Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models

Christophe Biernacki

UMR CNRS 6623 - Université de Franche-Comté

Gilles Celeux[1]

INRIA Rhône-Alpes

Gérard Govaert

UMR CNRS 6599 - UTC Compiègne

Abstract

Simple methods to choose sensible starting values for the EM algorithm to get maximum likelihood parameter estimation in mixture models are compared. They are based on random initialization, using a Classification EM algorithm (CEM), a Stochastic EM algorithm (SEM) or previous short runs of EM itself. Those initializations are included in a Search/Run/Select strategy which can be compounded by repeating the three steps. They are compared in the context of multivariate Gaussian mixtures on the basis of numerical experiments on both simulated and real data sets in a target number of iterations. The main conclusions of those numerical experiments are the following. The simple random initialization which is probably the most employed way of initiating EM is often outperformed by strategies using CEM, SEM or shorts runs of EM before running EM. Also, it appears that compounding is generally profitable since using a single run of EM can often lead to suboptimal solutions. Otherwise, none of the experimental strategies can be regarded as the best one and it is difficult to characterize situations where a particular strategy can be expected to outperform the other ones. However, the strategy initiating EM with short runs of EM can be recommended. This strategy, which as far as we know was not used before the present study, has some advantages. It is simple, performs well in a lot of situations presupposing no particular form of the mixture to be fitted to the data and seems little sensitive to noisy data.

Keywords: Classification EM, EM algorithm, Initialization Strategies, Multivariate Gaussian Mixture, Optimization, Stochastic EM.

1 Introduction

In most applications, parameters of a mixture model are estimated by maximizing the likelihood and the standard tool to maximum likelihood (ML) estimation for mixture models is the EM algorithm (Dempster, Laird and Rubin 1977). But EM solution can highly depend on its starting position especially in a multivariate context. This jeopardizes statistical analysis of mixture for two reasons. First ML estimation is expected to provide sensible estimates of the mixture parameters. Secondly, the highest maximized likelihood enters the definition of numerous criteria aiming to select a good mixture model and especially to

[1]Corresponding author: G. Celeux, Inria Rhône-Alpes, 655 avenue de l'Europe Montbonnot St Martin, F38330 St Ismier Cedex, email: Gilles.Celeux@inria.fr, tel: 33476615325, fax: 33476615252.

choose a relevant number of mixture components. Thus, it is important to get the highest criterion value when estimating the parameters of a mixture through maximum likelihood. Let us illustrate this fact with a simple example. We consider a sample of size $n = 50$ from a two-component univariate Gaussian mixture with proportions $p_1 = p_2 = 0.5$, means $\mu_1 = -0.8$, $\mu_2 = 0.8$ and variances $\sigma_1^2 = 1$, $\sigma_2^2 = 1.5$. All the parameters are supposed to be known, except the means μ_1 and μ_2. The likelihood has two local maxima as shown in Figure 1.

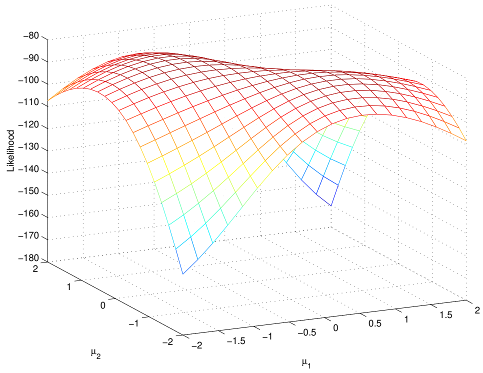


Figure 1: A two-mode likelihood surface

If the lowest likelihood maximum is selected, it can have consequence for choosing the number of components K. For instance, Table 1 gives the AIC criterion values (Akaike 1974) for $K = 1$ and for the two different ML solutions for $K = 2$. Thus, despite its marked tendency to favor too complex models, AIC concludes wrongly for a single Gaussian distribution when the lowest local maximum likelihood is selected.

	1 comp.	2 comp. highest ML	2 comp. lowest ML
AIC	-85.29	-84.88	-85.95

Table 1: AIC criterion values for different ML estimates

In this paper, we present and compare simple strategies, on the basis of numerical experiments, to deal with the problem of getting the highest likelihood value in the framework of multivariate Gaussian mixtures. We pay attention to this particular mixture model for simplicity and because it is by far the most employed mixture model with many applications in cluster analysis and statistical pattern recognition (see McLachlan and Peel 2000). Moreover, EM can be thought of as to be much too sensitive to initial position in a multivariate context. However, it must be noticed that getting parameter estimates maximizing the likelihood for a general Gaussian mixture model is an ill-posed problem since the likelihood is unbounded and the EM algorithm can lead to spurious solutions. (Typically, one of the estimated mixture component has a variance matrix near the null matrix.) Avoiding such spurious solutions must be done on a subjective ground and can be controversial (see McLachlan and Peel 2000 chapter 3 section 10). To avoid an unbounded likelihood, we restricted attention to variance matrices with equal determinants.

Our goal is identifying a simple method that would give the highest likelihood in a fixed number of iterations. The key issue is finding the trade-off between adequately searching the parameter space with initial positions versus iterating the EM algorithm sufficiently to get close to the maximum. Given a total iteration constraint, we basically consider a three step Search/Run/Select (S/R/S) strategy for maximizing the likelihood:

1. Build a search method for generating p initial positions. This could be based on random starts or the output from an algorithm like a Classification EM (CEM) algorithm, a Stochastic EM (SEM) algorithm or short runs of the standard EM algorithm. The parameter p is depending on an allotment of iterations.

2. Run the EM algorithm a set number of times at each initial position with a fixed number of iterations.

3. Select the solution providing best likelihood among the p trials, say θ^*.

This three-step strategy can be compounded by repeating the three steps x times and using the $\theta_1^*, \ldots, \theta_x^*$ as the starting positions in step 1. By compounding, one increases starting position variation, but one must decrease the length of the EM runs possible within the steps in order to fix the total number of steps.

We used the CEM and the SEM algorithm in the Search step for the following reasons. The CEM algorithm tends to produce a mixture with well separated components which can be regarded as a good initial solution for EM. The SEM algorithm by using random drawing at each iteration, prevents one from converging to the first stationary point of the loglikelihood it encounters. For this reason it can avoid sub-optimal maxima of the loglikelihood function.

The paper is organized as follows. The setting of our study and the material we need to define the competing strategies are presented in Section 2. In particular, we present the Classification and the Stochastic EM for mixtures. Our strategies are presented and briefly discussed in Section 3. Numerical experiments are presented in Section 4: In Section 4.1 the game rules of the experiments are detailed and extensive numerical experiments on both simulated and real data are presented in Section 4.2. In Section 5 the conclusions of our study are drawn and additional comments are given.

2 Gaussian mixture maximum likelihood estimation

Gaussian mixture model is a powerful model for clustering, pattern recognition and multivariate density estimation (McLachlan and Peel 2000). In this model, data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\mathbf{R}^d$ are assumed to arise from a random vector with density

$$f(\mathbf{x}) = \sum_{k=1}^{K} p_k \phi(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{1}$$

where the p_k's are the mixing proportions ($0 < p_k < 1$ for all $k = 1, \ldots, K$ and $\sum_k p_k = 1$) and $\phi(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the density of a Gaussian distribution with mean vector $\boldsymbol{\mu}$ and variance matrix $\boldsymbol{\Sigma}$. Generally, the mixture parameters $\theta = (p_1, \ldots, p_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$ are estimated by maximizing the loglikelihood

$$L(\theta|\mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{i=1}^{n} \ln \left[\sum_{k=1}^{K} p_k \phi(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right]. \tag{2}$$

The EM algorithm. The standard tool for finding maximum likelihood solution is the EM algorithm. In the mixture context (see McLachlan and Peel 2000 for details), it can be summarized as follows. Starting from an initial parameter θ^0, an iteration of the EM algorithm consists of the E step in which the current conditional probabilities $\hat{p}_k(\mathbf{x}_i)(1 \leq i \leq n, 1 \leq k \leq K)$ that $\mathbf{x}_i$ arises from the kth mixture component are computed, and the M step in which the ML estimates $\hat{p}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k$ are updated using the conditional probabilities $\hat{p}_k(\mathbf{x}_i)$ as conditional mixing weights.

The characteristics of the EM algorithm are well documented (see for instance McLachlan and Krishnam 1997). It leads in general to simple equations, has the nice property of increasing the loglikelihood at each iteration until stationarity, and in many circumstances, it derives sensible parameter estimates and consequently it is a popular tool to derive maximum likelihood estimation. However, EM is known to converge slowly in some situations. We do not address this important aspect of EM here. It has received much attention recently and many algorithms aiming to speed up the convergence of EM while preserving its simplicity have been proposed (see the chapter 4 of McLachlan and Krishnam 1997, and, for algorithms specific to the mixture context Liu and Sun 1997 and Celeux, Chrétien, Forbes and Mkhadri 2001, Pilla and Lindsay 2001; see also Böhning 1999 and McLachlan and Peel 2000 which review many faster algorithms specific to mixtures). The second important drawback of EM is that its solution can highly depend on its starting position and consequently produce sub-optimal maximum likelihood estimates. The strategies that we propose are aiming to overcome this limitation. Note that both drawbacks, slow convergence and dependence from initial positions, can be regarded as linked in practical situations. Actually, it is possible that starting from some position leads to a slower convergence rate for EM and that EM is stopped before reaching a sensible value of the likelihood. To act against this high dependency of EM on its initial position, we make use of related algorithms, CEM and SEM, that we present now.

The CEM algorithm. This algorithm (see Celeux and Govaert 1992) incorporates a classification step between the E and M steps of EM. Starting from an initial parameter θ^0, an iteration of CEM consists of three steps.

- **E step:** The conditional probabilities $\hat{p}_k(\mathbf{x}_i)(1 \leq i \leq n, 1 \leq k \leq K)$ for the current value of θ as done in the standard EM.
- **C step:** A partition $P = (P_1, \ldots, P_K)$ of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is designed by assigning each point $\mathbf{x}_i$ to the component maximizing the conditional probability $\hat{p}_k(\mathbf{x}_i)$.
- **M step:** The ML estimates $(\hat{p}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)$ are computed using the cluster P_k as sub-sample $(1 \leq k \leq K)$ of the kth mixture component.

The main features of CEM are the following. CEM is a *K-means*-like algorithm (see MacQueen 1967) and contrary to EM, it converges in a finite number of iterations. CEM is not maximizing the observed loglikelihood (2) but is maximizing in θ and $z_1, \ldots, z_n$ the complete data loglikelihood CL where the missing component label z_i of each sample point is included in the data set:

$$CL(\theta|z_1, \ldots, z_n, \mathbf{x}_1, \ldots, \mathbf{x}_n) = \sum_{k=1}^{K} \sum_{\{i/z_i=k\}} \ln\, [p_k \phi(\mathbf{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)], \tag{3}$$

where $\{i/z_i = k\}$ is the set of units arising from the kth mixture component. As a consequence, CEM is not expected to converge to the ml estimate of θ and yields inconsistent estimates of the parameters especially when the mixture components are overlapping or are in disparate proportions (see McLachlan and Peel 2000, Section 2.21).

The SEM algorithm. This algorithm is a stochastic algorithm incorporating between the E and M steps a restoration of the unknown component labels z_i, $i = 1, \ldots, n$, by drawing them at random from their current conditional distribution. SEM algorithm is same as the Monte Carlo EM algorithm with a single replication (see McLachlan and Krishnam 1997, chapter 6). Starting from an initial parameter θ^0, an iteration of SEM consists of three steps.

- **E step:** The conditional probabilities $\hat{p}_k(\mathbf{x}_i)(1 \leq i \leq n, 1 \leq k \leq K)$ for the current value of θ as done in the standard EM.

- **S step:** A partition $P = (P_1, \ldots, P_K)$ of $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is designed by assigning each point $\mathbf{x}_i$ at random to one of the mixture components according to the multinomial distribution with parameter $(\hat{p}_k(\mathbf{x}_i), 1 \leq k \leq K)$.

- **M step:** The ML estimates $(\hat{p}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)$ are computed using the cluster P_k as sub-sample $(1 \leq k \leq K)$ of the kth mixture component.

The main features of SEM are the following. SEM is a Data Augmentation algorithm (see Wei and Tanner 1990). SEM does not converge pointwise. It generates a Markov chain whose stationary distribution is more or less concentrated around the ML parameter estimator. A natural parameter estimate from a SEM sequence $(\theta^r)_{r=1,\ldots,R}$ is the mean $\sum_{r=b+1}^{R} \theta^r/(R-b)$ of the iterates values where the first b burn-in iterates have been discarded in forming this mean. An alternative estimate is to consider the parameter value leading to the highest likelihood in a SEM sequence.

3 Experimental strategies

In this section we present the strategies that we consider in our numerical experiments. These strategies consist essentially of initializing the EM algorithm from random positions (standard strategy), from solutions given by the CEM algorithm or by the SEM algorithm and finally, from solutions obtained after short runs of EM algorithm. To perform these numerical experiments, we used our software MIXMOD coded in C++ with an interface in Scilab, devoted to the identification of Gaussian mixtures[2]. We do not consider sophisticated data analysis tools to deal with the initiating problem of EM (see for instance Ueda and Nakano 1998). In our opinion, such strategies can give good results on specific examples, but are painfully slow and may be not beneficial in a general context (see McLachlan and Peel 2000 chapter 2, section 4).

Random inialization The standard way to initiate the EM algorithm consists of initializing it from a random position. Usually this random initial position is obtained by drawing at random component means in the data set. Since this is probably the most employed way of initiating EM, it can be regarded as a reference strategy. It is denoted "1EM" in the following. An extension of this simple strategy consists of repeating it x times from different random positions and selecting the solution maximizing the likelihood among those x runs. We denote "xEM" this strategy which is the basic S/R/S algorithm.

[2] MIXMOD is a free software available at `http://www.inrialpes.fr/is2/pub/software/MIXMOD`

Using the CEM algorithm In many circumstances and especially in a cluster analysis setting, the mixture component means are expected to be different. Thus, a reasonable and largely employed way of initiating EM consists of starting from the solution of a K-means type algorithm. Acting in such a way, it is hoped that the initial position will be sensible. This point of view leads to the following strategies. Runs of CEM from random positions followed by EM from the position providing the highest *complete* data loglikelihood obtained with CEM. For each CEM run, we consider the same assumptions on the mixture components that we consider when running EM. We denote "1CEM-EM" this strategy. And, x repetitions of the previous strategy give rise to an additional strategy denoted "xCEM-EM". In xCEM-EM, the basic three steps are random search, CEM runs, and selection, compounded by x outer loops with EM runs.

Using short runs of EM One advantage of initiating EM with CEM lies in the fact that CEM converges generally in a small number of iterations. Thus, without consuming a large amount of CPU times, several runs of CEM can be performed before passing to EM with the best solution among those CEM runs. But, it is possible to recover this feature with short runs of EM. By a short run of EM, we mean that we do not wait for convergence and that we stop the algorithm as soon as

$$\frac{L^q - L^{q-1}}{L^q - L^0} \leq 10^{-2}, \tag{4}$$

L^q denoting the observed loglikelihood at qth iteration. Here 10^{-2} represents a threshold value which has to be chosen on a pragmatic ground. From our experiments, we chose this value to get a number of iterations for such an EM short run approximatively equal to the number of iterations obtained with CEM. It leads to the following strategies. Several short runs of EM from random positions followed by a long run of EM from the solution maximizing the *observed* loglikelihood. We denote "1em-EM" this strategy. And, x repetitions of the previous strategy lead to the so called "$x$em-EM" strategy. In $x$em-EM, the basic three steps are random search, short runs of EM, and selection, compounded by x outer loops with EM runs.

Using Stochastic EM The stochastic EM algorithm generates an ergodic Markov chain. Thus a sequence of parameter estimates via SEM is expected to visit the whole parameter space with long sojourns in the neighborhood of sensible maxima of likelihood functions. This characteristic of SEM invites to use the following strategies.

- A run of SEM, followed by a run of EM from the solution obtained by computing the mean values of the sequence of parameter estimates provided by SEM after a burn-in period. We denote "SEMmean-EM" this strategy. The idea underlying this strategy is that SEM is expected to spend most of the time near sensible likelihood maxima with a large attractive neighborhood.

- The *same* run of SEM followed by a run of EM from the position leading to the highest maximum likelihood value reached by the SEM sequence of parameter iterates. Here, the idea is that an SEM sequence is expected to enter rapidly in the neighborhood of the global maximum of the likelihood function. We denote "SEMmax-EM" this strategy.

The SEM method use a slight variation on the basic S/R/S theme since *search* and *run* are mixed in a single step. Otherwise "SEMmean-EM" and "SEMmax-EM" differ by their respective selection step.

4 Numerical experiments

As pointed out in Meila and Heckerman (2001), we should not expect to find an initialization strategy that outperforms all the others on all data sets. We simply hope to find honest initialization strategies working well for large classes of situations arising in practice. The only way to answer this question is to perform extensive numerical experiments for various data sets. We will conduct simulation studies and carry out real data analysis and present the results in Section 4.2. In addition, we describe the criteria used for these numerical experiments in Section 4.1.

First, recall that we restricted attention to variance matrices with equal determinants in order to avoid that the likelihood is unbounded.

4.1 The game rules

Available CPU time. We are essentially interested in finding the highest likelihood in a target CPU time. We are not interested to find strategies leading the faster to a local maximum likelihood. Thus to compare the strategies in competition we proceed as follows. We suppose the user accepts to wait a time t to get the solution. In terms of computation time, it appears that there is no sensitive difference between EM, CEM and SEM to perform one iteration. For instance, for the last example presented in 4.2 concerning a population of size $n = 3641$ described by $d = 5$ for which a $K = 10$ component Gaussian mixture has been considered, it took 0.181s for EM, 0.188s for CEM and 0.196s for SEM. More generally, the CPU time variation between the three algorithms never exceeds 10%. Thus, we assume for simplicity that t, the price to be paid for any strategy, can be converted in the total number of iterations. It means that each strategy will have the same total number of available iterations ITEMAX. Moreover, the partition of iterations is equal for each run and each algorithm inside a run. For instance, suppose that ITEMAX = 1000 iterations are available. For the eight strategies, they are shared in the following way, taking $x = 10$ for the strategies including repeated algorithm runs.

- 1EM: 1000 iterations for EM.
- 10EM: 100 iterations for each EM run.
- 1CEM-EM: 500 iterations for CEM and 500 iterations for EM.
- 10CEM-EM: 10 repetitions of 50 iterations for CEM and 50 iterations for EM.
- 1em-EM: 500 iterations for em and 500 iterations for EM.
- 10em-EM: 10 repetitions of 50 iterations for em and 50 iterations for EM run.
- SEMmean-EM and SEMmax-EM: 500 iterations for SEM and 500 iterations for EM.

In the strategies combining two algorithms we gave half iterations for both. From additional experiments not reported here, it appears that it is a good choice and there is little interest to choose other proportion for sharing the total number of iterations.

Stopping rules. The EM algorithm is stopped with the number of iterations. We do not use stopping criteria based on the relative change of the estimates or loglikelihood because the slow convergence of the EM makes such criteria hazardous. See Lindsay (1995) and McLachlan and Peel (2000) for more on stopping criteria.

Initial conditions. Initial proportions are equal, random initial component means are drawn in the data set, and initial variance matrices are diagonal with diagonal terms containing the empirical variance of the variables.

Managing the search step Short runs of EM in strategies em-EM are stopped using the criterion (4) and a new short run of EM is started again at random until no more iteration is available for this step. For instance suppose that 50 iterations are available for em and that the first short run of EM takes 14 iterations. Thus, there are 36 iterations left for em. Suppose that a second short run of EM takes 19 iterations, then there are 17 iterations left for additional short EM runs, etc. The CEM algorithm is stopped when the complete data loglikelihood had reached stationarity, and started again until no more iteration is available for this algorithm as exemplified for em-EM. The solution providing the largest complete data loglikelihood is then selected to initiate EM. Obviously, the stopping rule for SEM is the total number of iterations.

4.2 Results summary

Monte Carlo experiments. Six types of data in $\mathbf{R}^2$ have been considered. Data P1 arose from a two well-separated component Gaussian mixture with $p_1 = p_2 = 0.5$, $\boldsymbol{\mu}_1 = (0,0)'$, $\boldsymbol{\mu}_2 = (2.5,0)'$, and diagonal variance matrices with $\text{diag}(\boldsymbol{\Sigma}_1) = (3,1/3)$ and $\text{diag}(\boldsymbol{\Sigma}_2) = (1/3,3)$. Data P2 arose from a two poorly separated component Gaussian mixture with $p_1 = 0.7$, $p_2 = 0.3$, $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = (0,0)'$, and diagonal variance matrices with $\text{diag}(\boldsymbol{\Sigma}_1) = (3,1/3)$ and $\text{diag}(\boldsymbol{\Sigma}_2) = (1/3,3)$. Data P3 arose from a four component Gaussian mixture with $p_1 = p_2 = p_3 = p_4 = 0.25$, $\boldsymbol{\mu}_1 = (0,-2)'$, $\boldsymbol{\mu}_2 = (2,0)'$, $\boldsymbol{\mu}_3 = (0,2)'$, $\boldsymbol{\mu}_4 = (-2,0)'$, and diagonal variance matrices with $\text{diag}(\boldsymbol{\Sigma}_1) = \text{diag}(\boldsymbol{\Sigma}_3) = (3,1/3)$ and $\text{diag}(\boldsymbol{\Sigma}_2) = \text{diag}(\boldsymbol{\Sigma}_4) = (1/3,3)$. Three alternative data P1noise, P2noise and P3noise are designed by adding noisy data arising from a uniform distribution $[-0.8,+0.8]\times[-0.8,+0.8]$ with proportion 0.2. For each type of data, we generated 30 samples of size $n = 200$. In our experiments, the algorithms were run with two components for P1, P2, P1noise and P2noise and with four components for P3 and P3noise. Moreover, we used the strategies with various numbers of iterations ITEMAX chosen with an exponential scale from an arbitrary value, namely ITEMAX= (60, 120, 240, 480, 960, ... until 15360). In what follows, we distinguish *small* number of iterations (ITEMAX < 960) and *large* number of iterations (ITEMAX ≥ 960).

Table 2 provides the percentage of times a *small* or a *large* number of iterations leads to a higher likelihood for each strategy. This table is to be read down the columns. For instance, for the strategy 1EM, 7% of times, the *small* iteration version of the method beat the *large* iteration method, 35% of times, the *large* iteration method beat the *small* iteration method, and, consequently, it means that on 58% of the occasions it was difficult to choose between them.

Table 3 provides the percentage of times a single run or repeating runs leads to a larger likelihood for strategies EM, CEM-EM and em-CEM, and the percentage of times SEMmean-EM or SEMmax-EM leads to a larger likelihood for strategies SEM-EM for all data sets. This table is to be read by rows. For instance, for the *large* iteration method, 10EM beat 1EM 36% of times, 1EM beat 10EM 5% of times, and, consequently both methods performed very similarly 59% of times. The two tables show that using a large number of iterations is highly preferable, and that, in this situation, compounding the strategy is generally profitable. It means that EM cannot be trusted to find the optimum from a single start, but that at a smaller number of iterations, the advantage of searching is lower than running. Also, it appears no surprisingly that SEMmax-EM can be preferred to SEMmean-EM to provide a higher likelihood value.

Tables 4-5 provide for each data structure and for *large* number of iterations comparisons by pairs of the best representative of each kind of strategy, namely 10EM, 10CEM-EM, 10em-EM and SEMmax-EM. In Table 4, each row gives the score of a method against an another method for each type of data and in Table 5, each row gives the mean and standard deviation of the maximum loglikelihood obtained by a method for each type of data. For instance, from Table 4, it appears that, for data P1noise, 10EM performed better than 10CEM-EM 38% of times, performed worse 27% of times, and that both methods performed very similarly 35% of times.

For data sets without noise, 10CEM-EM is outperformed by the three other strategies which provide similar results. But, the difference between 10CEM-EM and the other strategies decreases with the component separation. For noisy data, conclusions are not so clear: 10em-EM can be preferred to the other strategies except for data P3noise where 10CEM-EM outperforms all the other strategies.

	EM		CEM-EM		em-EM		SEM-EM	
nb. it.	1	10	1	10	1	10	mean	max
small	7	0	6	4	0	0	6	4
large	35	98	33	87	45	98	46	41

Table 2: Percentage of times a small or a large number of iterations leads to a higher likelihood for each strategy. Here "nb. it." is the abbreviation for number of iterations.

	EM		CEM-EM		em-EM		SEM-EM	
nb. it.	1	10	1	10	1	10	mean	max
small	79	21	67	29	88	12	11	46
large	5	36	5	20	10	12	0	6

Table 3: Percentage of times a single run or repeated runs strategies and SEMmean-EM or SEMmax-EM strategies leads to a higher likelihood. Here "nb. it." is the abbreviation for number of iterations.

	P1	P1 noise	P2	P2 noise	P3	P3 noise
10EM vs. 10CEM-EM	2-0	38-27	19-0	30-42	36-0	3-86
10EM vs. 10em-EM	0-0	8-28	1-0	7-43	5-1	39-39
10EM vs. SEMmax-EM	0-0	43-8	0-0	43-21	5-3	23-64
10CEM-EM vs. 10em-EM	0-2	6-40	0-19	2-30	1-35	83-4
10CEM-EM vs. SEMmax-EM	0-2	54-30	0-19	61-21	0-35	66-8
10em-EM vs. SEMmax-EM	0-0	57-10	0-1	63-6	5-7	29-56

Table 4: Strategy comparison by pairs with a large iteration number.

Real data sets. For each real data set, we only used the compounding version of each method with $x = 10$ repetitions and SEMmax-EM. The first example on real data sets

	P1	P1 noise	P2	P2 noise	P3	P3 noise
10EM	-659.8 (14.6)	-909.2 (13.1)	-616.1 (17.8)	-881.7 (17.3)	-754.3 (13.2)	-928.2 (13.7)
10CEM-EM	-659.8 (14.6)	-909.9 (12.5)	-617.9 (18.9)	-881.2 (18.4)	-755.6 (13.3)	-919.8 (12.3)
10em-EM	-659.8 (14.6)	-908.3 (12.3)	-616.1 (17.8)	-880.2 (17.6)	-754.3 (13.2)	-927.4 (14.0)
SEMmax-EM	-659.8 (14.6)	-911.1 (13.9)	-616.1 (17.8)	-883.7 (17.8)	-754.3 (13.2)	-925.5 (13.0)

Table 5: Means and standard deviations of maximum loglikelihood.

concerns a population of 2370 stars described by their velocity U toward the galactic center and their velocity V toward the galactic rotation (see Soubiran 1993). The second example concerns data on 272 eruptions of the Old Faithful geyser in Yellowstone National Park. Each observation consists of two measurements: the duration (in minutes) of the eruption, and the waiting time (in minutes) before the next eruption (see Venables and Ripley 1994). For those two examples we display, for the four mentioned selected strategies, the maximum likelihood values as a function of ITEMAX (Figures 2 and 4) and we depict both solutions in competition for each data set (Figures 3 and 5). Those figures provide isodensity ellipses for each component of the two mixtures in competition. The last example concerns 3641 observations in dimension five with no clear partitioning structure for which we consider a ten component Gaussian mixture. This data set concerns biological profiles of patients (Sandor 1976). Figure 6 displays the maximum likelihood values for the four strategies as a function of ITEMAX.

The reader can be surprised to see that EM strategies can lead to lower likelihood as ITEMAX increases. It is not a paradox. It must be kept in mind that each time a new value of ITEMAX is fixed the method is started afresh. Thus it is quite possible that the Search step leads to a lower maximum for EM.

As it appears from Monte Carlo experiments, a large number of iterations is required to ensure a sensible maximum likelihood value. Thus, we pay attention to experiments with ITEMAX ≥ 960. Conclusions derived from noisy simulated data sets are confirmed: 10em-EM and 10CEM-EM perform the best in most cases. They provide similar results. Otherwise, as for simulated data sets, the standard random strategy 10EM gives at best similar results than 10em-EM. Thus, this standard random strategy cannot be recommended from those experiments.

5 Discussion

We have presented and experimented simple methods to deal with the important problem of getting a sensible maximum likelihood value when using the EM algorithm in mixture inference. Our study does not include all the simple methods for choosing starting values for the EM algorithm as the methods described in Böhning (1999), Section 3.6, McLachlan

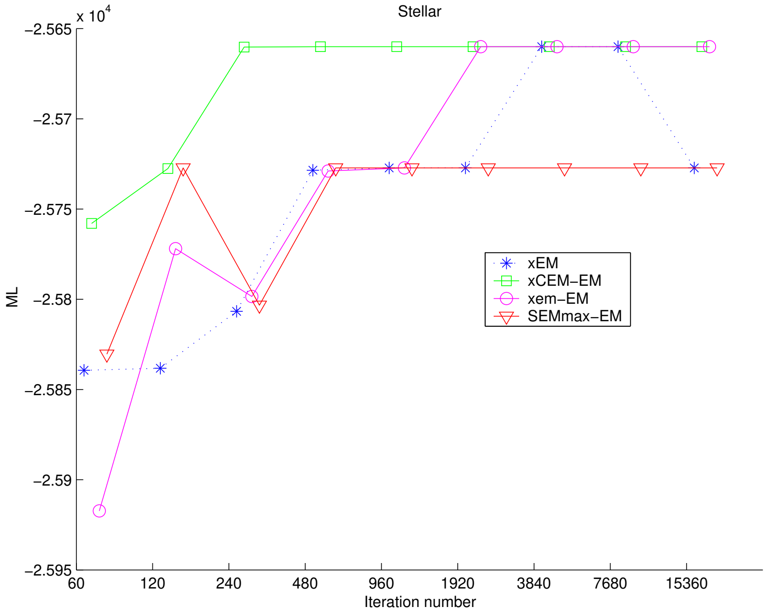
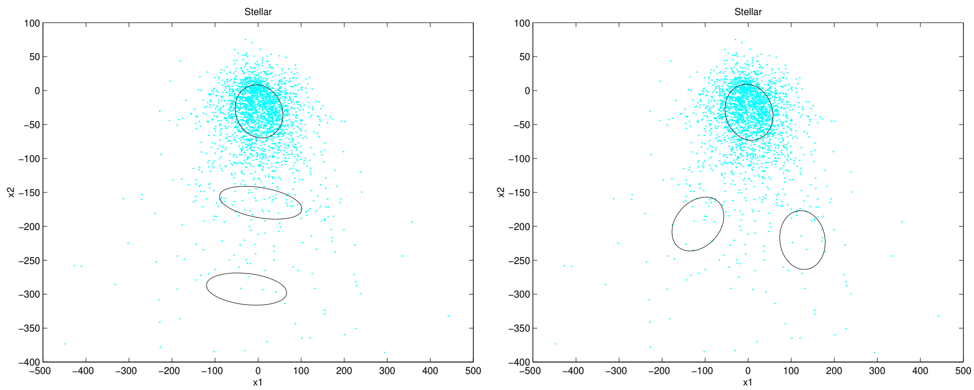


Figure 2: The loglikelihood values as a function of the available number of iterations ITEMAX for the real data set “stars”.



solution 1 solution 2

Figure 3: Two solutions in competition for the real data set “stars”. Solution 1 provides the highest likelihood.

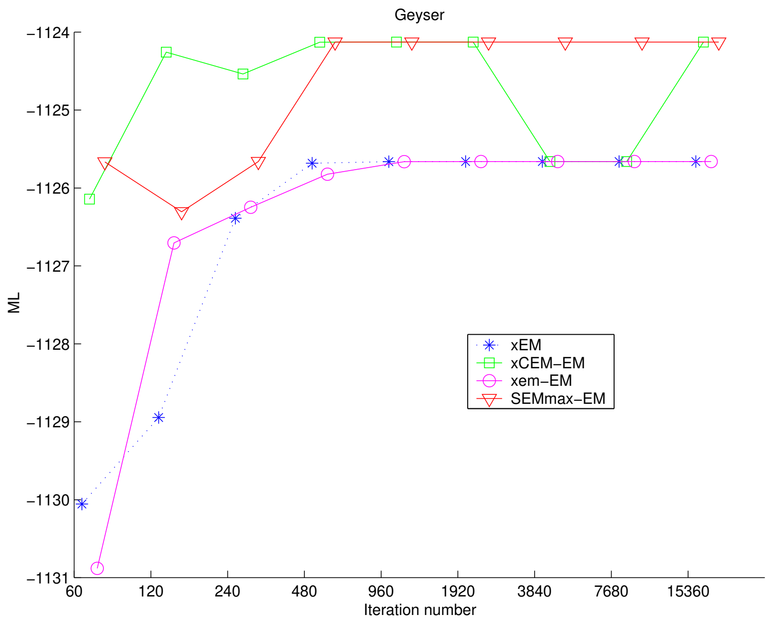


Figure 4: The loglikelihood values as a function of the available number of iterations ITEMAX for the real data set "geyser".

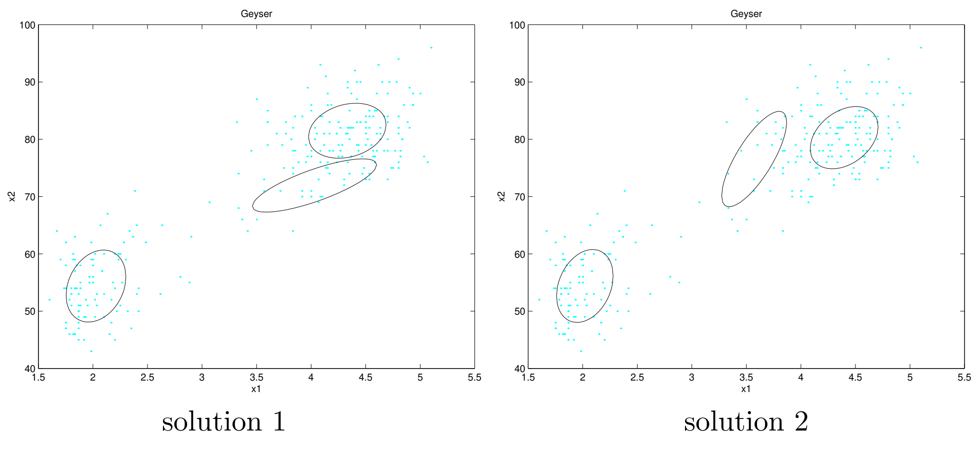


Figure 5: Two solutions in competition for the real data set "geyser". Solution 1 provides the highest likelihood.

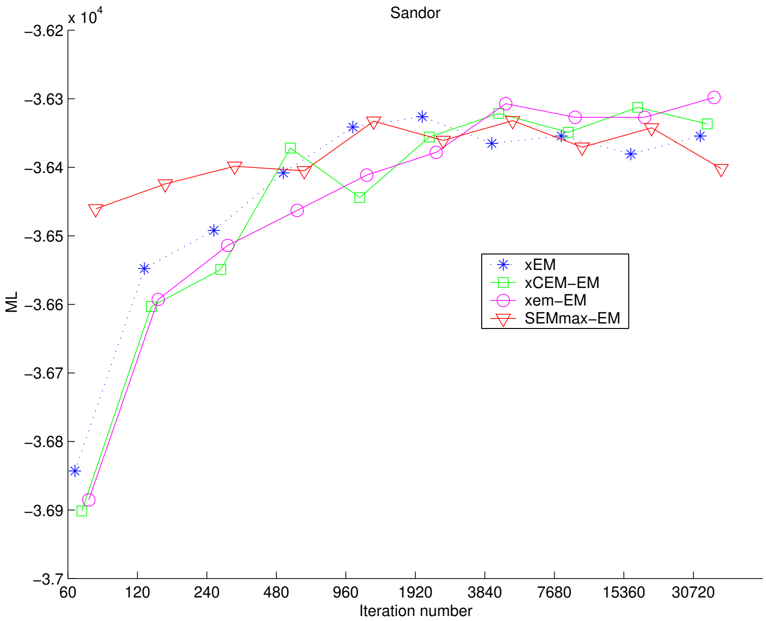


Figure 6: The loglikelihood values as a function of the available number of iterations ITEMAX for the real data set "biological profiles".

and Peel (2000) Section 2.12 and the hierarchical step of Dasgupta and Raftery (1998) which is especially efficient in the clustering context.

All the methods we experimented are obtained by combining and repeating algorithms CEM, SEM and EM in a Search/Run/Select strategy. From our experiments, the following comments can be made.

- For a good solution, do not skimp on the number of iterations.
- The value of using compounding to increase the search range is becoming obvious with the number of available iterations. Compounding can be regarded as a way of increasing the number of initial points, but focusing down on a few winners in order to save computations.
- There is no sensitive differences between the strategies, but em-EM is maybe slightly better than the other ones, and the basic S/R/S EM method is generally worse than the other ones.
- CEM-EM can perform well especially when a few iterations are available, but is the less stable strategy. In fact, CEM is more sensitive to the starting value than EM, which in turn is more sensitive to the initial values than SEM (see Celeux and Govaert 1992). The main advantage of CEM is to converge in a few steps. However, due to the presence of the C step, this algorithm provides always a suboptimal solution. Thus, it is not surprising that the method of short EM runs is more appropriate.

- Finally, we can sketch the following procedure: For the data set at hand, the cost of a single iteration of the EM-like algorithms can be easily evaluated. Thus, the number of available number of iterations ITEMAX can be decided. If ITEMAX can be regarded as large with respect to the complexity of the problem, we recommend to use $x$em-EM although its sensitivity remains to be studied. (That is the default strategy in our software MIXMOD.) Otherwise, compounding can be worse than useless and it is difficult to recommend one of the methods we experimented.

We can add the following remarks.

Here we focused on heuristics to get the highest likelihood value. But, in practice, especially when spurious local maximizers can occur, it may appear to be more interesting to select the local maximizer which has the largest attraction region because such a maximizer can be thought of as more stable. We did not deal with this problem in the present paper, avoiding the possibility of spurious local maximizers in our experiments, but proposing heuristics to get stable maximizers deserves attention. From this point of view, it is possible that exploiting the stationary distribution of SEM could be of particular interest.

We think that the Search/Run/Select strategy we adopted can be useful in a general context and can be used without difficulty for instance for faster EM-based algorithms as those discussed by Böhning (1999) and McLachlan and Peel (2000).

Acknowledgements. We are grateful to the Associate Editor and the two Referees for their helpful comments and suggestions.

References

Akaike, H., A new look at the statistical identification model, *IEEE Trans. Auto. Control*, (1974), **19**, 716-723.

Böhning D., *Computer-Assisted Analysis of Mixtures and Applications: Meta-Analysis, Disease Mapping and Others*, (Chapman & Hall, New York 1999).

Celeux, G. and Govaert, G., A Classification EM Algorithm and two Stochastic Versions, *Computational Statistics and Data Analysis*, (1992), **14**, 315-332.

Celeux, G., Chrétien, S., Forbes, F. and Mkhadri, A., A Component-wise EM Algorithm for Mixtures, *Journal of Computational and Graphical Statistics*, (2001), **10**, 699-712.

Dasgupta, A. and Raftery, A. E., Detecting features in spatial point process with clutter viamodel-based clustering, *Journal of the American Statistical Association*, (1998), **93**, 294-302.

Dempster, A. P., Laird, N. M. and Rubin, D. B., Maximum likelihood from incomplete data via the EM algorithm (with discussion), *J. R. Statis. Soc. B*, (1977), **39**, 1-38.

Lindsay, B. G. *Mixture Models: Theory, Geometry and Applications*, NSF-CBMS Regional Conference Series in Probability and Statistics, Vol 5, (Institute of Mathematical Statistics, California 1995).

Liu, C. and Sun, D. X., Acceleration of EM Algorithm for Mixtures Models using ECME, *ASA Proceedings of The Stat. Comp. Session*, (1997), pp. 109-114.

MacQueen J., Some methods for classification and Analysis of multivariate observations, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, eds. L. M. LeCam and J. Neyman*, University of California Press (1967) pp. 281-297.

McLachlan, G. J. and Krishnam, T., *The EM algorithm and Extensions,* (Wiley, New York 1997).

McLachlan, G. J. and Peel, D., *Finite Mixture Models,* (Wiley, New York 2000).

Meila, M. and Heckerman, D., An Experimental Comparison of Model-Based Clustering Methods, *Machine Learning*, (2001), **42**, 9-29.

Pilla, R. S. and Lindsay, B. G., Alternative EM methods for Nonparametric Finite Mixture Models, *Biometrika*, (2001), **88**, 535-550.

Sandor, G. *Sémiologie biologique des protéines sériques*, (Maloine, Paris 1976).

Soubiran, C. Kinematics of the Galaxy's steller population from a proper motion survey. *Astronomy Astrophysics*, (1993), **274**, 181-188.

Ueda, N. and Nakano, R., Deterministic Annealing EM Algorithm, *Neural Networks*, (1998), **11**, 271-282.

Venables, W. N. and Ripley, B. D. (1994), *Modern Applied Statistics with S-plus,* (Springer-Verlag, New York 1994)

Wei, G. C. G. and Tanner, M. A., A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms, *Journal of the American Statistical Association*, (1990), **85**, 699-704.