to escape out of the convergence zone of the local maxima. Though this method might introduce some additional cost, one has to realize that existing approaches are much more expensive due to their stochastic nature. Specifically, for a problem in this context, where there is a non-uniform distribution of local maxima, it is difficult for most of the methods to search neighboring regions [20]. For this reason, it is more desirable to apply TRUST-TECH based Expectation Maximization (TT-EM) algorithm after obtaining some point in a promising region. The main advantages of the proposed algorithm are that it:

- Explores most of the neighborhood local optimal solutions unlike the traditional stochastic algorithms.
- Acts as a flexible interface between the EM algorithm and other global methods.
- Allows the user to work with existing clusters obtained from the traditional approaches and improves the qual-

ity of the solutions based on the maximum likelihood criteria.

- Helps the expensive global methods to truncate early.
- Exploits the fact that promising solutions are obtained by faster convergence of the EM algorithm.

3 Preliminaries

We now introduce some necessary preliminaries on mixture models, EM algorithm and stability regions. First, we describe the notation used in the rest of the paper:

Table 1. Description of the Notations used in the paper

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

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*. Θ represents the parameter set $(\alpha_1, \alpha_2, ...\alpha_k, \theta_1, \theta_2, ...\theta_k)$ and p is a univariate Gaussian density parameterized by θ_i(i.e. μ_i and σ_i):

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

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

$$0 \leq \alpha_i \leq 1 \text{ , } \forall i = 1, .., k, \; and \; \sum_{i=1}^{k} \alpha_i = 1 \tag{3}$$

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

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

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 :

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

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 [9].

3.2 Expectation Maximization

The EM algorithm assumes $\mathcal{X}$ to be *observed* data. The missing part, termed as *hidden* data, 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 [9]. 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

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

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

The EM algorithm produces a sequence of estimates $\{\widehat{\Theta}(t), t = 0, 1, 2, ...\}$ 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:

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

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

$$\begin{aligned} w_i^{(j)} &\equiv E\;[\;z_i^{(j)}|\mathcal{X},\widehat{\Theta}(t)\;] \\ &= Pr\;[\;z_i^{(j)}=1|x^{(j)},\widehat{\Theta}(t)\;] \\ &= \frac{\widehat{\alpha}_i(t)p(x^{(j)}|\widehat{\theta}_i(t))}{\sum_{i=1}^{k}\widehat{\alpha}_i(t)p(x^{(j)}|\widehat{\theta}_i(t))} \end{aligned} \quad (8)$$

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 :

$$\widehat{\Theta}(t+1) = arg\max_{\Theta}\{Q(\Theta,\widehat{\Theta}(t))\} \quad (9)$$

3.3 EM for GMMs

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 [19]. The $Q-function$ for GMM is given by:

$$\begin{aligned} Q(\Theta|\widehat{\Theta}(t)) = \sum_{j=1}^{n}\sum_{i=1}^{k} w_i^{(j)}[log\frac{1}{\sigma_i\sqrt{2\pi}} \\ -\frac{(x^{(j)}-\mu_i)^2}{2\sigma_i^2}+log\,\alpha_i] \end{aligned} \quad (10)$$

where

$$w_i^{(j)} = \frac{\frac{\alpha_i(t)}{\sigma_i(t)}e^{-\frac{1}{2\sigma_i(t)^2}(x^{(j)}-\mu_i(t))^2}}{\sum_{i=1}^{k}\frac{\alpha_i(t)}{\sigma_i(t)}e^{-\frac{1}{2\sigma_i(t)^2}(x^{(j)}-\mu_i(t))^2}} \quad (11)$$

The maximization step is given by the following equation :

$$\frac{\partial}{\partial\Theta_k}Q(\Theta|\widehat{\Theta}(t)) = 0 \quad (12)$$

where Θ_k is the parameters for the k^{th} component. Because of 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:

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

3.4 Stability Regions

This section mainly deals with the transformation of the original log-likelihood function into its corresponding nonlinear dynamical system and introduces some terminology pertinent to comprehend our algorithm. This transformation gives the correspondence between all the critical points of the s-dimensional likelihood surface and that of its dynamical system. For the case of spherical Gaussian mixtures with k components, we have the number of unknown parameters $s = 3k-1$. For convenience, the maximization problem is transformed into a minimization problem defined by the following objective function :

$$\begin{aligned} \min_{\Theta} f(\Theta) &= \min_{\Theta}\;\{\;-\;log\,p(\mathcal{Y}|\Theta)\;\} \\ &= \max_{\Theta}\;\{\;log\,p(\mathcal{Y}|\Theta)\;\} \end{aligned} \quad (13)$$

where $f(\Theta)$ is assumed to be in $C^2(\Re^s,\Re)$.

Definition 1 $\bar{\Theta}$ *is said to be a critical point of (13) if it satisfies the following condition*

$$\nabla f(\bar{\Theta}) = 0 \quad (14)$$

A critical point is said to be *nondegenerate* if at the critical point $\bar{\Theta}\in\Re^s$, $d^T\nabla^2 f(\bar{\Theta})d \neq 0$ $(\forall d \neq 0)$. We construct the following *gradient system* in order to locate critical points of the objective function (13):

$$\dot{\Theta}(t) = -\nabla f(\Theta) \quad (15)$$

where the state vector Θ belongs to the Euclidean space $\Re^s$, and the vector field $f:\Re^s\rightarrow\Re^s$ satisfies the sufficient condition for the existence and uniqueness of the solutions. The solution curve of Eq. (15) starting from Θ at time $t=0$ is called a *trajectory* and it is denoted by $\Phi(\Theta,\cdot):\Re\rightarrow\Re^s$. A state vector Θ is called an *equilibrium point* of Eq. (15) if $f(\Theta)=0$. An equilibrium point is said to be *hyperbolic* if the Jacobian of f at point $\bar{\Theta}$ has no eigenvalues with zero real part. The gradient system for the log-likelihood function in the case of spherical Gaussians is constructed as follows :

$$[\dot{\mu}_1(t) \;..\; \dot{\mu}_k(t) \; \dot{\sigma}_1(t) \;..\; \dot{\sigma}_k(t) \; \dot{\alpha}_1(t) \;..\; \dot{\alpha}_{k-1}(t)]^T$$

$$= \; - \; \left[\frac{\partial f}{\partial \mu_1} \,..\, \frac{\partial f}{\partial \mu_k} \; \frac{\partial f}{\partial \sigma_1} \,..\, \frac{\partial f}{\partial \sigma_k} \; \frac{\partial f}{\partial \alpha_1} \,..\, \frac{\partial f}{\partial \alpha_{k-1}} \right]^T$$

where

$$\begin{aligned} \frac{\partial f}{\partial \mu_i} &= \sum_{j=1}^{n} w_i^{(j)} \frac{(x^{(j)} - \mu_i)}{2\sigma_i^2} && \forall i = 1,..,k \\ \frac{\partial f}{\partial \sigma_i} &= \sum_{j=1}^{n} w_i^{(j)} \left[-\frac{1}{\sigma_i} + \frac{(x^{(j)} - \mu_i)^2}{\sigma_i^3} \right] && \forall i = 1,..,k \\ \frac{\partial f}{\partial \alpha_i} &= \frac{1}{\alpha_i} \sum_{j=1}^{n} w_i^{(j)} && \forall i = 1,..,k-1 \end{aligned} \tag{16}$$

For simplicity, we show the construction of the gradient system for the case of spherical Gaussians. It can be easily extended to the full covariance Gaussian mixture case. It should be noted that only (k-1) α values are considered in the gradient system because of the unity constraint. The dependent variable α_k is written as follows:

$$\alpha_k = 1 - \sum_{j=1}^{k-1} \alpha_j \tag{17}$$

Definition 2 *A hyperbolic equilibrium point is called a (asymptotically) stable equilibrium point (SEP) if all the eigenvalues of its corresponding Jacobian have negative real part. Conversely, it is an unstable equilibrium point if some eigenvalues have a positive real part.*

An equilibrium point is called a *type-k equilibrium point* if its corresponding Jacobian has exact k eigenvalues with positive real part. The *stable* ($W^s(\tilde{x})$) and *unstable* ($W^u(\tilde{x})$) manifolds of an equilibrium point, say $\tilde{x}$, is defined as:

$$W^s(\tilde{x}) = \{x \in \Re^s \; : \; \lim_{t \to \infty} \Phi(x,t) = \tilde{x}\} \tag{18}$$

$$W^u(\tilde{x}) = \{x \in \Re^s \; : \; \lim_{t \to -\infty} \Phi(x,t) = \tilde{x}\} \tag{19}$$

The task of finding multiple local maxima on the log-likelihood surface is transformed into the task of finding multiple stable equilibrium points on its corresponding gradient system. The advantage of our approach is that this transformation into the corresponding dynamical system will yield more knowledge about the various dynamic and geometric characteristics of the original surface and leads to the development a powerful method for finding improved solutions. In this paper, we are particularly interested in the properties of the local maxima and their one-to-one correspondence to the stable equilibrium points. To comprehend the transformation, we need to define *energy function*. A smooth function $V(\cdot) \; : \; \Re^s \; \rightarrow \; \Re^s$ satisfying $\dot{V}(\Phi(\Theta,t)) \; < \; 0$, $\forall \; x \notin$ {set of equilibrium points (E)} and t $\in \Re^+$ is termed as energy function.

Theorem 3.1 *[3]: $f(\Theta)$ is a energy function for the gradient system (15).*

Definition 3 *A type-1 equilibrium point x_d (k=1) on the practical stability boundary of a stable equilibrium point x_s is called a decomposition point.*

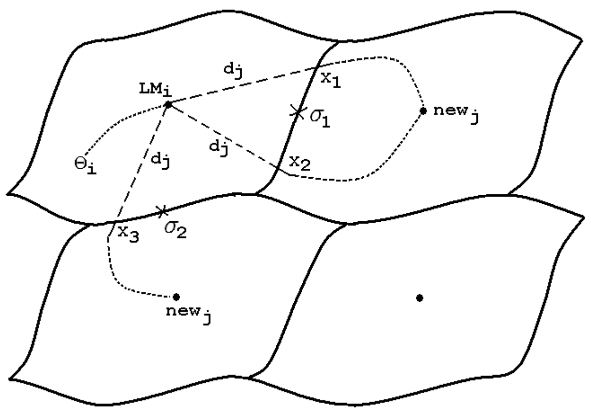
Definition 4 *The practical stability region of a stable equilibrium point x_s of a nonlinear dynamical system (15), denoted by $A_p(x_s)$ and is the interior of closure of the stability region $A(x_s)$ which is given by :*

$$A(x_s) = \{x \in \Re^s \; : \; \lim_{t \to \infty} \Phi(x,t) = x_s\} \tag{20}$$

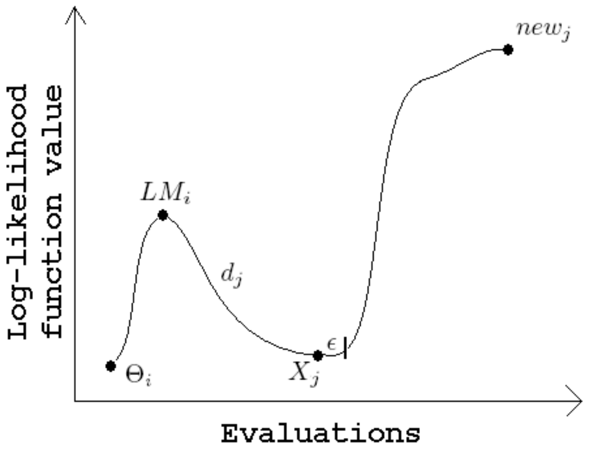
The boundary of practical stability region is called the *practical stability boundary* of x_s and will be denoted by $\partial A_p(x_s)$. Theorem 3.2 asserts that the practical stability boundary is contained in the union of the closure of the stable manifolds of all the decomposition points on the practical stability boundary. Hence, if the decomposition points can be identified, then an explicit characterization of the practical stability boundary can be established using (21). This theorem gives an explicit description of the geometrical and dynamical structure of the practical stability boundary.

Theorem 3.2 *(Characterization of practical stability boundary)[7]: Consider a negative gradient system described by (15). Let σ_i , i=1,2,... be the decomposition points on the practical stability boundary $\partial A_p(x_s)$ of a stable equilibrium point, say x_s. Then*

$$\partial A_p(x_s) = \bigcup_{\sigma_i \in \partial A_p} \overline{W^s(\sigma_i)}. \tag{21}$$



(a) Parameter Space



(b) Function Space

Figure 2. Various stages of our algorithm in (a) Parameter space - the solid lines indicate the practical stability boundary. Points highlighted on the stability boundary (σ_1, σ_2) are the decomposition points. The dotted lines indicate the convergence of the EM algorithm. d_j are the promising directions generated at the local maximum LM_i. The dashed lines indicate the stability region phase. x_1, x_2 and x_3 are the exit points on the practical stability boundary (b) Different variables in the function space and their corresponding log-likelihood values.

Our approach takes advantage of TRUST-TECH (TRansformation Under STability-reTaining Equilibra CHaracterization) to compute neighborhood local maxima on likelihood surface using stability regions. Originally, the basic idea of our algorithm was to find decomposition points on the practical stability boundary. Since, each decomposition point connects two local maxima uniquely, it is important to obtain the saddle points from the given local maximum and then move to the next local maximum through this decomposition point [14]. Though, this procedure gives a guarantee that the local maximum is not revisited, the computational expense for tracing the stability boundary and identifying the decomposition point is high compared to the cost of applying the EM algorithm directly using the exit point without considering the decomposition point. One can use the saddle point tracing procedure described in [14] for applications where the local methods like EM are much expensive.

4 Our Algorithm

Our framework consists mainly of two phases which are repeated in the promising subspaces of the parameter search space. It is more effective to use our algorithm at only these promising subspaces which are usually obtained by stochastic global methods. The first phase is the local phase (or the EM phase) where the promising solutions are refined to the corresponding locally optimal parameter set. The second phase which is the main contribution of this paper, is the stability region phase, where the exit points are computed and the neighborhood solutions are systematically explored through these exit points. Fig. 2 shows the different steps of our algorithm both in (a) the parameter space and (b) the function space.

This approach can be treated as a hybrid between global methods for initialization and the EM algorithm which gives the local maxima. One of the main advantages of our approach is that it searches the parameter space more deterministically. This approach differs from traditional local methods by computing multiple local solutions in the neighborhood region. This also enhances user flexibility by allowing the users to choose between different sets of good clusterings. Though global methods give promising subspaces, it is important to explore this subspace more thoroughly especially in problems like parameter estimation. Algorithm 1 describes our approach.

In order to escape out of this local maximum, our methods needs to compute certain promising directions based on the local behaviour of the function. One can realize that generating these promising directions is one of the important aspects of our algorithm. Surprisingly, choosing random directions to move out of the local maximum works

Algorithm 1 Stability Region based EM Algorithm

```
Input: Parameters $\Theta$, Data $\mathcal{X}$, tolerance $\tau$, Step $S_p$
Output: $\widehat{\Theta}_{MLE}$
Algorithm:
Apply global method and store the q promising solutions
$\Theta_{init} = \{\Theta_1, \Theta_2, .., \Theta_q\}$    Initialize E= $\phi$
while $\Theta_{init} \neq \phi$ do
    Choose $\Theta_i \in \Theta_{init}$, set $\Theta_{init} = \Theta_{init} \backslash \{\Theta_i\}$
    $LM_i = EM(\Theta_i, \mathcal{X}, \tau)$    $E = E \cup \{LM_i\}$
    Generate promising direction vectors $d_j$ from $LM_i$
    for each $d_j$ do
        Compute Exit Point $(X_j)$ along $d_j$ starting from $LM_i$ by evaluating the log-likelihood function given by (4)
        $New_j = EM(X_j + \epsilon \cdot d_j, \mathcal{X}, \tau)$
        if $new_j \notin E$ then
            $E = E \cup New_j$
        end if
    end for
end while
$\widehat{\Theta}_{MLE} = max\{val(E_i)\}$
```

well for this problem. One might also use other directions like eigenvectors of the Hessian or incorporate some domain-specific knowledge (like information about priors, approximate location of cluster means, user preferences on the final clusters) depending on the application that they are working on and the level of computational expense that they can afford. We used random directions in our work because they are very cheap to compute. Once the promising directions are generated, exit points are computed along these directions. *Exit points* are points of intersection between any given direction and the practical stability boundary of that local maximum along that particular direction. If the stability boundary is not encountered along a given direction, it is very likely that one might not find any new local maximum in that direction. With a new initial guess in the vicinity of the exit points, EM algorithm is applied again to obtain a new local maximum.

5 Implementation Details

Our program is implemented in MATLAB and runs on Pentium IV 2.8 GHz machine. The main procedure implemented is TT_EM described in Algorithm 2. The algorithm takes the mixture data and the initial set of parameters as input along with step size for moving out and tolerance for convergence in the EM algorithm. It returns the set of parameters that correspond to the Tier-1 neighboring local optimal solutions. The procedure $eval$ returns the log-likelihood score given by (4). The Gen_Dir procedure generates promising directions from the local maxima. Exit points are obtained along these generated directions. The procedure $update$ moves the current parameter to the next parameter set along a given k^{th} direction $Dir[k]$. Some of the directions might have one of the following two problems: (i) Exit points might not be obtained in these directions. (ii) Even if the exit point is obtained it might converge to a less promising solution. If the exit points are not found along these directions, search will be terminated after $Eval\_MAX$ number of evaluations. For all exit points that are successfully found, EM procedure is applied and all the corresponding neighborhood set of parameters are stored in the $Params[\,]$ [1]. Since, different parameters will be of different range, care must be taken while multiplying with the step sizes. It is important to use the current estimates to get an approximation of the step size with which one should move out along each parameter in the search space. Finally, the solution with the highest likelihood score amongst the original set of parameters and the Tier-1 solutions is returned.

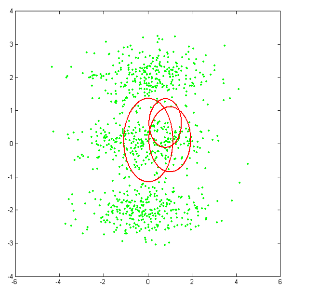
Algorithm 2 Params[ ] $TT\_EM(Pset, Data, Tol, Step)$

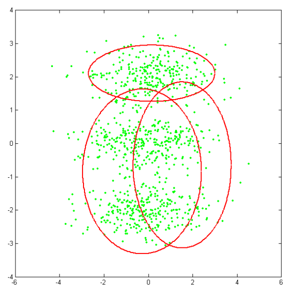
```
$Val = eval(Pset)$
$Dir[\,] = Gen\_Dir(Pset)$
$Eval\_MAX = 500$
for $k = 1$ to $size(Dir)$ do
    $Params[k] = Pset$    $ExtPt = OFF$
    $Prev\_Val = Val$    $Cnt = 0$
    while (! $ExtPt$) && $(Cnt < Eval\_MAX)$ do
        $Params[k] = update(Params[k], Dir[k], Step)$
        $Cnt = Cnt + 1$
        $Next\_Val = eval(Params[k])$
        if $(Next\_Val > Prev\_Val)$ then
            $ExtPt = ON$
        end if
        $Prev\_Val = Next\_Val$
    end while
    if $count < Eval\_MAX$ then
        $Params[k] = update(Params[k], Dir[k], ASC)$
        $Params[k] = EM(Params[k], Data, Tol)$
    else
        $Params[k] = NULL$
    end if
end for
Return $max(eval(Params[\,]))$
```

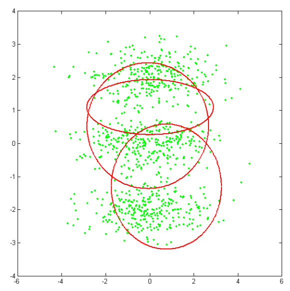
6 Results and Discussion

Our algorithm has been tested on both synthetic and real datasets. The initial values for the centers and the covari-

[1]To ensure that the new initial points are in the different stability regions, one should move along the directions 'ϵ' away from the exit points.


(a)


(b)


(c)

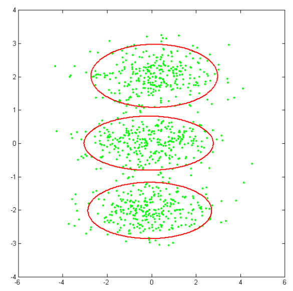

(d)

Figure 3. Parameter estimates at various stages of our algorithm on the three component Gaussian mixture model (a) Poor random initial guess (b) Local maximum obtained after applying EM algorithm with the poor initial guess (c) Exit point obtained by our algorithm (d) The final solution obtained by applying the EM algorithm to the initial point in the neighboring stability region.

ances were chosen uniformly random. Uniform priors were chosen for initializing the components. For real datasets, the centers were chosen randomly from the sample points.

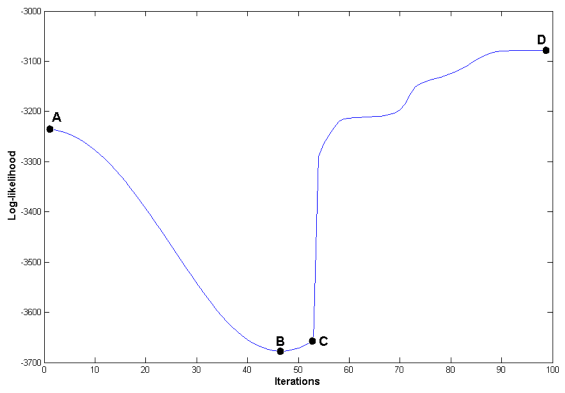


Figure 4. Graph showing likelihood vs Evaluations. A corresponds to the original local maximum (L=-3235.0). B corresponds to the exit point (L=-3676.1). C corresponds to the new initial point in the neighboring stability region (L=-3657.3) after moving out by 'ϵ'. D corresponds to the new local maximum (L=-3078.7).

6.1 Synthetic Datasets

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 containing $n = 900$ data points. 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 [16]. All the three mixtures have uniform priors. Fig. 3 shows various stages of our algorithm and demonstrates how the clusters obtained from existing algorithms are improved using our algorithm. The initial clusters obtained are of low quality because of the poor initial set of parameters. Our algorithm takes these clusters and applies the stability region step and the EM step simultaneously to obtain the final result. Fig. 4 shows the value of the log-likelihood during the stability region phase and the EM iterations.

In the third synthetic dataset, a more complicated overlapping Gaussian mixtures are considered [6]. 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 $\alpha_4 = 0.1$.

$$\Sigma_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 6 & -2 \\ -2 & 6 \end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \qquad \Sigma_4 = \begin{bmatrix} 0.125 & 0 \\ 0 & 0.125 \end{bmatrix}$$

6.2 Real Datasets

Two real datasets obtained from the UCI Machine Learning repository [1] were also used for testing the performance of our algorithm. Most widely used Iris data with 150 samples, 3 classes and 4 features was used. Wine data set with

Table 2. Performance of our algorithm on an average of 100 runs on various synthetic and real datasets

Dataset	Samples	Clusters	Features	EM (mean $\pm$ std)	TRUST-TECH-EM (mean $\pm$ std)
Spherical	40	5	2	38.07$\pm$2.12	43.55$\pm$0.6
Elliptical	900	3	2	-3235$\pm$0.34	-3078.7$\pm$0.03
Full covariance 1	500	4	2	-2345.5 $\pm$175.13	-2121.9$\pm$ 21.16
Full covariance 2	2000	4	2	-9309.9 $\pm$694.74	-8609.7 $\pm$37.02
Iris	150	3	4	-198.13$\pm$27.25	-173.63$\pm$11.72
Wine	178	3	13	-1652.7$\pm$1342.1	-1618.3$\pm$1349.9

178 samples was also used for testing. Wine data had 3 classes and 13 features. For these real data sets, the class labels were deleted thus treating it as unsupervised learning problem. Table 2 summarizes our results over 100 runs. The mean and the standard deviations of the log-likelihood values are reported. The traditional EM algorithm with random starts is compared against our algorithm on both synthetic and real data sets. Our algorithm not only obtains higher likelihood value but also produces it with high confidence. The low standard deviation of our results indicates the robustness of obtaining the global maximum. In the case of the wine data, the improvements with our algorithm are not much significant compared to the other datasets. This might be due to the fact that the dataset might not have Gaussian components. Our method assumes that the underlying distribution of the data is mixture of Gaussians. Table 3 gives the results of TRUST-TECH compared with other methods like split and merge EM and k-means+EM proposed in the literature.

Table 3. Comparison of TRUST-TECH-EM with other methods

Method	Elliptical	Iris
RS+EM	-3235 $\pm$ 14.2	-198 $\pm$ 27
K-Means+EM	-3195 $\pm$ 54	-186 $\pm$ 10
SMEM	-3123 $\pm$ 54	-178.5 $\pm$ 6
TRUST-TECH-EM	-3079 $\pm$ 0.03	-173.6 $\pm$ 11

6.3 Discussion

It will be effective to use our algorithm for those solutions that appear to be promising. Due to the nature of the problem, it is very likely that the nearby solutions surrounding the existing solution will be more promising. One of the primary advantages of our method is that it can be used along with other popular methods available and improve the quality of the existing solutions. In clustering problems, it is an added advantage to perform refinement of the final clusters obtained. Most of the focus in the literature was on new methods for initialization or new clustering techniques which often do not take advantage of the existing results and completely start the clustering procedure "*from scratch*". Though shown only for the case of multivariate Gaussian mixtures, our technique can be effectively applied to any parametric finite mixture model.

Table 4 summarizes the average number of iterations taken by the EM algorithm for the convergence to the local optimal solution. We can see that the most promising solution produced by our TRUST-TECH methodology converges much faster. In other words, our method can effectively take advantage of the fact that the convergence of the EM algorithm is much faster for high quality solutions. This is an inherent property of the EM algorithm when applied to the mixture modeling problem. We exploit this property of the EM for improving the efficiency of our algorithm. Hence, for obtaining the Tier-1 solutions using our algorithm, the threshold for the number of iterations can be significantly lowered.

Table 4. Number of iterations taken for the convergence of the best solution.

Dataset	Avg. no. of iterations	No. of iterations for the best solution
Spherical	126	73
Elliptical	174	86
Full covariance	292	173

7 Conclusion and Future Work

A novel stability region based EM algorithm has been introduced for estimating the parameters of mixture models. The EM phase and the stability region phase are applied alternatively in the context of the well-studied mixture model parameter estimation problem. The concept of stability region helps us to understand the topology of the original log-likelihood surface. Our method computes the neighborhood local maxima on likelihood surface using stability regions of the corresponding nonlinear dynamical system. The algorithm has been tested successfully on various synthetic and real datasets and the improvements in the performance are clearly manifested. Some properties of the EM algorithm about the rate of convergence have been exploited efficiently.

Our algorithm can be easily extended to popularly used k-means clustering technique. In the future, we plan to work on applying these stability region based methods for other widely used EM related parameter estimation problems like training Hidden Markov Models, Mixture of Factor Analyzers, Probabilistic Principal Component Analysis, Bayesian Networks etc. We would also plan to extend our technique to Markov Chain Monte Carlo strategies like Gibbs sampling for the estimation of mixture models.

References

[1] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. *http://www.ics.uci.edu/mlearn/MLRepository.html, University of California, Irvine, Dept. of Information and Computer Sciences*, 1998.

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

[3] H.D. Chiang and C.C. Chu. A systematic search method for obtaining multiple local optimal solutions of nonlinear programming problems. *IEEE Transactions on Circuits and Systems: I Fundamental Theory and Applications*, 43(2):99–109, 1996.

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

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

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

[7] J. Lee and H.D. Chiang. A dynamical trajectory-based methodology for systematically computing multiple optimal solutions of general nonlinear programming problems. *IEEE Transactions on Automatic Control*, 49(6):888 – 899, 2004.

[8] J. Q. Li. *Estimation of Mixture Models*. PhD thesis, Department of Statistics,Yale University, 1999.

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

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

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

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

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

[14] C. K. Reddy and H.D. Chiang. A stability boundary based method for finding saddle points on potential energy surfaces. *Journal of Computational Biology*, 13(3):745–766, 2006.

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

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

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

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

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

[20] B. Zhang, C. Zhang, and X. Yi. Competitive EM algorithm for finite mixture models. *Pattern Recognition*, 37(1):131–144, 2004.