On Stochastic Versions of the EM Algorithm

Gilles Celeux, Didier Chauveau, Jean Diebolt

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On Stochastic Versions of the EM Algorithm

Gilles CELEUX - Didier CHAUVEAU - Jean DIEBOLT

N° 2514
Mars 1995

PROGRAMME 5

ISSN 0249-6399

INRIA
RHÔNE-ALPES

On Stochastic Versions of the EM Algorithm

Gilles CELEUX - Didier CHAUVEAU - Jean DIEBOLT

PROGRAMME 5 - Traitement du signal,
automatique et productique
Projet SYSTOL

Rapport de recherche n°2514 - Mars 1995 - 22 pages

ABSTRACT : We compare three different stochastic versions of the EM algorithm: The SEM algorithm, the SAEM algorithm and the MCEM algorithm. We suggest that the most relevant contribution of the MCEM methodology is what we call the simulated annealing MCEM algorithm, which turns out to be very close to SAEM. We focus particularly on the mixture of distributions problem. In this context, we review the available theoretical results on the convergence of these algorithms and on the behavior of SEM as the sample size tends to infinity. The second part is devoted to intensive Monte Carlo numerical simulations and a real data study. We show that, for some particular mixture situations, the SEM algorithm is almost always preferable to the EM and simulated annealing versions SAEM and MCEM. For some very intricate mixtures, however, none of these algorithms can be confidently used. Then, SEM can be used as an efficient data exploratory tool for locating significant maxima of the likelihood function. In the real data case, we show that the SEM stationary distribution provides a contrasted view of the loglikelihood by emphasizing sensible maxima.

KEY-WORDS : INCOMPLETE DATA MODELS, STOCHASTIC ALGORITHMS, MIXTURE OF DOSTRIBUTION, MONTE-CARLO EXPERIMENTS

Résumé : tsvp

Sur des versions stochastiques de l'algorithme EM

RESUME : Nous comparons différentes versions stochastiques de l'algorithme EM : les algorithmes SEM, SAEM et MCEM. On suggère que l'utilisation la plus pertinente de MCEM réside dans une version de type recuit simulé qui le rapproche de SAEM. On analyse particulièrement le comportement de ces algorithmes pour l'identification d'un mélange de lois de probabilité. Dans ce contexte, on passe en revue les différents résultats théoriques sur la convergence de ces algorithmes et aussi sur le comportement asymptotique de SEM. Une deuxième partie est consacrée à la comparaison des algorithmes sur la base de simulations de Monte Carlo intensives et sur une étude de cas réel. De ces études, il ressort que SEM est souvent préférable aux autres, mais que pour des mélanges très imbriqués, aucun des algorithmes ne donnent des résultats fiables. Ainsi, SEM peut être vu comme un outil efficace pour détecter les maxima locaux intéressants de la vraisemblance. De plus l'étude du cas réel, met bien en évidence le fait que la distribution stationnaire de SEM donne une vue exploitable de la fonction de vraisemblance en accentuant les différences entre les maxima locaux de cette vraisemblance.

MOTS-CLES : MODELES A DONNEES INCOMPLETES, ALGORITHMES STOCHASTIQUES, MELANGES DE LOIS DE PROBABILITE, SIMULATIONS DE MONTE CARLO

ON STOCHASTIC VERSIONS OF THE EM ALGORITHM

Gilles Celeux*, Didier Chauveau**, Jean Diebolt***

* INRIA Rhône-Alpes
** Université Marne-la-Vallée
*** CNRS, Université Paris VI

Abstract. We compare three different stochastic versions of the EM algorithm: The SEM algorithm, the SAEM algorithm and the MCEM algorithm. We suggest that the most relevant contribution of the MCEM methodology is what we call the simulated annealing MCEM algorithm, which turns out to be very close to SAEM. We focus particularly on the mixture of distributions problem. In this context, we review the available theoretical results on the convergence of these algorithms and on the behavior of SEM as the sample size tends to infinity. The second part is devoted to intensive Monte Carlo numerical simulations and a real data study. We show that, for some particular mixture situations, the SEM algorithm is almost always preferable to the EM and simulated annealing versions SAEM and MCEM. For some very intricate mixtures, however, none of these algorithms can be confidently used. Then, SEM can be used as an efficient data exploratory tool for locating significant maxima of the likelihood function. In the real data case, we show that the SEM stationary distribution provides a contrasted view of the loglikelihood by emphasizing sensible maxima.

1. Introduction

The EM algorithm (Dempster, Laird and Rubin 1977) is a popular and often efficient approach to maximum likelihood (ML) estimation or for locating the posterior mode of a distribution (Tanner and Wong 1987, Green 1990 and Wei and Tanner 1990) for incomplete data. However, despite appealing features, the EM algorithm has several well-documented limitations: Its limiting position can strongly depend on its starting position, its rate of convergence can be painfully slow and it can provide a saddle point of the likelihood function (l.f.) rather than a local maximum. Moreover, in some cases, the maximization step of EM is intractable.

Several authors have proposed various nonstochastic improvements on the EM algorithm (e.g., Louis 1982, Meilijson 1989, Nychka 1990, Silverman, Jones, Wilson and Nychka 1990, Green 1990). However, none of these improvements resulted in a completely satisfactory version of EM. The basic motivation of each of the three stochastic versions of the EM algorithm that we study in the present paper is to overcome the above-mentioned limitations of EM. These stochastic versions of EM are the SEM algorithm (Broniatowski, Celeux and Diebolt, 1983 and Celeux and Diebolt, 1985), the SAEM algorithm (Celeux and Diebolt, 1989) and the MCEM algorithm (Wei and Tanner, 1990 and Tanner, 1991). The purpose of the present paper is to compare the characteristics of these stochastic versions of EM and to focus on the relationships between MCEM and the two other algorithms.

The motivations of the introduction of a simulation step making use of pseudorandom draws at each iteration are not the same for SEM and MCEM (SAEM is nothing but a variant of SEM). On the one hand, the simulation step of SEM relies on the Stochastic Imputation Principle (SIP): Generate pseudo-completed samples by drawing potential unobserved samples from their conditional density given the observed data, for the current fit of the parameter. On the other hand, MCEM replaces analytic computation of the conditional expectation of the log-likelihood of the complete data given the observations by a Monte Carlo approximation. However, despite

Key words and phrases. Stochastic Iterative Algorithms; Incomplete Data; Maximum Likelihood Estimation; Stochastic Imputation Principle; Ergodic Markov Chain.

different motivations, both SEM-SAEM and MCEM can be considered as random perturbations of the discrete-time dynamic system generated by EM. This is the reason for their successful behavior: First, the random perturbations prevent these algorithms from staying near the unstable or hyperbolic fixed points of EM, as well as from its stable fixed points corresponding to insignificant local maxima of the l.f. As a consequence, the above mentioned possible slow convergence situations of EM are avoided. Moreover, the underlying EM dynamics helps them to find good estimates of the parameter in a comparatively small number of iterations. Finally, the statistical considerations directing the simulation step of these algorithms lead to a proper data-driven scaling of the random perturbations. In Section 2, we present each of these three algorithms in the same general setting, the ideas which underly them and their key properties. In Section 3, we show how these algorithms apply to the mixture problem. Given some reference measure, a density $f(y)$ is a finite mixture of densities from some parameterized family $\mathcal{F} = \{\varphi(x,a) : a \in A\}$ if

$$f(y) = \sum_{k=1}^{K} p_k \, \varphi(y, a_k) \tag{1.1}$$

for some finite integer K, where the weights p_k are in $(0,1)$ and sum up to one. The mixture problem consists in identifying the weighting parameters $p_1, \ldots, p_K$ and the parameters $a_1, \ldots, a_K$ of the component densities, on the basis of a sample of i.i.d. observations $y_1, \ldots, y_N$ issued from (1.1). The mixture problem and its variations and extensions are among the most relevant areas of application of the EM methodology (Redner and Walker 1984, Titterington, Smith and Makov 1985). In Section 4, the main results concerning the convergence properties of the algorithms EM, SEM, SAEM and MCEM in a mixture problem setting are reviewed. In Section 5, detailed Monte Carlo simulations illustrate and compare the practical behavior of each algorithm in several finite mixture situations. In Section 8, EM and SEM are also compared for a real data set studied by Basford and McLachlan (1985).

2. The EM algorithm and its stochastic versions

2.1 The EM Algorithm.

The EM algorithm (Dempster *et al.* 1977) is an iterative procedure designed to find ML estimates in the context of parametric models where the observed data can be viewed as incomplete. In this subsection, we briefly review the main features of EM. The observed data $\boldsymbol{y}$ are supposed to be issued from the density $\boldsymbol{g}(\boldsymbol{y}|\theta)$ with respect to some σ-finite measure $d\boldsymbol{y}$. Our objective is to estimate θ by $\hat{\theta} = \arg\max L(\theta)$, where $L(\theta) = \log \boldsymbol{g}(\boldsymbol{y}|\theta)$. The basic idea of EM is to take advantage of the usual expressibility in a closed form of the ML estimate of the complete data $\boldsymbol{x} = (\boldsymbol{y}, \boldsymbol{z})$. Here, $\boldsymbol{z}$ denotes the unobserved (or latent) data. The EM algorithm replaces the maximization of the unknown l.f. $\boldsymbol{f}(\boldsymbol{x}|\theta)$ of the complete data $\boldsymbol{x}$ by successive maximizations of the conditional expectation $Q(\theta', \theta)$ of $\log \boldsymbol{f}(\boldsymbol{x}|\theta')$ given $\boldsymbol{y}$ for the current fit of the parameter θ. More formally, let $\boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta) = \boldsymbol{f}(\boldsymbol{x}|\theta)/\boldsymbol{g}(\boldsymbol{y}|\theta)$ denote the conditional density of $\boldsymbol{z}$ given $\boldsymbol{y}$ with respect to some σ-finite measure $d\boldsymbol{z}$. Then

$$Q(\theta', \theta) = \int_{\mathfrak{Z}} \log\big(\boldsymbol{f}(\boldsymbol{x}|\theta')\big) \boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta) \, d\boldsymbol{z}, \tag{2.1}$$

where $\mathfrak{Z}$ denotes the sample space of the latent data $\boldsymbol{z}$. Given the current approximation θ^r to the ML estimate of the observed data, the EM iteration $\theta^{r+1} = T_N(\theta^r)$ involves two steps: The E step computes $Q(\theta, \theta^r)$ and the M step determines $\theta^{r+1} = \arg\max_\theta Q(\theta, \theta^r)$. This updating process is repeated until convergence is apparent.

The EM algorithm has the basic property that each iteration increases the l.f., i.e. $L(\theta^{r+1}) \geq L(\theta^r)$ with equality iff $Q(\theta^{r+1}, \theta^r) = Q(\theta^r, \theta^r)$. A detailed account of convergence properties of the sequence $\{\theta^r\}$ generated by EM can be found in Dempster *et al.* (1977) and Wu (1983). Under suitable regularity conditions, $\{\theta^r\}$ converges to a stationary point of $L(\theta)$. But when there are

several stationary points (local maxima, minima, saddle points), $\{\theta^r\}$ does not necessarily converge to a significant local maximum of $L(\theta)$. In practical implementations, the EM algorithm has been observed to be extremely slow in some (important) applications. As noted in Dempster *et al.* (1977), the convergence rate of EM (at least when the initial position is not too far from the true value of the parameter) is linear and governed by the fraction of missing information. Thus, slow convergence generally appears when the proportion of missing information is high. Moreover, when the log-likelihood surface is littered with saddle points and sub-optimal maxima, the limiting position of EM greatly depends on its initial position.

2.2 The SEM Algorithm.

The SEM algorithm (Broniatowski, Celeux and Diebolt 1983, and Celeux and Diebolt 1985, 1987) has been designed to answer the above-mentioned limitations of EM. The basic idea underlying SEM is to replace the computation and maximization of $Q(\theta, \theta^r)$ by the much simpler computation of $\boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta^r)$ and simulation of an unobserved pseudosample $\boldsymbol{z}^r$, and then to update θ^r on the basis of the pseudo-completed sample $\boldsymbol{x}^r = (\boldsymbol{y}, \boldsymbol{z}^r)$. Thus, SEM incorporates a stochastic step (S step) between the E and M steps. This S step is directed by the Stochastic Imputation Principle: Generate a completed sample $\boldsymbol{x}^r = (\boldsymbol{y}, \boldsymbol{z}^r)$ by drawing $\boldsymbol{z}^r$ from the conditional density $\boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta^r)$ given the observed data $\boldsymbol{y}$, for the current fit θ^r of the parameter. SEM basically tests the mutual consistency of the current guess of the parameter and of the corresponding pseudo-completed samples. Since the updated estimate θ^{r+1} is the ML estimate computed on the basis of $\boldsymbol{x}^r$, an analytic expression of θ^{r+1} as a function of $\boldsymbol{x}^r$ can be derived in a closed form in all relevant situations.

First, it is worth noting that, in some situations, the M step of the EM algorithm is not analytically tractable (e.g., the M step in the context of Weibull mixtures with censored data, see Chauveau 1992, or in the case of a convolution or blind source separation model, see Lavielle 1993, and Lavielle and Moulines 1995). Since SEM maximizes the log-l.f. of the pseudo-completed data, it does not involve such difficulties.

The random drawings prevent the sequence $\{\theta^r\}$ generated by SEM from converging to the first stationary point of the log-l.f. it encounters. At each iteration, there is a non-zero probability of accepting an updated estimate θ^{r+1} with lower likelihood value than θ^r. This is the basic reason why SEM can avoid the saddle points or the insignificant local maxima of the l.f. Thus, EM can be initiated with the SEM parameter estimate θ^r for which the l.f. attains its maximum value. However, SEM can be exploited in a better fashion.

The random sequence $\{\theta^r\}$ generated by SEM does not converge pointwise. It turns out that this sequence is an homogeneous Markov chain, which is irreducible whenever $\boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta)$ is positive for almost every θ and $\boldsymbol{z}$. This condition is satisfied in most contexts where SEM can be applied. If $\{\theta^r\}$ turns out to be ergodic, then it converges to the unique stationary probability distribution ψ of this Markov chain.

Since SEM can be seen as a stochastic perturbation of the EM algorithm, it is still directed by the EM dynamics (see (4.1) below). Thus, SEM can be expected to detect the most stable fixed point of EM in a comparatively small number of iterations. The degree of stability of a fixed point of EM is measured by the largest eigenvalue of the matrix $(I - J_c^{-1} J_{\text{obs}})(\theta_f)$, where I is the identity matrice, and $J_c(\theta_f)$ and $J_{\text{obs}}(\theta_f)$ are the complete and observed Fisher information matrices, respectively (see, e.g., Dempster *et al.*, 1977, and Redner and Walker 1984). As shown by numerical evidence, (see, e.g. Celeux and Diebolt 1985, 1987, 1991 and Soubiran *et al.* 1991) the stationary distribution ψ of the SEM sequence can be expected to concentrate around the stable fixed point of EM for which the information available without knowing the missing data, $J_c^{-1} J_{\text{obs}}$, is the largest. This is in accordance with the approach of Windham and Cutler (1992), who base their estimate of the number of mixture components on the smallest eigenvalue of $J_c^{-1} J_{\text{obs}}$.

This is a situation very similar to that prevailing in the Bayesian approach, except that ψ cannot be viewed as a *posterior* probability resulting from Bayes formula. As in the Bayesian perspective, all the information for inference on θ is contained in the probability distribution ψ and the empirical

mean based on a sufficiently large number of simulations of ψ provides a point estimate of θ. Since the simulation step of θ in any Bayesian sampling algorithm (see below) is replaced in SEM by a deterministic ML step, the variance matrix of ψ can be expected to be smaller than the inverse of the observed-data Fisher information matrix.

In a Bayesian perspective, the tractability of the complete data likelihood $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}|\theta)$ is viewed as that of the *posterior* density of θ given the complete data,

$$\pi(\theta|\boldsymbol{y}, \boldsymbol{z}) = \frac{\varphi(\boldsymbol{y}, \boldsymbol{z}|\theta)\, \pi(\theta)}{\int_\Theta \varphi(\boldsymbol{y}, \boldsymbol{z}|u)\, \pi(u)\, du}, \tag{2.2}$$

where $\pi(\theta)$ is the prior density on θ. In this context, it is natural to replace the M step of SEM by a step of simulation of θ^r from $\pi(\theta|\boldsymbol{y}, \boldsymbol{z})$. This is actually the essence of the Data Augmentation algorithm of Tanner and Wong (1987), which can therefore be considered as the Bayesian version of SEM. Alternatively, SEM can be recovered from the Data Augmentation algorithm by taking a suitable noninformative prior $\pi(\theta)$ and replacing the simulation step of θ^r from $\pi(\theta|\boldsymbol{y}, \boldsymbol{z})$ by an imputation step where θ^r is updated as $\theta^{r+1} = \int \theta\, \pi(\theta|\boldsymbol{y}, \boldsymbol{z}^r)\, d\theta$. See Diebolt and Robert (1994) for more details in the mixture context.

Finally, the SIP can be expected to provide a satisfactory data-driven magnitude for the random perturbations of SEM. For instance, when the sample of observed data is small and contains little information about the true value of the parameter θ, the variance of these random perturbations turns out to become large. This is natural since in such a case no guess θ^r of θ is very likely, so that the updated θ^{r+1} arising from the pseudo-completed sample $\boldsymbol{x}^r = (\boldsymbol{y}, \boldsymbol{z}^r)$ generated from $\boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta^r)$ is comparatively far from θ^r with high probability and the variance of the stationary distribution ψ of SEM is large. Such an erratic behavior makes SEM difficult to handle for small sample sizes. This is the reason why we have introduced the SAEM algorithm, described in the next subsection.

2.3 The SAEM Algorithm.

The SAEM algorithm (Celeux and Diebolt 1992) is a modification of the SEM algorithm such that convergence in distribution can be replaced by a.s. convergence and the possible erratic behavior of SEM for small data sets can be attenuated without sacrificing the stochastic nature of the algorithm. This is accomplished by making use of a sequence of positive real numbers (γ_r) decreasing to zero (with $\gamma_0 = 1$), which parallels the temperatures in Simulated Annealing (see, e.g., van Laarhoven and Aarts 1987). More precisely, if θ^r is the current fit of the parameter via SAEM, the updated approximation to θ is

$$\theta^{r+1} = (1 - \gamma_{r+1})\theta_{EM}^{r+1} + \gamma_{r+1}\theta_{SEM}^{r+1}, \tag{2.3}$$

where θ_{EM}^{r+1} (resp. θ_{SEM}^{r+1}) is the updated approximation of θ via EM (resp. SEM).

SAEM is going from pure SEM at the beginning towards pure EM at the end. The choice of the rate of convergence to 0 of γ_r is very important. Typically, a slow rate of convergence is necessary for good performance. From a practical point of view, it is important that γ_r stays near $\gamma_0 = 1$ during the first iterations to let the algorithm avoid suboptimal stationary values of $L(\theta)$. From a theoretical point of view, we will see in Section 4 that, in the mixture setting, we essentially need the conditions $\lim_{r\to\infty}(\gamma_r/\gamma_{r+1}) = 1$ and $\sum_r \gamma_r = \infty$ to ensure the a.s. convergence of SAEM to a local maximizer of the log-l.f. $L(\theta)$ whatever the starting point.

2.4 The MCEM Algorithm.

The MCEM algorithm (Wei and Tanner 1990) proposes a Monte Carlo implementation of the E step. It replaces the computation of $Q(\theta, \theta^r)$ by that of an empirical version $Q_{r+1}(\theta, \theta^r)$, based on m ($m \gg 1$) drawings of $\boldsymbol{z}$ from $\boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta^r)$. More precisely, the rth step is: (a) Generate an i.i.d. sample $\boldsymbol{z}^r(1), \ldots, \boldsymbol{z}^r(m)$ from $\boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta^r)$ and (b) update the current approximation to $Q(\theta, \theta^r)$ as

$$Q_{r+1}(\theta, \theta^r) = \frac{1}{m}\sum_{j=1}^{m} \log \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}^r(j)|\theta). \tag{2.4}$$

(c) Then, the M step provides $\theta^{r+1} = \arg\max_\theta Q_{r+1}(\theta, \theta^r)$.

If $m = 1$, MCEM reduces to SEM. If m is very large, MCEM works approximately like EM; thus it has the same drawbacks as EM. Moreover, if $m > 1$, maximizing $Q_{r+1}(\theta, \theta^r)$ can turn out to be nearly as difficult as maximizing $Q(\theta, \theta^r)$.

Wei and Tanner motivated the introduction of the MCEM algorithm as an alternative which replaces analytic computation of the integral in (2.1) by numerical computation of a Monte Carlo approximation to this integral. On the contrary, SEM does not involve an exact or approximate computation of $Q(\theta, \theta^r)$: The only computation involved in its E step is that of the conditional density $\boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta^r)$. Moreover, its M step is generally straightforward, since it consists in maximizing the likelihood $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}^r|\theta)$ of the completed sample $(\boldsymbol{y}, \boldsymbol{z}^r)$. Thus, in all situations where SEM works well, it should be preferred to MCEM.

This discussion points out that numerical integration of (2.1) is not the real interest of MCEM. From the comments of Wei and Tanner (1990) and Tanner (1991) about the specification of m, it turns out that the real interest of MCEM is its simulated annealing type version, in the spirit of the SAEM algorithm. Indeed, Wei and Tanner recommend to start with small values of m and then to increase m as θ^r moves closer to the true maximizer of $L(\theta)$. More precisely, if we select a sequence $\{m_r\}$ of integers such that $m_0 = 1$ and m_r increases to infinity as $r \to \infty$ at a suitable rate and perform the rth iteration with $m = m_r$, then we go from pure SEM ($m_0 = 1$) to pure EM ($m = \infty$) as $r \to \infty$. Since the variance of the random perturbation term then decreases to zero, the resulting MCEM version can be viewed as a particular type of simulated annealing method with $1/m_r$ playing the role of the temperature. For brevity, we call this algorithm the simulated annealing MCEM algorithm (s.a. MCEM) throughout this paper. Note that the s.a. MCEM can still be used when no tractable expression of $Q(\theta, \theta^r)$ can be derived, in contrast with SAEM. For instance, in the context of Weibull mixtures with censored data, MCEM avoids the numerical integration involved in the M step of EM, and consequently, in the M step of SAEM (Chauveau, 1992).

In a stimulating paper, Qian and Titterington (1991) have introduced several stochastic algorithms relying on a general Restoration-Maximization (RM) principle. This principle consists in restoring the missing data at each iteration. Qian and Titterington described several possible methods of restoration. The following one is closely related to MCEM. The rth step of this RM algorithm is: (a) Generate an i.i.d. sample $\boldsymbol{z}^r(1), \ldots, \boldsymbol{z}^r(m)$ from $\boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta^r)$ and compute the average $\bar{\boldsymbol{z}}^r = m^{-1} \sum_j \boldsymbol{z}^r(j)$ and (b) update the current estimation of θ as $\theta^{r+1} = \arg\max_\theta \boldsymbol{f}(\boldsymbol{y}, \bar{\boldsymbol{z}}^r|\theta)$. When $m = 1$, this RM algorithm reduces to SEM. For $m > 1$, RM coincides with MCEM whenever $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}|\theta)$ is a linear function of $\boldsymbol{z}$. This is not true in general. An important situation where this linearity occurs is the mixture problem. When $\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}|\theta)$ is not linear in $\boldsymbol{z}$, the estimators obtained through the use of RM and MCEM are different. In such a case, RM does not provide the ML estimator and can be expected to give poor results. For instance, in estimating the coefficients of the variance matrix of a multivariate Gaussian vector from a sample with values missing at random, it can be shown that RM introduces an underestimating bias. On the other hand, the RM algorithm avoids the potential computational drawbacks of MCEM.

Wei and Tanner established no convergence result for MCEM or its simulated annealing version. In Section 4, we make reference to a theorem which ensures the a.s. convergence of the simulated annealing MCEM to a local maximizer of $L(\theta)$ for suitable sequences $\{m_r\}$, under reasonable assumptions, in the mixture setting. This result shows the interest of this version of MCEM. It is proved in Biscarat (1994) and has been derived from previous results (Celeux and Diebolt 1992) about the convergence of SAEM.

3. A basic example: The mixture case

3.1. The incomplete data structure of mixture data.

We now focus on the mixture of distributions problem. It is one of the areas where the EM methodology has found its most significant contributions. Many authors have studied the behavior

of EM in this context from both a practical and a theoretical point of view: e.g., Redner and Walker (1984), Titterington, Smith and Makov (1985), Celeux and Diebolt (1985), McLachlan and Basford (1989) and Titterington (1990). For simplicity, we will restrict ourselves to mixtures of densities from the same exponential family (see, e.g., Redner and Walker 1984 and Celeux and Diebolt 1992).

The observed i.i.d. sample $\boldsymbol{y} = (y_1, \ldots, y_N)$ is assumed to be drawn from the mixture density

$$f(y) = \sum_{k=1}^{K} p_k \, \varphi(y, a_k), \tag{3.1}$$

where $y \in \mathbb{R}^d$, the mixing weights $0 < p_k < 1$ sum up to one and

$$\varphi(y, a) = D(a)^{-1} n(y) \exp\langle a, b(y) \rangle, \tag{3.2}$$

where a is a s-dimensional vector parameter, $n : \mathbb{R}^d \to \mathbb{R}$ and $b : \mathbb{R}^d \to \mathbb{R}^s$ are sufficiently smooth functions, $D(a)$ is a normalizing factor and $\langle \ , \ \rangle$ is the standard inner product over $\mathbb{R}^d$. The parameter to be estimated $\theta = (p_1, \ldots, p_{K-1}, a_1, \ldots, a_K)$ lies in some subset Θ of $\mathbb{R}^{K-1+sK}$.

In this context, the complete data can be written $\boldsymbol{x} = (\boldsymbol{y}, \boldsymbol{z}) = \big((y_i, z_i)\big)_{1 \le i \le N}$, where each vector of indicator variables $z_i = (z_{ij}, j = 1, \ldots, K)$ is defined by $z_{ij} = 1$ or 0 depending on whether the ith observation y_i has been drawn from the jth component density $\varphi(y, a_j)$. Owing to independence, $\boldsymbol{k}(\boldsymbol{z}|\boldsymbol{y}, \theta)$ can be split into the product $\prod_{i=1}^{N} k(z_i|y_i, \theta)$, where the probability vector $k(z|y, \theta)$ is defined by

$$k(z|y, \theta) = \frac{p(z) \, \varphi(y, a(z))}{\sum_{h=1}^{K} p_h \, \varphi(y, a_h)}, \tag{3.3}$$

with $p(z) = p_j$ and $a(z) = a_j$ iff $z_j = 1$. Conditionally on the observations, the probability that y_i has been drawn from the jth component is

$$t_j(y_i, \theta) = \frac{p_j \, \varphi(y_i, a_j)}{\sum_{h=1}^{K} p_h \, \varphi(y_i, a_h)}. \tag{3.4}$$

The log-l.f. takes the form $L(\theta) = \sum_{i=1}^{N} \log\big(\sum_{j=1}^{K} p_j \, \varphi(y_i, a_j)\big)$ and (Titterington *et al.* 1985)

$$Q(\theta', \theta) = \sum_{i=1}^{N} \sum_{j=1}^{K} t_j(y_i, \theta) \big(\log p'_j + \log \varphi(y_i, a'_j)\big). \tag{3.5}$$

3.2. EM.

The E step of the EM algorithm computes the *posterior* probabilities $t_{ij}^r = t_j(y_i, \theta^r)$, $i = 1, \ldots, N$ and $j = 1, \ldots, K$, according to (3.3) and (3.4) and the M step provides the updating formulas

$$p_j^{r+1} = \frac{1}{N} \sum_{i=1}^{N} t_{ij}^r \quad \text{and} \quad a_j^{r+1} = \frac{\sum_{i=1}^{N} t_{ij}^r b(y_i)}{\sum_{i=1}^{N} t_{ij}^r}, \quad j = 1, \ldots, K. \tag{3.6}$$

3.3. SEM and SAEM.

The E step of SEM is the same as above. The S step independently draws each z_i^r, $i = 1, \ldots, N$, from a multinomial distribution with parameters $(t_{ij}^r, j = 1, \ldots, K)$. If

$$\frac{1}{N} \sum_{i=1}^{N} z_{ij}^r \ge c(N) \quad \text{for all } j = 1, \ldots, K, \tag{3.7}$$

then it goes to the M step below. Here, $c(N)$ is a threshold satisfying $0 < c(N) < 1$ and $c(N) \to 0$ as $N \to \infty$. The role of condition (3.7) is to avoid absorbing points of the Markov Chain generated by SEM, and numerical singularities in the M step. Typically, we chose $c(N) = (d+1)/N$ (Celeux and Diebolt 1985). If (3.7) is not satisfied, then the new z_i^r's are drawn from some preassigned distribution on $\mathfrak{Z}$ such that (3.7) holds and then the algorithm goes to the M step. The M step provides

$$p_j^{r+1} = \frac{1}{N}\sum_{i=1}^{N} z_{ij}^r \quad \text{and} \quad a_j^{r+1} = \frac{\sum_{i=1}^{N} z_{ij}^r b(y_i)}{\sum_{i=1}^{N} z_{ij}^r}, \quad j = 1, \ldots, K. \tag{3.8}$$

The formulas for the SAEM algorithm can be directly derived from the above descriptions of EM and SEM and from (2.3).

3.4. MCEM.

We now turn to the description of the MCEM algorithm in the mixture case. Again, the E step is as in Subsection 3.2. The Monte Carlo step generates m independent samples of indicator variables $z^r(h) = \big(z_1^r(h), \ldots, z_N^r(h)\big)$ $(h = 1, \ldots, m)$ from the conditional distributions $\big(t_{ij}^r, j = 1, \ldots, K\big)$, $i = 1, \ldots, N$. Thus, from the definition of MCEM, the updated approximation θ^{r+1} maximizes

$$Q_{r+1}(\theta, \theta^r) = \frac{1}{m}\sum_{h=1}^{m}\sum_{i=1}^{N}\Big\{\log p(z_i^r(h)) + \log \varphi\big(y_i, a(z_i^r(h))\big)\Big\} \tag{3.9}$$

where $p(z_i^r(h)) = p_j^r$ and $a(z_i^r(h)) = a_j^r$ iff $z_{ij}^r(h) = 1$. Equality (3.9) can also be written

$$Q_{r+1}(\theta, \theta^r) = \sum_{i=1}^{N}\sum_{j=1}^{K} u_{ij}^r\big(\log p_j + \log \varphi(y_i, a_j)\big), \tag{3.10}$$

where $u_{ij}^r = \#\{h : 1 \le h \le m, z_{ij}^r(h) = 1\}/m$ represents the frequency of assignment of y_i to the jth mixture component, at the rth iteration, along the m drawings. Comparing (3.5) and (3.10), it appears that MCEM just replaces the probabilities t_{ij}^r by the frequencies u_{ij}^r in the formula (3.6) resulting from the M step of the EM algorithm.

As for SEM, the random drawings which lead to the $\boldsymbol{z}^r(h)$'s are started afresh from a suitable distribution on $\mathfrak{Z}$ if condition (3.7) is not satisfied.

As noticed above, starting with $m = 1$ and increasing m to infinity as the iteration index grows produces the s.a. MCEM algorithm, quite analogous to SAEM. If the s.a. MCEM is, in some sense, more elegant and natural than SAEM, it is dramatically more time consuming than SAEM, since it involves more and more random drawings. This will be illustrated by the simulations in Sections 5–7.

4. Convergence Properties

This section surveys the main results concerning the convergence properties of the algorithms examined in this paper, in the particular setup of finite mixtures from some exponential family, which is the area of application of EM and its various versions where the most precise results are available.

Concerning EM, Redner and Walker (1984) have proved a local convergence result in the context of mixtures from some exponential family: If the Fisher information matrix evaluated at the true θ is positive and the mixture proportions are positive, then with probability 1, for N sufficiently large, the unique strongly consistent solution θ_N of the likelihood equations is well defined and the sequence $\{\theta^r\}$ generated by EM converges linearly to θ_N whenever the starting point θ^0 is sufficiently near θ_N. Chauveau (1991) has established a similar result in the context of mixtures with censored data from exponential and certain non-exponential families.

Concerning SEM, Celeux and Diebolt (1986, 1992) have proved the ergodicity of the sequence $\{\theta^r\}$ generated by SEM in the mixture context. The proof reduces to showing that the sequence $\{\boldsymbol{z}^r\}$ is a finite-state homogeneous irreducible and aperiodic Markov chain. This result guarantees the weak convergence of the distribution of θ^r to the unique stationary distribution ψ_N of the ergodic Markov chain generated by SEM. (The index N indicates dependence on the sample). However, such a result does not guarantee that ψ_N is concentrated around the consistent ML estimator θ_N of θ. Celeux and Diebolt (1986) and Diebolt and Celeux (1993) have examined the asymptotic behavior of ψ_N. They start by showing that the SEM sequence satisfies the recursive relation

$$\theta^{r+1} = T_N(\theta^r) + V_N(\theta^r, \boldsymbol{z}^r), \tag{4.1}$$

where T_N denotes the EM operator and $V_N : \mathbb{R}^s \times \mathfrak{Z} \to \mathbb{R}^s$ is a measurable function such that $\sqrt{N}V_N(\theta^r, \boldsymbol{z}^r)$ converges in distribution as $N \to \infty$, uniformly in $\theta^r \in G_N$ (compact subset of Θ), to some Gaussian r.v. with mean 0 and positive variance matrix. In the particular case of a two-component mixture where the mixing proportion p is the only unknown parameter, they have established that the stationary distribution ψ_N of SEM is asymptotically Gaussian with mean the unique maximizer of the l.f. and variance of order $\mathcal{O}(N^{-1})$ (Diebolt and Celeux 1993). A similar result for a censored mixture can be found in Chauveau (1995). These results suggest that a similar behavior should hold in the general mixture context. Celeux and Diebolt (1986) could only prove such a result under the stringent assumption that θ_N is the unique fixed point of T_N.

Concerning SAEM, which can be expressed as

$$\theta^{r+1} = T_N(\theta^r) + \gamma_r V_N(\theta^r, \boldsymbol{z}^r) \tag{4.2}$$

(see (2.3) and (4.1)), Celeux and Diebolt (1992) have established that, in the context of finite mixtures from some exponential family, the sequence $\{\theta^r\}$ generated by SAEM converges a.s. to a local maximizer of the l.f., whatever its starting point. This result holds under the basic assumption that γ_r decreases to 0 as $r \to \infty$, $\lim_{r\to\infty}(\gamma_r/\gamma_{r+1}) = 1$ and $\sum_r \gamma_r = \infty$. For EM, the possibility of convergence to a saddle point of the l.f. is always present. This result ensures that SAEM does not converge to such a point, a.s. The basic reason why SAEM achieves better results than EM is that SAEM does not necessarily terminate in the first local maximum encountered.

Concerning the s.a. MCEM algorithm, which can be expressed as

$$\theta^{r+1} = T_N(\theta^r) + \frac{1}{\sqrt{m(r)}} U_N^r(\theta^r, \xi^r), \tag{4.3}$$

where $\xi^r = \{\boldsymbol{z}^r(h), h = 1, \ldots, m(r)\}$ represents the vector of the $m(r)$ samples drawn at iteration r and $U_N^r : \mathbb{R}^s \times \mathfrak{Z}^{m(r)} \to \mathbb{R}^s$ is a measurable function, Biscarat (1994) has established a similar result in the same context under the assumption that there exists a positive constant α such that $r^\alpha = o(m(r))$ $(r \to \infty)$.

5. The simulation procedure

One purpose of this paper is to compare the practical behavior of EM and all the proposed stochastic versions in an objective way. To this end, we proceeded intensive Monte Carlo numerical experiments. The examples that we have chosen reflect and summarize some typical situations we want to point out. However, we tried many more situations not shown here. The general description of the simulation procedure in this section is necessary to catch the wide range of the experiment.

5.1. Estimation.

First, since SEM does not provide directly a pointwise estimate, we used the two following SEM schemes to derive such an estimate. These schemes have been proposed by Celeux and Diebolt 1985, 1987 and 1991.

- **SEM-mean** is a two-stage SEM procedure: A "warm-up" step consisting of several iterations of SEM from its starting position is first performed, to reach the stationary regime.

Then we average the estimates over the next iterations. This provides a point estimate θ_{SEM}.

- **SEM-EM** is an hybrid algorithm where several SEM iterations are first performed (as a "warm-up"). Then we run EM starting from the position which achieved the largest value of the observed likelihood function $L(\theta)$, which is supposed to be a good starting position for EM.

For both these SEM algorithms, we chose the duration of the warm-up step as a fixed proportion (3/4-th) of the total number of iterations, the same for all the algorithms (see section 5.3). Hence the general procedure compares the five algorithms: EM, SEM-mean, SEM-EM, SAEM and MCEM. For SAEM, we used the cooling schedule defined from previous experiments in Celeux and Diebolt 1992, i.e.

$$\gamma_r = \cos(r\alpha)\mathbf{1}_{\{0\leq r\leq 20\}} + \frac{c}{\sqrt{r}}\mathbf{1}_{\{r>20\}}, \quad \text{where } \cos(20\alpha) = \frac{c}{\sqrt{20}} = 0.3$$

For MCEM, an equivalent cooling schedule is obtained by choosing the appropriate m_r using (4.2)–(4.3).

5.2. The mixtures under consideration.

We consider mixtures of univariate Gaussian distributions, where the parameter of component densities $\varphi(\cdot, a)$ is $a = (\mu, \sigma^2)$. The selected values were chosen to give situations somehow hard to identify; of course, for well-separated mixtures, the five algorithms perform well. Below, we give the true parameters $\theta = (p_1, \ldots, p_{K-1}, a_1, \ldots, a_K)$ of the mixtures under consideration; the corresponding densities are depicted in figure 1.

$M1$: An intricate two-component mixture, with means close enough for the resulting density to be unimodal and skewed to the right. The parameter is

$$\theta = (p_1 = 0.33, a_1 = (0, 1), a_2 = (0.8, 2.25)).$$

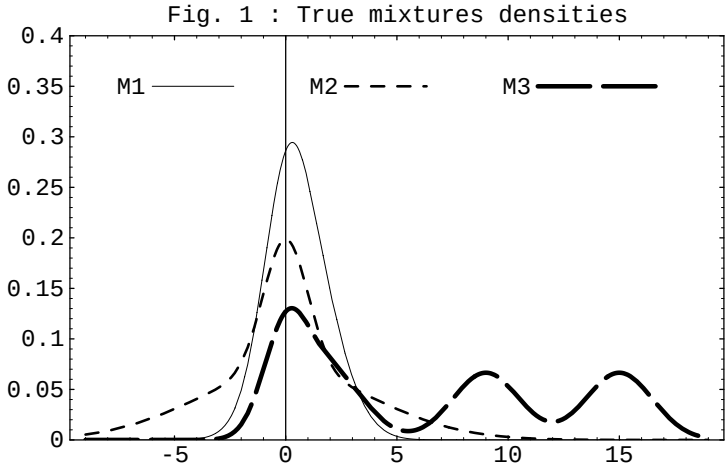
$M2$: A two-component mixture, with same zero means and different variances, resulting in a unimodal symmetric density with heavy tails. The parameter is

$$\theta = (p_1 = 0.33, a_1 = (0, 1), a_2 = (0, 16)).$$

$M3$: A four-component mixture, with two well-separated subpopulations and two intricate ones. The parameter θ is

$$(p_k = 1/4 \text{ for } k = 1, \ldots, 4, \quad a_1 = (0, 1), a_2 = (2, 2.25), a_3 = (9, 2.25), a_4 = (15, 2.25)).$$

Fig. 1 : True mixtures densities



5.3. Sample sizes and number of iterations.

Each Monte Carlo experiment was based on 50 trials (replications based on 50 different independent samples), for selected sample sizes from small, $N = 100$, to large, $N = 200$, 500, and selected numbers of iterations $n = 100$, 200, 600, 1000 or even 10 000 for some cases. The warm-up step (when needed) consisted of the first $3n/4$ iterations, and the recording step (or EM step for SEM-EM) of the last $n/4$ iterations.

5.4. Initialization.

For each trial m ($m = 1, \ldots, 50$), one initial position $\theta^0_{(m)}$ was computed using one of the three initialization schemes defined below. All the algorithms then used $\theta^0_{(m)}$ as their initial position.

We considered the following three initialization schemes:

True: just starts from the true parameter;

Equal: computes one SEM S-step starting with $t_{ij} = 1/K$ for any i and j, then the SEM M-step (3.8) yields the initial parameter;

K-means: draws K seeds $c_1, \ldots, c_K$ uniformly among the N observations and then aggregates the N observations around the c_js using a nearest neighbor procedure; the resulting K-partition determines the vectors of indicators z_i, $i = 1, \ldots, N$, from which the SEM M-step (3.8) is performed, yielding the initial position θ^0.

5.5. Restarts.

We sometimes need to restart the SEM-based algorithms, when the threshold condition given by equation (3.7) is not fulfilled. Since our goal is to give an equal chance of success to each of the algorithms, we had to define a condition similar to (3.7) for the algorithms which do not have the need of a restart procedure, namely EM and MCEM. Such a natural condition is obtained by replacing the z^r_{ij}s in (3.7) by the probabilities t^r_{ij} for EM, or the simulated frequencies u^r_{ij} for MCEM.

Also, to give an equal chance to each algorithm, we defined the restarting procedure for one trial (sample) m as follows: A starting position $\theta^0_{(m)}$ is computed using one of the methods given in 5.4, and each algorithm starts from this position. If an algorithm fails to fulfill its threshold condition for some iteration $r > 0$, it is restarted **from the same initial position** $\theta^0_{(m)}$, and its whole path is computed again. If an algorithm fails more than 2000 consecutive times in this way, a failure is recorded instead of an estimate for that algorithm and that trial m. Notice first that for EM, one restart is identical to immediate failure since the EM sequence is determined by the sample and the starting position. For the others, the random drawings along the trajectory (S-steps, or Monte Carlo steps for MCEM) give a chance of success for each restart from the same θ^0. Notice also that this procedure consisting in taking a Dirac δ_{θ^0} at θ^0 as the restarting distribution is not in favor of the SEM methodology. Indeed, one of the major reasons for an SEM-based algorithm to fail is a poor initial position (e.g., too close to the boundary). Therefore, a natural restart strategy would have been to use a diffuse restart distribution, in order to restart from a possibly better initial position. The point is that using such a strategy would have lead to **non-comparable** trials: In that case, for each trial m, we would in general have observed results given by algorithms started from different initial positions, say $\theta^0_{(m)}$(EM), $\theta^0_{(m)}$(SEM-mean), $\theta^0_{(m)}$(SEM-EM), $\theta^0_{(m)}$(SAEM) and $\theta^0_{(m)}$(MCEM).

5.6. Switching the estimates.

One well-known problem arising in the mixture setting is that the likelihood function attains its largest local maximum at several different choices of θ (see, e.g., Redner and Walker 1984). Indeed, the value of $L(\theta)$ will not change if the component pairs (p_i, a_i) and (p_j, a_j) for some i and j are interchanged in θ. This effect of a label switching is usually of no concern when the goal is just to find an estimate for a particular mixture on the basis of real data, since one estimate is just as good as any other obtained through a label switching.

Unfortunately, in a Monte Carlo experiment based on several replications of the same situation,

this switching is of great importance. Even in situations where a largest local maximum is "easy to find", the sequence θ^r generated by an EM or SEM-type algorithm can actually converge to any one of the $K!$ equivalent maxima obtained by label switching among the pairs (p_i, a_i) for $1 \leq i \leq K$. In that case, computing statistics such as averages over several replications can (and often does) lead to wrong conclusions. For example, with two components of an intricate mixture for which the equivalent maxima are mutually close (such as $M1$), the label switching can occur for roughly 50% of the trials. Computing the average over the trials without taking care of a possible label switching would obviously result in averaging heterogeneous component estimates.

To overcome this problem, we had to determine whether or not a label switching occurred in the sequence which produced an estimate θ, for each particular trial. A reasonable idea is to compare the estimate $\hat{\theta}$ given by the algorithm under consideration with the known true parameter θ^*, e.g. by computing some "distance" $d(\hat{\theta}, \theta^*)$, and then trying to find which permutation among the pairs $(\hat{p}_i, \hat{a}_i)$ in $\hat{\theta}$ minimizes $d(\hat{\theta}, \theta^*)$. We checked several possible expressions for d, like an Euclidean distance, or a distance based on just one coordinate in θ. We finally retained three methods for deciding whether or not a switch occurred, and trying to switch back to the right permutation. These methods behaved properly for the simple situations where we could switch back to the right permutation by just looking closely at the results. We expect that they work also in more intricate situations. We tried the three following methods.

Mean: Based on the distribution means $\mu_k = \mathbb{E}(X)$ for $X \sim \varphi(x, a_k)$, this method tries to find the switch in $\hat{\theta}$ for which the distance

$$d(\hat{\theta}, \theta^*) = \sum_{k=1}^{K} |\hat{\mu}_k - \mu_k^*|$$

is the smallest. As expected, this method works well with mixtures with significantly different means.

Var: Same as *Mean*, but based on distribution variances. This method works well with mixtures with same means and significantly different variances, such as $M2$.

%Class: This method uses a measure of accuracy of the estimate, namely the percentage of individuals in the sample assigned by the algorithm to their original subpopulation. This measure is discussed in 5.7 below. The method tries to switch back to the permutation maximizing this measure, and can be expected to behave well for more general situations than the previous ones.

5.7. Comparing the algorithms.

One of the major difficulties was to extract a good comparison of the algorithms under consideration from the results given by these extensive simulations. One obvious way to do this should be to find the algorithm giving the most accurate estimates. But things are not so simple, since there are many results for each mixture $M1$, $M2$ or $M3$, depending on various sample sizes, number of iterations, initialization schemes. Finding an overall behavior for the algorithms in so many situations is not easy. Hence we just outlined significant tendencies revealed by the experiments. We based our judgments on the estimates and the following considerations.

One important property in the mixture setting is the ability for the estimation procedure to properly separate the components.This means that a measure like some Euclidean distance between the true parameter and its estimate is not necessarily the best and only choice (we did not even compute such a distance since there is no meaningful way of choosing weights for the coordinates). For this reason, we computed a measure of the ability to find back the original partition among the K components, denoted in the tables *%Class*. This measure is just the ratio (in percentage) of the number of individuals in the sample, assigned by an algorithm to their proper component. Since we use simulated data, the true component from which each observation y_i has been drawn, say z_i^*, is known. We define the assignment $\hat{z}_i$ made by an algorithm for the observation y_i by

$$\hat{z}_i = \underset{1 \leq k \leq K}{\arg\max} \left(\hat{t}_{ik}\right),$$

where $\hat{t}_{i,k} = t_k(y_i, \hat{\theta})$ given by (3.4) and $\hat{\theta}$ is the point estimate given by the algorithm (the computation of this estimate depends on the selected algorithm). We obtain the *%Class* by comparing z_i^* and $\hat{z}_i$ for $1 \leq i \leq N$. Preliminary experiments with well-separated mixtures showed that this measure is a one-dimensional good indicator. Note that this indicator has already been employed by Ganesalingam (1989) and Celeux and Govaert (1993) to evaluate the performance of mixture estimation methods.

We also provide some additional information to help comparisons. These are explained below. They mainly concern the number of failures and restarts necessary for an algorithm to find an estimate.

5.8. Reading the tables.

Each table is relative to one experiment. One experiment consists of the selection of the following factors :

(1) a mixture ($M1$, $M2$ or $M3$);
(2) a sample size N;
(3) a number of iterations n;
(4) an initialization method (*True*, *Equal*, *K-means*);
(5) a switching method (*Mean*, *Var*, *%Class*);

and of the computation of 50 replications based on these factors.

In each table, there is one column (TRUE) for the true values of the parameters, and one column (MLE) for the maximum likelihood estimates of the parameters, based on the complete data (i.e. knowing the indicator z_i of the component density for each observation i). This estimate gives insight about the available information in the complete data (and helps to check the random number generator). The subsequent columns give the estimates and other information for the five selected algorithms.

The information given by the first set of rows are :

Failed: Number of trials (over 50 replications) for which the algorithm failed during 2000 consecutive restarts.
Restarts: Number of restarts, averaged over the successful trials.
RepRest: Number of successful trials which required **at least** one restart.
Time(ms): Average elapsed time per trial (on a SUN-Sparc 10 workstation).
NbSwitch: Frequency of component switching per trial.
%Class: The measure (the overall empirical success rate), described in 5.7, averaged over the successful trials.

The subsequent rows give the parameter estimates, together with their standard deviation computed over the 50 replications. We point out that **this standard deviation is not a measure of the variance of the estimates**.

5.9. Computer and Softwares.

The programs for computing the simulation procedure described in this section were written on a SUN SPARC 10–51 workstation in SPARCompiler PASCAL 3.0.1., using double precision. The floating point software follows IEEE standard 754, and the pseudorandom number generator used is the routine `d_addran_()` defined in the IEEE math library `libm.a`. Figures were drawn using *Mathematica 2.1.*

6. Selected results

As we said before, the results given in this section reflect typical behaviors among roughly 120 experiments, for the three models $M1$, $M2$ and $M3$. In particular, we give more results for situation $M1$, which is difficult to handle.

6.1. Mixture with equal means.

We ran several experiments for the mixture model $M2$, several sample sizes, iteration numbers, and the three initialization schemes. It appeared that the SEM-mean and SEM-EM algorithms were almost always better than EM or the simulated annealing versions SAEM and MCEM. The difference was more significant for the *Equal* or *K-means* initializations, since in those cases θ^0 can be a very bad guess. The table 1 below summarizes the typical results we obtained for that situation and moderate sample sizes, with a large number of iterations ($n > 200$).

We highlight the overall measure of accuracy *%Class*, which is particularly significant, and separates the SEM-based algorithms (SEM-mean and SEM-EM) from **all** the other ones. The estimates are also significantly better for SEM-mean and SEM-EM, especially μ_2, σ_1^2 and σ_2^2. Moreover, standard deviations over replications are also smaller for SEM-mean and SEM-EM than for any of the others. We can see that SEM-mean needed 65 restarts in average per replication, but never failed over the 50 trials (this means that it never needed more than 2000 restarts). Note that EM is the only one which failed twice.

Table 1: mixture $M2$, $N = 200$, $n = 600$, init *K-means*, switching *Var*

ITEMS	TRUE	MLE	EM	SEM-mean	SEM-EM	SAEM	MCEM
Failed			2	0	0	0	0
Restarts			0.000	65.100	29.020	11.680	3.480
RepRest			0	25	26	12	7
Time(ms)			4667	32613	23691	9823	81435
NbSwitch			0.354	0.460	0.500	0.360	0.360
%Class			**67.010**	**72.100**	**72.130**	**67.650**	**68.050**
p_1	0.333	0.338	0.377	0.326	0.368	0.410	0.361
$\sigma(p_1)$		0.038	0.277	0.164	0.153	0.244	0.250
μ_1	0.000	0.005	0.046	0.041	0.023	0.223	0.294
$\sigma(\mu_1)$		0.115	3.674	1.672	1.642	2.664	3.149
σ_1^2	1.000	0.969	2.026	1.116	1.334	2.030	1.829
$\sigma(\sigma_1^2)$		0.167	2.149	1.311	1.253	2.183	2.073
μ_2	0.000	0.028	0.398	0.007	0.012	0.147	0.134
$\sigma(\mu_2)$		0.385	2.561	1.493	1.423	2.510	2.332
σ_2^2	16.000	15.268	12.693	14.240	14.905	13.331	13.221
$\sigma(\sigma_2^2)$		2.178	5.718	3.972	3.939	5.435	5.485

6.2. Intricate mixture.

The mixture $M1$ is harder to recover than $M2$. This is a situation where things work better for SEM in some experiments, and for EM considering other experiments: A general tendency is not clear. A detailed discussion about this situation is given in section 7.

We first notice that, in the $M1$ situation, the overall empirical success rate *%Class* often separates the five algorithms into three sets:

- the EM algorithm;
- the SEM-based algorithms (SEM-mean and SEM-EM);
- the simulated annealing based algorithms (SAEM and MCEM).

This is significant in tables 2 and 3, and is in favor of using this measure for comparisons.

For that mixture, as expected in section 5.6, a label switching was detected by the switching method more than 40% of the time.

Table 2. In table 2, the measure *%Class* is clearly in favor of SEM-EM and SEM-mean against the other algorithms, but the estimates are not as good as in the previous situation.

If we consider the ability to separate the means $\mu_1 = 0$ and $\mu_2 = 0.8$, then SEM-mean performs better. But the SEM estimates for p_1 and σ_1^2 are poor. We also highlight the fact that SEM-mean needed more than 387 restarts in average to succeed, and SEM-EM 259 restarts. A consequence is that SEM-mean required more computing time than the others. Moreover, SEM-mean failed 24 trials out of the 50 ones, so the estimate averages were computed on the basis of 26 successful trials only.

We should also notice that in this experiment, the sample size is small ($N = 100$), and we ran a comparatively large number of iterations ($n = 600$). This and the intrication of $M1$ explain the large number of restarts needed by the SEM-based algorithms. Notice also that EM itself failed 6 trials.

Table 2: mixture $M1$, $N = 100$, $n = 600$, init *K-means*, switching *%Class*

ITEMS	TRUE	MLE	EM	SEM-mean	SEM-EM	SAEM	MCEM
Failed			6	24	10	0	0
Restarts			0.000	387.423	259.700	43.260	3.680
RepRest			0	25	38	22	10
Time(ms)			4401	350626	186946	23301	88668
NbSwitch			0.432	0.423	0.400	0.520	0.460
%Class			**61.636**	**65.577**	**64.825**	**62.640**	**62.760**
p_1	0.333	0.326	0.273	0.186	0.165	0.247	0.216
$\sigma(p_1)$		0.043	0.210	0.248	0.233	0.208	0.210
μ_1	0.000	-0.035	0.578	0.332	0.503	0.425	0.526
$\sigma(\mu_1)$		0.195	1.697	1.535	1.572	1.441	1.681
σ_1^2	1.000	0.973	0.855	0.316	0.365	0.717	0.675
$\sigma(\sigma_1^2)$		0.206	0.992	0.576	0.855	0.963	0.975
μ_2	0.800	0.784	0.830	0.780	0.750	0.754	0.747
$\sigma(\mu_2)$		0.160	0.567	0.542	0.530	0.535	0.527
σ_2^2	2.250	2.170	1.566	1.683	1.706	1.682	1.637
$\sigma(\sigma_2^2)$		0.412	0.539	0.611	0.569	0.607	0.517

Table 3 . This is an example where the *%Class* is just slightly better for the SEM-based algorithms, and not really informative. Indeed, considering the estimates, we see that none of the algorithms succeeded in separating the two means. There is even an inversion (i.e. $\hat{\mu}_1 > \hat{\mu}_2$) for EM, SEM-EM and MCEM. Estimation of the variance σ_1^2 is especially poor for the SEM-based algorithms, and estimation of σ_2^2 is poor for all the algorithms.

Notice that the number of restarts and failures for SEM is much smaller than in the previous case, due to the larger sample size.

Table 3: mixture $M1$, $N = 200$, $n = 600$, init *K-means*, switching *%Class*

ITEMS	TRUE	MLE	EM	SEM-mean	SEM-EM	SAEM	MCEM
Failed			1	8	0	0	0
Restarts			0.000	147.452	99.240	6.900	0.200
RepRest			0	38	42	8	6
Time(ms)			8840	322395	150505	20673	158388
NbSwitch			0.469	0.476	0.480	0.360	0.420
%Class			60.816	62.238	62.820	61.630	61.010
p_1	0.333	0.337	0.306	0.239	0.201	0.287	0.263
$\sigma(p_1)$		0.033	0.244	0.214	0.207	0.232	0.227
μ_1	0.000	0.003	0.879	0.634	0.981	0.770	0.864
$\sigma(\mu_1)$		0.135	1.415	1.618	1.629	1.402	1.468
σ_1^2	1.000	0.993	1.076	0.637	0.793	1.063	1.071
$\sigma(\sigma_1^2)$		0.163	1.058	0.678	1.022	1.035	1.089
μ_2	0.800	0.818	0.818	0.755	0.674	0.788	0.723
$\sigma(\mu_2)$		0.137	0.628	0.413	0.472	0.618	0.595
σ_2^2	2.250	2.223	1.767	1.844	1.828	1.714	1.727
$\sigma(\sigma_2^2)$		0.292	0.533	0.418	0.478	0.493	0.483

Table 4 . This experiment reflects a situation with a small sample size ($N = 100$), and a small number of iterations ($n = 200$).

The *%Class* is non significant for comparing the algorithms. Here, EM, and especially MCEM, performed better in separating the means. The estimates provided by EM are in general more accurate than the others. The smaller number of restarts is explained by the smaller number of iterations required to compute a point estimate.

Table 4: mixture $M1$, $N = 100$, $n = 200$, init *K-means*, switching *%Class*

ITEMS	TRUE	MLE	EM	SEM-mean	SEM-EM	SAEM	MCEM
Failed			3	1	1	1	0
Restarts			0.000	41.429	17.061	2.531	3.200
RepRest			0	39	36	14	10
Time(ms)			1480	17219	6404	2630	10595
NbSwitch			0.468	0.510	0.510	0.449	0.460
%Class			62.170	62.082	62.939	61.714	62.180
p_1	0.333	0.337	0.330	0.308	0.210	0.315	0.232
$\sigma(p_1)$		0.049	0.223	0.254	0.210	0.207	0.218
μ_1	0.000	0.003	0.027	0.170	0.538	0.275	0.063
$\sigma(\mu_1)$		0.188	1.319	1.412	1.640	1.371	1.429
σ_1^2	1.000	0.893	0.846	0.703	0.523	0.908	0.639
$\sigma(\sigma_1^2)$		0.233	0.911	0.746	0.789	0.990	0.983
μ_2	0.800	0.797	0.968	0.945	0.747	0.881	0.829
$\sigma(\mu_2)$		0.181	0.505	0.634	0.526	0.563	0.557
σ_2^2	2.250	2.212	1.680	1.525	1.678	1.661	1.681
$\sigma(\sigma_2^2)$		0.358	0.592	0.510	0.508	0.554	0.544

All these results confirm the difficulty for any of these algorithms to recover such an intricate mixture. SEM appears slightly better than the others if used with a sufficiently large number of iterations, and according to the *%Class* comparison. EM seems to be preferable with a small number of iterations.

About EM slow convergence. A usual question about EM against SEM is the possibility of using a huge amount of EM iterations in order to overcome the inconvenient of its slow convergence. To illustrate the fact that, even with many iterations, EM can lead to non satisfactory estimates, we tried some experiments using mixture $M1$, and $n = 10\,000$ iterations! The results for one such experiment are displayed in table 5 below. Only the EM algorithm is computed, because of the huge amount of computing time it would take to run such an experiment for the five algorithms.

We notice that the means are poorly separated; estimates of μ_1 and σ_2^2 are far from the true values. Indeed, the results are not significantly better than in the previous experiments computed with more reasonable numbers of iterations.

Table 5: mixture $M1$, $N = 100$, $n = 10\,000$, init *Equal*, switching *%Class*

ITEMS	TRUE	MLE	EM
Failed			5
Restarts			0.000
RepRest			0
Time(ms)			38882
NbSwitch			0.533
%Class			61.511
p_1	0.333	0.331	0.318
$\sigma(p_1)$		0.043	0.196
μ_1	0.000	-0.028	0.531
$\sigma(\mu_1)$		0.196	1.391
σ_1^2	1.000	0.968	1.100
$\sigma(\sigma_1^2)$		0.243	1.167
μ_2	0.800	0.772	0.848
$\sigma(\mu_2)$		0.179	0.632
σ_2^2	2.250	2.206	1.519
$\sigma(\sigma_2^2)$		0.299	0.599

6.3. Mixture with four components.

The mixture $M3$ has two intricate components (labeled 1 and 2), and two components easier to recover. Table 6 illustrates the typical results we obtained.

The *%Class* is clearly in favor of the SEM-based algorithms against all the others. However, none of the algorithms properly separates the two first means $\mu_1 = 0$ and $\mu_2 = 2$. The distinction

between SEM and the others lies in the estimation of components 3 and 4. SEM-mean provides the most accurate estimates for μ_3 and μ_4, and SEM-EM is slightly less accurate. EM and the simulated annealing SAEM and MCEM give poor estimates of those means. The same conclusions can be derived for the estimates of σ_3^2 and σ_4^2. Moreover, the standard deviations over replications for these estimates (μ_k, σ_k^2), $k = 3, 4$ are significantly smaller for the SEM-based algorithms. This is particularly obvious on the last row $\sigma(\sigma_4^2)$. Notice also the frequency of switches, which is always greater than 0.95, because there are here 4! equivalent local maxima instead of 2 in the previous experiments.

Table 6: mixture $M3$, $N = 200$, $n = 600$, init *K-means*, switching *%Class*

ITEMS	TRUE	MLE	EM	SEM-mean	SEM-EM	SAEM	MCEM
Failed			4	22	15	2	2
Restarts			0.000	147.679	128.743	49.542	12.354
RepRest			0	20	27	21	13
Time(ms)			9326	249554	208432	24217	173908
NbSwitch			0.957	1.000	0.971	0.979	0.979
%Class			**70.348**	**76.768**	**74.129**	**70.698**	**69.188**
p_1	0.250	0.255	0.300	0.334	0.298	0.305	0.344
$\sigma(p_1)$		0.025	0.137	0.141	0.167	0.136	0.132
μ_1	0.000	-0.014	1.070	1.021	0.804	0.764	0.710
$\sigma(\mu_1)$		0.139	2.791	2.390	2.206	2.148	1.970
σ_1^2	1.000	0.942	1.071	1.280	1.233	1.142	1.467
$\sigma(\sigma_1^2)$		0.176	0.769	0.858	1.003	0.778	1.619
p_2	0.250	0.252	0.210	0.186	0.226	0.201	0.164
$\sigma(p_2)$		0.031	0.135	0.142	0.167	0.134	0.128
μ_2	2.000	1.957	3.306	4.311	3.778	3.418	3.668
$\sigma(\mu_2)$		0.225	3.344	4.233	4.169	3.776	3.874
σ_2^2	2.250	2.223	1.433	1.463	1.385	1.426	1.141
$\sigma(\sigma_2^2)$		0.470	1.400	1.415	1.265	1.363	1.373
p_3	0.250	0.246	0.219	0.233	0.249	0.235	0.227
$\sigma(p_3)$		0.030	0.147	0.076	0.102	0.146	0.155
μ_3	9.000	9.090	7.839	9.276	9.155	8.137	8.034
$\sigma(\mu_3)$		0.209	4.162	0.997	2.354	3.946	3.885
σ_3^2	2.250	2.193	3.458	2.465	3.322	3.743	3.847
$\sigma(\sigma_3^2)$		0.373	4.986	2.132	3.660	5.207	5.374
μ_4	15.000	15.017	12.859	14.409	13.790	12.707	12.599
$\sigma(\mu_4)$		0.197	4.414	2.721	3.768	4.742	4.740
σ_4^2	2.250	2.201	4.061	2.426	2.211	3.665	4.047
$\sigma(\sigma_4^2)$		0.405	4.378	1.016	1.005	4.057	4.605

7. Detailed executions

The Monte Carlo experiments have shown the difficulty for these algorithms to recover the intricate mixture $M1$. To figure out why the stationary distribution of SEM did not seem to always concentrate on the maxima of the likelihood function, we tried to plot the paths of some SEM-mean and EM trials.

Each figure below is relative to one trial, i.e. a selection of a set of factors (1)–(5) in section 5.8, and computation of one run for the selected algorithms, with their possible failures and restarts. Since we are interested in only one trial here, there is no need for a switch.

Each figure pictures the projection of the path of SEM-mean (and sometimes EM), over (μ_1, μ_2). The smaller dots are the SEM-mean or EM iterations, the big dots are iterations preceding a failure for SEM-mean (i.e. locations where SEM-mean needed to restart). In addition, some particular locations of interest are shown: The true value (μ_1, μ_2), the starting position θ^0, the SEM-mean estimate, and the EM estimate when EM is computed. When present, the EM iterations can be easily distinguished from the SEM-mean iterations, since the distance between two consecutive EM iterations is quite small: The EM path usually looks like a curved line from the starting position to the EM estimate.

Figure 2. This figure depicts the recovering of mixture $M1$, from a small sample and large number of iterations.

First notice that SEM-mean needed a huge number of restarts before finding an estimate: There are 1678 restarts roughly located on the two lines $m1 = 0.6$ and $m2 = 0.6$. These restarts correspond typically to situations where SEM tried to fit a unique Gaussian population, giving the other component a weight $p_k \to 0$.

This figure illustrates the ability for the stationary distribution of SEM to concentrate on the two equivalent symmetric maxima of the likelihood function obtained by the switching of the two components (close to locations $(0, 0.8)$ and $(0.8, 0)$). The initialization method *Equal* gives a θ^0 halfway from the two equivalent symmetric maxima. Fortunately, in that trial, the SEM estimate was computed while the path was close to $(0.8, 0)$, and the estimate is reasonable (with a label switching). We will see in the next figure that this is not necessarily the case.

The EM path first walked near the switched local maximum, and then went to a "restart direction": A computation over some more iterations would have lead to EM's failure.

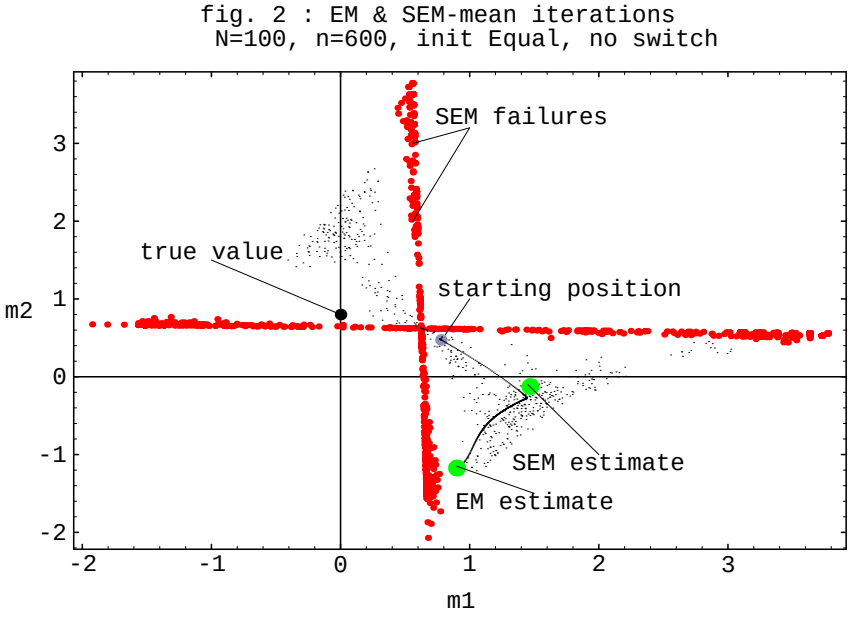
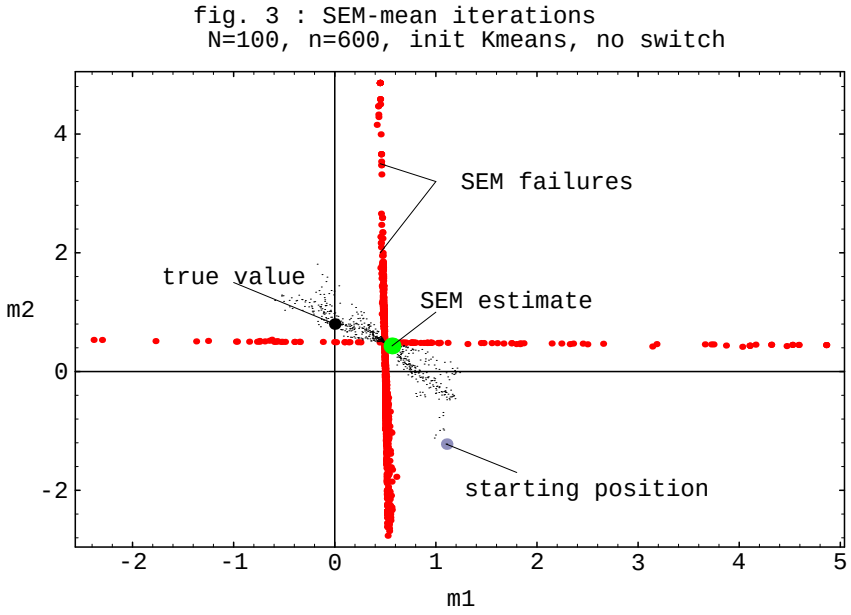


Figure 3. This figure depicts the same situation, but only SEM-mean is plotted. Notice that the initialization method *K-means* gives a starting position clearly belonging to the half-plane containing the switched value $(0.8, 0)$.

There were 1493 restarts. It is interesting to see that SEM-mean concentrates again around the two symmetric maxima, despite the initial position. However, the estimate computed by averaging the iterations in the SEM stationary regime is fairly poor, halfway from the two equivalent true values. The reason for this is clear: In its stationary regime, SEM visited alternatively the two symmetric maxima of the likelihood function, and its stationary distribution presents two equivalent modes. In such a case, computing an average over several iterations can lead to really poor estimates.



8. EM and SEM for a real data set

The data set under consideration was taken from Habbema, Hermans and van den Broek (1974). It consists in a population of 75 women (45 were Haemophilia A carriers and 30 were not) described by two variables $x_1 = 100\log(AHF$ activity) and $x_2 = 100\log(AHF$-like antigen). This data set has been used by Basford and McLachlan (1985) to illustrate the possibility of multiple roots of the likelihood equations when analyzing a mixture of multivariate normal distributions. In their case study, Basford and McLachlan considered the normal mixture model with equal variance matrices (homoscedastic model) and with unequal variance matrices (heteroscedastic model) and they identified the likelihood estimates by applying the EM algorithm. Using a wide choice of starting values, they concluded to the existence of two local maxima for the heteroscedastic model (denoted θ^1, θ^2 and displayed in table 7) and three local maxima for the homoscedastic model (denoted θ^1_*, θ^2_* and θ^3_* and displayed in table 8). In this section, the parameter subscripts denote the component indicator. Here, we use this data set to illustrate the behavior of SEM-EM and SEM-mean, defined in Section 5.1, and to describe the behavior of the stationary distribution of SEM in a situation where the existence of several local maxima is suspected. In order to have a good estimate of the SEM stationary distribution, we ran SEM for 10 000 iterations (with no warm-up step) from different starting values chosen at random or chosen among the solutions exhibited by Basford and McLachlan.

In both situations, SEM-EM always lead to the parameter estimate which provides the global maximum of the loglikelihood function. Under homoscedasticity, this global maximizer is θ^1_* given in table 8. Under heteroscedasticity, the parameter

$$\theta^3 = (0.961, -24.3, -3.3, -12.9, -6.7, 298, 126, 231, 13, 3, 0.8)$$

provides the loglikelihood -612.09. It is worth noting that the survey-refine strategy of Basford and McLachlan, consisting in initiating EM from a number of different positions, failed to locate this global maximum. Hence, SEM-EM appears to be a better survey-refine strategy, at least when the parameter space is of high dimension.

8.1. Homoscedastic model.

We now consider the homoscedastic model. First, it appears that θ_*^3 is **not** a fixed point of EM. Actually, running EM from the initial position θ_*^3 for a sufficiently large number of iterations leads to a **very slow** convergence to θ_*^2. (For a small number of iterations, θ_*^3 seems to be a fixed point of EM, hence the result in Basford and McLachlan.) It must be pointed out that a short run (100 iterations) of SEM-EM from the initial position θ_*^3 leads to an EM fixed point not detected by Basford and McLachlan; this new EM fixed point is

$$\theta_*^4 = (0.890, -21.2, -0.9, -45.4, -24.7, 235, 64, 167)$$

and its loglikelihood is -617.77. Therefore, for this homoscedastic model, SEM reveals itself as an efficient strategy to avoid slow convergence situations met with EM.

We now turn to the analysis of the SEM stationary distribution.

This analysis illustrates the possibility for SEM to produce a label switching as is apparent from figure 4, which displays the histogram of the proportion p_1 for one SEM run of 10 000 iterations: This distribution is nearly symmetric with respect to 0.5. We propose here the following heuristic in an attempt to overcome this switching difficulty: We only consider the SEM iterations such that $p_1 > 0.5$ if most of iterations satisfy this condition, and iterations for which $p_1 \leq 0.5$ otherwise. It appears that this **restricted** SEM stationary distribution does not depend upon the starting position. Table 9 summarizes this distribution from a starting value chosen at random using the *Equal* scheme described in Section 5.4. It can be noted that the median of each coordinate distribution is closer to the corresponding coordinate for the global maximizer θ_*^1 than the mean.

For the purpose of having a graphical view of the SEM restricted stationary distribution, a two-dimensional histogram is displayed for the proportion p_1 and mean μ_2^2. Actually, the proportion is an important parameter in a mixture model, and looking at table 8 and θ_*^4, it seems that μ_2^2 is able to discriminate between the fixed points of EM. Figure 5 shows that SEM concentrates most of the time around the global maximizer θ_*^1. Besides the peak around θ_*^1 there is a low ridge line joining the non-desirable local maxima θ_*^2, θ_*^3 and θ_*^4 (corresponding to the slow convergence area for EM). Thus, the SEM stationary distribution can be seen as providing a contrasted view of the loglikelihood function by flattening insensible local maxima and emphasizing sensible maxima.

Table 7: The Basford-McLachlan solutions under heteroscedasticity

	p_1	μ_1^1	μ_1^2	μ_2^1	μ_2^2	Σ_1^{11}	Σ_1^{12}	Σ_1^{22}	Σ_2^{11}	Σ_2^{12}	Σ_2^{22}	loglik.
θ^1	0.503	-11.4	-2.4	-36.4	-4.5	111	65	123	160	321	150	-613.73
θ^2	0.814	-21.9	-7.1	-32.4	12.4	305	165	184	148	87	81	-613.97

Table 8: The Basford-McLachlan solutions under homoscedasticity

	p_1	μ_1^1	μ_1^2	μ_2^1	μ_2^2	Σ^{11}	Σ^{12}	Σ^{22}	loglik.
θ_*^1	0.716	-20.6	-8.0	-32.1	7.9	265	158	171	-615.77
θ_*^2	0.528	-12.1	-1.9	-37.0	-5.2	137	100	220	-617.22
θ_*^3	0.681	-15.3	1.2	-42.0	-13.5	138	35	175	-617.42

Table 9: Summary of the empirical restricted SEM stationary distribution; $q_{1/4}$ is the first quartile, $q_{3/4}$ is the third quartile and Std is the standard deviation

	$q_{1/4}$	Median	$q_{3/4}$	Mean	Std.
p_1	0.67	0.71	0.75	0.70	0.06
μ_1^1	-21.3	-20.7	-19.0	-20.0	2.5
μ_1^2	-8.3	-7.7	-7.0	-7.0	2.7
μ_2^1	-34.8	-32.8	-30.8	-33.1	4.3
μ_2^2	4.6	7.2	9.1	5.1	7.2
Σ^{11}	239	261	273	249	40
Σ^{12}	148	156	160	145	31
Σ^{22}	162	174	188	175	19

fig. 4 : Marginal stationary distribution of SEM for the proportion p1 (10 000 iterations)

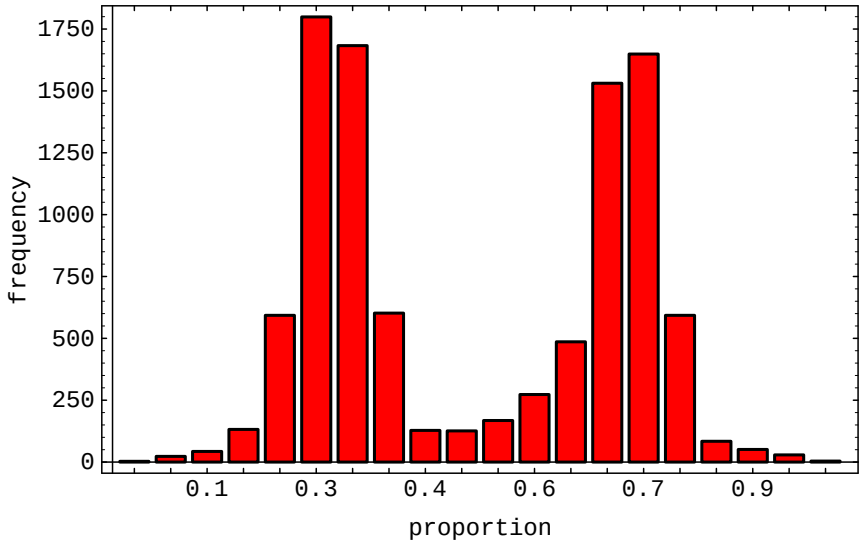
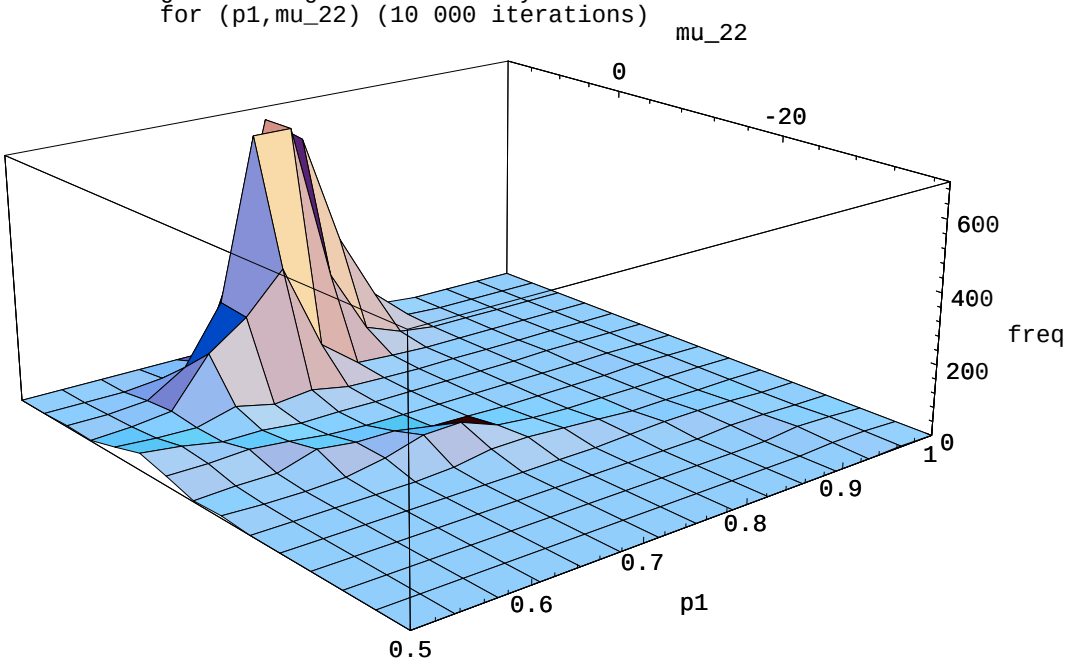


fig. 5 : Marginal stationary distribution of SEM for (p1,mu_22) (10 000 iterations)



9. Conclusion

We have seen that for some mixture situations such as $M2$, or even $M3$, that is mixtures with equal means, or mixtures with some well-separated populations, the SEM-based algorithms are in general better than EM or the simulated annealing versions.

For more intricate mixtures such as $M1$, no general preference can be proposed. The detailed executions show the basic reason why SEM-mean leads to poor estimates: The averaging method used for computing a point estimate is not adapted to a stationary distribution with multiple equivalent modes.

For such intricate mixtures for which none of the algorithms can be used confidently, SEM reveals itself as an efficient data exploratory tool. The illustrations in figures 2 and 3 together with the real data experiment show that running SEM with a very large number of iterations can be useful to locate the significant (potentially equivalent) local maxima of the likelihood function. Then, regions of interest of the parameter space Θ can be determined from SEM's results as well as practical considerations such as prior information coming from the experimental field.

References

Basford, K. E. and McLachlan, G. J. (1985), *Likelikood Estimation with Normal Mixture Models*, Applied Statistics **34**, 282–289.

Biscarat, J.C. (1994), *Almost Sure Convergence of a Class of Stochastic Algorithms*, Stochatic Processes and their Applications **50**, 83–99.

Broniatowski, M., Celeux, G. and Diebolt, J. (1983), *Reconnaissance de Mélanges de Densités par un Algorithme d'Apprentissage Probabiliste*, Data Analysis and Informatics (Diday E. *et al;* eds.) **3**, 359–374 Amsterdam, North Holland.

Celeux, G. and Diebolt, J. (1985), *The SEM Algorithm: a Probabilistic Teacher Algorithm Derived from the EM Algorithm for the Mixture Problem*, Computational Statistics Quaterly **2**, 73–82.

Celeux, G. and Diebolt, J. (1986), *Comportement Asymptotique d'un Algorithme d'Apprentissage Probabiliste pour les Mélanges de Lois de Probabilité*, Rapport de recherche INRIA **563**.

Celeux, G. and Diebolt, J. (1987), *A Probabilistic Teacher Algorithm for Iterative Maximum Likelikood Estimation*, Classification and related methods of Data Analysis, (Bock H.H. ed.), 617-623 Amsterdam, North Holland.

Celeux, G. and Diebolt, J. (1991), *The EM and the SEM algorithms for mixtures: Statistical and numerical aspects*, Cahiers du CERO **32**, 135–151.

Celeux, G. and Diebolt, J. (1992), *A Stochastic Approximation Type EM Algorithm for the Mixture Problem*, Stochastics and Stochastics Reports **41**, 119–134.

Celeux, G. and Govaert, G. (1993), *Comparison of the Mixture and the Classification Maximum Likelihood in Cluster Analysis*, Journal of Statis. Comput. Simul. **47**, 127–146.

Chauveau, D. (1991), *Extension des Algorithmes EM et SEM à la Reconnaissance de Mélanges Censurés de Distributions de Défaillances*, Ph.D. Thesis, Université Paris-Sud, Orsay, France.

Chauveau, D. (1992), *Algorithmes EM et SEM pour un mélange censuré de distributions de défaillances – Application à la fiabilité*, Revue de Statistique Appliquée **40**, 67–76.

Chauveau, D. (1995), *A Stochastic EM Algorithm for Mixtures with Censored Data*, J. Statist. Plann. Inference (to appear).

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977), *Maximum Likelikood from Incomplete Data via the EM Algorithm (with discussion)*, Journal of the Royal Statistical Society B **39**, 1–38.

Diebolt, J. and Celeux, G. (1993), *Asymptotic Properties of a Stochastic EM Algorithm for estimating mixture proportions*, Stochastic Models **9**, 599–613.

Diebolt, J. and Robert, C. P. (1994), *Estimation of Finite Mixture Distributions through Bayesian Sampling*, Journal of the Royal Statistical Society B **56**, 363–375.

Ganesalingam, S. (1989), *Classification and Mixture Approach to Clustering via Maximum Likelihood*, Applied Statistics **38**, 455–466.

Green, P. J. (1990), *On Use of the EM Algorithm for Penalized Likelihood Estimation*, Journal of the Royal Statistical Society B **52**, 443–452.

Habbena, J.D.F., Hermans, J. and van den Brock, K. (1974), *A Stepwise Discriminant Analysis Program Using Density Estimation*, Compstat 1974, Proceedings in Computational Statistics, 101–110 Wien, Physica verlag.

van Laarhoven, P. J. M. and Aarts, E. H. L. (1987), *Simulated Annealing: Theory and Applications*, Reidel: Dordrecht.

Lavielle, M. (1993), *A Stochastic Algorithm for Parametric and Non-Parametric Estimation in the case of Incomplete Data*, Signal Processing (to appear).

Lavielle, M. and Moulines, E. (1995), *On a Stochastic approximation version of the EM algorithm*, preprint, Université Paris-Sud (submitted to JRSS B).

Louis, T. A. (1982), *Finding the Observed Information Matrix when Using the EM Algorithm*, Journal of the Royal Statistical Society B **44**, 226–233.

McLachlan, G.J. and Basford, K.E. (1989), *Mixture models - Inference and applications to Clustering*, New York, Marcel Dekker.

Meilijson, I. (1989), *A Fast Improvement to the EM Algorithm on its Own Terms*, Journal of the Royal Statistical Society B **51**, 127–138.

Nychka, D.W. (1990), *Some Properties of Adding a Smoothing Step to the EM Algorithm*, Statistics and Probability Letters **9**, 187–193.

Qian, W. and Titterington, D. M. (1991), *Estimation of parameters in hidden Markov models*, Phil. Trans. R. Soc. Lond **A 337**, 407–428.

Redner, R. A. and Walker, H. F. (1984), *Mixtures Densities, Maximum Likelikood and the EM Algorithm*, SIAM Review **26**, 195–249.

Silverman, B. W., Jones, M. C., Wilson, J. D. and Nychka, D. W. (1990), *A Smoothed EM Approach to Indirect Estimation Problems, with Particular Reference to Stereology and Emission Tomography*, Journal of the Royal Statistical Society B **52**, 271–324.

Soubiran, C., Celeux, G., Diebolt, J. and Robert, C. P. (1991), *Analyse de mélanges gaussiens pour de petits échantillons: application à la cinématique stellaire*, Revue de Statistique Appliquée **39**, 17–36.

Tanner, M. A. (1991), *Tools for Statistical Inference.*, Lectures Notes in Statistics 67, New York, Springer-Verlag.

Tanner, M. A. and Wong, W. H. (1987), *The Calculation of Posterior Distribution by Data Augmentation (with discussion)*, Journal of the American Statistical Association **82**, 528–550.

Titterington, D. M. (1990), *Some Recent Research in the Analysis of Mixture Distribution*, Statistics **21**, 619–640.

Titterington, D. M., Smith, A. F. M. and Makov U. E. (1985), *Statistical Analysis of Finite Mixture Distribution*, New York, Wiley.

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

Windham, M. P. and Cutler, A. (1992), *Information Ratios for Validating Cluster Analyses*, Journal of the American Statistical Association **87**, 1188–1192.

Wu, C.F. (1983), *On the Convergence Properties of the EM Algorithm*, Annals of Statistics **11**, 95–103.

* INRIA Rhône-Alpes, Faculté de Médecine de Grenoble, Dept. de Statistique, 38700 Latronche, France

E-mail address: Gilles.Celeux@imag.fr

** Université Marne la Vallée, Équipe d'Analyse et de Mathématiques Appliquées, 2, rue de la Butte Verte, 93166 Noisy-le-Grand cedex, France.

E-mail address: chauveau@math.univ-mlv.fr

*** CNRS, URA 1321,45–55 3ème étage, Université Paris VI, 4 Place Jussieu 75252 Paris cedex, France.

E-mail address: jed@ccr.jussieu.fr