An exact line search scheme to accelerate the EM algorithm Application to Gaussian mixture models identification

Wentao Xiang, Ahmad Karfoul, Chunfeng Yang, Huazhong Shu, Régine Le Bouquin Jeannès

An exact line search scheme to accelerate the EM algorithm: Application to Gaussian Mixture Models identification

Wentao Xiang[a,b,c,d], Ahmad Karfoul[a,b,c], Chunfeng Yang[c,e], Huazhong Shu[c,e], Régine Le Bouquin Jeannès[a,b,c,*]

[a] *INSERM, U1099, Rennes, 35000, France*
[b] *Université de Rennes 1, LTSI, Rennes, 35000, France*
[c] *Centre de Recherche en Information Biomédicale sino-français (CRIBs), 35000, France*
[d] *Key Laboratory of Clinical and Medical Engineering, School of Biomedical Engineering and Informatics, Nanjing Medical University, Nanjing, 210029, China*
[e] *Laboratory of Image Science and Technology (LIST), School of Computer Science and Engineering, Southeast University, Nanjing, 210096, China*

Abstract

This paper tackles the slowness issue of the well-known Expectation-Maximization (EM) algorithm in the context of Gaussian Mixture Models. To cope with this slowness problem, an Exact Line Search scheme is proposed. It is based on exact computation of the step size required to jump, for a given search direction, towards the final solution. Computing this exact step size is easily done by only rooting a second-order polynomial computed from the initial log-likelihood maximization problem. Numerical results using both simulated and real dataset showed the efficiency of the proposed exact line search scheme when applied to the conventional EM algorithm as well as the Anti-Annealing based acceleration techniques based on either the EM or the Expectation Conjugate Gradient algorithm.

Keywords: Gaussian Mixture Models, Expectation-Maximization, Line Search strategy, Anti-Annealing techniques, Expectation Conjugate Gradient

*Corresponding author.
Email address: regine.le-bouquin-jeannes@univ-rennes1.fr (Régine Le Bouquin Jeannès)

1. Introduction

The Expectation Maximization (EM) algorithm initially proposed in [1] stands for the utmost popular algorithm in applied statistics notably for finding the maximum likelihood or maximum a posterior estimates in the presence of missing/hidden data given a set of available measurements and also for data clustering. Gaussian Mixture Models (GMMs) [2] is a powerful tool for data clustering which is of widespread applications such as in pattern recognition [3], feature selection/extraction [4], image segmentation [5], information retrieval [6], data mining [7] and in signal processing [8, 9]. GMMs-based analysis consists in modelling the dataset at hand as a linear mixture of Gaussian distributions. Identifying the GMM parameters, *i.e.* means and covariance matrices of those Gaussian distributions together with its related mixing coefficients is mandatory and efficiently performed using the EM algorithm. This is thanks to its simplicity and its proved convergence property (*e.g.* monotone convergence in likelihood values) [10, 11]. Despite these attractive properties, the convergence of the EM algorithm is still very slow in some clustering situations where (i) some mixing coefficients are small compared to other ones [12] and/or (ii) the data are relatively poorly-separated into distinct clusters [13]. To cope with the EM slowness, a number of studies have been conducted and a variety of solutions have been proposed [12, 13, 14, 15, 16, 17] to cite a few. While authors in [13, 14, 15, 16] employ the conventional optimization theory by resorting to either Newton or quasi-Newton approaches, authors in [17, 18] adopt for a hybrid EM wherein the EM algorithm is used in an early stage of the iterative process and the (quasi-)Newton scheme is employed later for a faster convergence. However, despite the efficiency of Newton-type and hybrid approaches, their use in practice is still moderate due to their high computational complexity with respect to the conventional EM method [1]. Therefore, simpler approaches have been proposed such as the Expectation Conjugate Gradient (ECG) approach [15] in which model parameters are estimated based on a gradient ascent scheme with the gradient of the log-likelihood exactly computed. Furthermore, an annealing strategy and an Anti-Annealing one were, respectively, proposed in [19] and [12], in the context of GMM with unbalanced coefficients. The key idea of the latter resides in the fact that the

posterior probability distribution is simply parametrized by a temperature parameter and a maximization of the log-likelihood is performed at each considered temperature. Beyond the aforementioned approaches, the convergence speed of the EM algorithm can be further improved using a simple but very efficient line search-based scheme.

Line Search (LS) scheme is extensively used in the optimization theory [20], especially for tensor optimization [21, 22] or for tensor decomposition [23] to cite a few. It finds its useful applicability for example when the question of accelerating the EM algorithm is addressed. The well-known Aitken acceleration procedure can be considered as a LS-like approach [24] where the partial derivatives of an appropriate mapping in the parameter domain are to be computed as a step size through a predefined search direction. Despite its efficiency, this approach requires, at each iteration, the computation of derivatives of some function which is often prone to computational issue. Furthermore, some LS-like approaches are based on the computation of either the inverse of the Hessian matrix of the objective function (*e.g.* the Newton approaches) or the inverse of its approximation (*i.e.* the Jacobian matrix) [14]. However, the latter are well-known to be numerically unstable for example in case of highly overlapped clusters. To cope with this instability issue, authors in [25] proposed a LS-like scheme where the search direction is defined as the difference between two successive estimates of the model parameters. As far as the step size is concerned, it is estimated as the mean of the ratio of the differences between individual parameter estimates obtained from the two most recent iterations [25]. More details regarding this approach are given in Section 2.3.

In this paper, an Exact LS (ELS) scheme is proposed to accelerate the convergences speed of the EM algorithm. Inspired from [21, 26] where the ELS is introduced in a pure deterministic framework, the proposed ELS procedure in this paper is applied after the E-step of the EM algorithm. The proposed ELS scheme leads to an exact computation of the step size for a given direction. This is simply done by rooting a second order polynomial computed from the considered objective function. The performance of the proposed approach is evaluated in the context of GMMs in situations where the EM algorithm suffers from slow convergence due to either unbalanced mixing coefficients or relatively high overlapped clusters. The behaviour of the proposed approach is compared

with the ones of simple but very efficient methods to accelerate the EM algorithm such as the ECG [15], the λ-EM method [25] and the Anti-Annealing based [12] algorithms.

2. Background

Let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n, ..., \boldsymbol{x}_N\}$ be a data set of N D-dimensional independent and identically distributed observation vectors $\boldsymbol{x}_n, 1 \leq n \leq N$. Under the GMM, $\boldsymbol{x}_n$ is modelled as a linear superposition of K Gaussian distributions with the following likelihood:

$$p\left(\boldsymbol{x}_n|\boldsymbol{\theta}\right) = \sum_{k=1}^{K} \alpha_k p(\boldsymbol{x}_n\left|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right.), \quad s.t.\ \alpha_k \geq 0, \ \sum_{k=1}^{K} \alpha_k = 1 \tag{1}$$

where α_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ stand for the mixing coefficient, mean vector and positive definite covariance matrix of the k-th Gaussian component. $\boldsymbol{\theta} = [{\boldsymbol{\theta}_1}^\intercal, ..., {\boldsymbol{\theta}_K}^\intercal]^\intercal$ is the global vector of parameters whose k-th component, $\boldsymbol{\theta}_k = \left[\alpha_k, {\boldsymbol{\mu}_k}^\intercal, \mathrm{vec}\left(\boldsymbol{\Sigma}_k\right)^\intercal\right]^\intercal$ ($\mathrm{vec}(.)$ is the matrix-to-vector transform), is the local vector of parameters associated to the k-th Gaussian distribution. Note that $p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ in Eq. (1) is given by:

$$p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{D/2}\det(\boldsymbol{\Sigma}_k)^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\intercal \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{x}_n - \boldsymbol{\mu}_k\right)\right\} \tag{2}$$

where $\det(\boldsymbol{\Sigma}_k)$ is the determinant of the matrix $\boldsymbol{\Sigma}_k$. Identifying the GMM consists in estimating its vector of parameters $\boldsymbol{\theta}$. Estimating the latter is performed by maximizing the likelihood of the observed data with respect to $\boldsymbol{\theta}$. However, for sake of clarity and computation facility, since the logarithm function is an increasing function, the log-likelihood formulation is used instead. Then, given the observation matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]$ of size $(D \times N)$, the optimization problem to be solved is defined as follows:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} L\left(\boldsymbol{\theta}\right) \ s.t.\ \alpha_k > 0, \ \sum_{k=1}^{K} \alpha_k = 1. \tag{3}$$

with the log-likelihood function $L\left(\boldsymbol{\theta}\right)$ defined by:

$$L\left(\boldsymbol{\theta}\right) = \log p(\boldsymbol{X}|\boldsymbol{\theta}) = \log \prod_{n=1}^{N} p\left(\boldsymbol{x}_n|\boldsymbol{\theta}\right) = \sum_{n=1}^{N} \log(\sum_{k=1}^{K} \alpha_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)). \tag{4}$$

The EM algorithm stands for the most common used algorithm to solve equation (3). This is due to its simplicity, efficiency and convergence property. Despite these attractive properties, the slowness of this algorithm is well-known in cases of unbalanced mixing coefficients and/or weakly separated clusters, as mentioned previously. Among the different algorithms proposed to cope with such drawbacks, the Expectation Conjugate Gradient (ECG) [15], the λ-EM method [25], the Anti-Annealing EM (AAEM) algorithm [12] and the Anti-Annealing ECG (AAECG) one can be considered as discussed in this paper. This is since the latter algorithms enjoy a simple structure from numerical point of view together with efficient performance, compared to other proposed solutions (the reader can refer to [14] for more details). Note that due to the simplicity of both the Anti-Annealing (AA) and the ECG methods, the AAECG method is a straightforward combination that we suggest in this paper between these two strategies. Description of the algorithms considered in this paper is given hereafter.

2.1. The EM algorithm

The EM algorithm, initially proposed in [1], deals easily with the optimization problem in Eq. (3) by considering the mixing coefficients α_k as prior probabilities for the GMM components. That is to say $p(z_n = k) = \alpha_k$ where z_n is a label variable indicating which Gaussian component is being considered for which data point. Thus, the log-likelihood in Eq. (4) can be rewritten using the complete data representation as:

$$\begin{aligned} L\left(\boldsymbol{\theta}\right) &= \sum_{n=1}^{N} \log(\sum_{k=1}^{K} p(\boldsymbol{x}_n, z_n = k|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) = \sum_{n=1}^{N} \log(\sum_{k=1}^{K} p(z_n = k)p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) \\ &= \sum_{n=1}^{N} \log(\sum_{k=1}^{K} \alpha_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)). \end{aligned} \tag{5}$$

Note that the knowledge of the latent variables z_n allows for an easy way to maximize the above log-likelihood function. Since the latter are unknown, their posterior probability distributions given the observed data point and the current estimate of $\boldsymbol{\theta}$ can however be computed. The EM algorithm [1] is an iterative process in which the algorithm alternates until convergence between two main steps: (i) The E-step where the posterior

probability distribution of the latent variables is computed:

$$\begin{aligned} &\mathrm{E}-\mathrm{step}: \\ &h_k^{(it)}(n) = \frac{\alpha_k^{(it)} p(\boldsymbol{x}_n|\boldsymbol{\mu}_k^{(it)}, \boldsymbol{\Sigma}_k^{(it)})}{\sum_{r=1}^{K} \alpha_r^{(it)} p(\boldsymbol{x}_n|\boldsymbol{\mu}_r^{(it)}, \boldsymbol{\Sigma}_r^{(it)})} \end{aligned} \tag{6}$$

and (ii) the M-step where an update of the model parameters is performed, at the $(it+1)$-th iteration, as a result of maximizing the expectation of the complete log-likelihood function, noted here by $Q\left(\boldsymbol{\theta}\left|\boldsymbol{\theta}^{(it)}\right.\right) = E\left[\sum_{n=1}^{N} \log(\alpha_k p\left(\boldsymbol{x}_n\left|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right.\right))\right]_{h_k^{(it)}(n)}$, under the posterior probability distributions, $h_k^{(it)}(n)$, computed from the E-step:

$$\begin{aligned} &\mathrm{M}-\mathrm{step}: \\ &\text{for } k = 1, ..., K \\ &\quad \alpha_k^{(it+1)} = \frac{1}{N} \sum_{n=1}^{N} h_k^{(it)}(n) \\ &\quad \boldsymbol{\mu}_k^{(it+1)} = \frac{\sum_{n=1}^{N} \boldsymbol{x}_n h_k^{(it)}(n)}{\sum_{n=1}^{N} h_k^{(it)}(n)} \\ &\quad \boldsymbol{\Sigma}_k^{(it+1)} = \frac{\sum_{n=1}^{N} \left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(it+1)}\right)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(it+1)}\right)^{\top} h_k^{(it)}(n)}{\sum_{n=1}^{N} h_k^{(it)}(n)} \\ &\text{end} \end{aligned} \tag{7}$$

Regarding the stop condition, the EM algorithm stops when a maximal number of iterations is reached or when the relative change (in absolute value) of $L(\boldsymbol{\theta})$ between two successive iterations exhibits a value that is smaller than a predefined threshold, τ:

$$\frac{|L(\boldsymbol{\theta}^{(it+1)}) - L(\boldsymbol{\theta}^{(it)})|}{L(\boldsymbol{\theta}^{(it+1)})} < \tau \tag{8}$$

2.2. The ECG algorithm

Essentially proposed to deal with the EM slowness in the case of highly overlapped clusters, the ECG algorithm [15] employs the conjugate gradient method to maximize the log-likelihood in Eq. (4). The key idea underlying the ECG approach is the established link between the step in the parameter space and the gradient of the log-likelihood function, at each iteration of the EM algorithm [11]. This link is characterized by the

so-called parameter-dependent projection matrix, denoted here by $\boldsymbol{P}$ which is a positive-definite matrix. Indeed, as the gradient computation of the log-likelihood function of the observed data is required for the optimization process, authors in [15] proposed an exact computation of this gradient based on the knowledge of both the partial derivative of the complete data probability distribution together with the posterior of the hidden variables given the observation and the current estimate of the model parameters (see [15] for more details). The ECG update rule is then defined by:

$$\widetilde{\boldsymbol{\theta}}^{(it+1)} = \widetilde{\boldsymbol{\theta}}^{(it)} + \boldsymbol{P}\left(\widetilde{\boldsymbol{\theta}}^{(it)}\right) \frac{\partial L}{\partial \widetilde{\boldsymbol{\theta}}} \mid_{\widetilde{\boldsymbol{\theta}}=\widetilde{\boldsymbol{\theta}}^{(it)}} \tag{9}$$

where $\widetilde{\boldsymbol{\theta}} = \boldsymbol{\Pi}\boldsymbol{\theta}$ with $\boldsymbol{\Pi}$ is a permutation matrix defined such that the Z-th ($Z = K(1 + D+D^2)$) dimensional vector $\widetilde{\boldsymbol{\theta}} = [\alpha_1, ..., \alpha_K, \boldsymbol{\mu}_1{}^\top, ..., \boldsymbol{\mu}_K{}^\top, \mathrm{vec}(\boldsymbol{\Sigma}_1)^\top, ..., \mathrm{vec}(\boldsymbol{\Sigma}_K)^\top]^\top$, where $\widetilde{\boldsymbol{\theta}}^{(it)}$ is the estimate of the vector of parameters $\widetilde{\boldsymbol{\theta}}$ at the it-th iteration, and:
$\boldsymbol{P}\left(\widetilde{\boldsymbol{\theta}}^{(it)}\right) =$

$$\begin{pmatrix}
\boldsymbol{P}\left(\boldsymbol{\alpha}^{(it)}\right) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & \boldsymbol{P}\left(\boldsymbol{\mu}_1^{(it)}\right) & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{P}\left(\boldsymbol{\mu}_K^{(it)}\right) & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{P}\left(\mathrm{vec}\left(\boldsymbol{\Sigma}_1^{(it)}\right)\right) & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{P}\left(\mathrm{vec}\left(\boldsymbol{\Sigma}_K^{(it)}\right)\right)
\end{pmatrix}$$

denotes a square block diagonal matrix of size $(Z \times Z)$ and:

$\frac{\partial L}{\partial \widetilde{\boldsymbol{\theta}}} \mid_{\widetilde{\boldsymbol{\theta}}=\boldsymbol{\theta}^{(it)}} =$
$\left[\left(\frac{\partial L}{\partial \boldsymbol{\alpha}}\mid_{\boldsymbol{\alpha}^{(it)}}\right)^\top; \left(\frac{\partial L}{\partial \boldsymbol{\mu}_1}\mid_{\boldsymbol{\mu}_1^{(it)}}\right)^\top, ..., \left(\frac{\partial L}{\partial \boldsymbol{\mu}_k}\mid_{\boldsymbol{\mu}_k^{(it)}}\right)^\top; \left(\frac{\partial L}{\partial vec[\boldsymbol{\Sigma}_1]}\mid_{\boldsymbol{\Sigma}_1^{(it)}}\right)^\top, ..., \left(\frac{\partial L}{\partial vec[\boldsymbol{\Sigma}_k]}\mid_{\boldsymbol{\Sigma}_k^{(it)}}\right)^\top\right]^\top$

with [27]:

$$\boldsymbol{P}\left(\boldsymbol{\alpha}^{(it)}\right) = \frac{1}{N}\left[\mathrm{diag}\left(\boldsymbol{\alpha}^{(it)}\right) - \boldsymbol{\alpha}^{(it)}\left(\boldsymbol{\alpha}^{(it)}\right)^{\mathsf{T}}\right]$$

$$\boldsymbol{P}\left(\boldsymbol{\mu}_k^{(it)}\right) = \frac{\boldsymbol{\Sigma}_k^{(it)}}{\sum\limits_{n=1}^{N} h_k^{(it)}(n)}$$

$$\boldsymbol{P}\left(vec\left(\boldsymbol{\Sigma}_k^{(it)}\right)\right) = \frac{2}{\sum\limits_{n=1}^{N} h_k^{(it)}(n)}\boldsymbol{\Sigma}_k^{(it)} \otimes \boldsymbol{\Sigma}_k^{(it)}$$

$$\frac{\partial L}{\partial \boldsymbol{\alpha}}|_{\boldsymbol{\alpha}^{(it)}} = \left[\frac{\sum\limits_{n=1}^{N} h_1^{(it)}(n)}{\alpha_1^{(it)}}, ..., \frac{\sum\limits_{n=1}^{N} h_K^{(it)}(n)}{\alpha_K^{(it)}}\right]^{\mathsf{T}}$$

$$\frac{\partial L}{\partial \boldsymbol{\mu}_k}|_{\boldsymbol{\mu}_k^{(it)}} = \sum_{n=1}^{N} h_k^{(it)}(n)\left(\boldsymbol{\Sigma}_k^{(it)}\right)^{-1}\left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(it)}\right)$$

$$\frac{\partial L}{\partial vec\left(\boldsymbol{\Sigma}_k^{(it)}\right)}|_{\boldsymbol{\Sigma}_k^{(it)}} = -\frac{1}{2}\sum_{n=1}^{N} h_k^{(it)}(n)\left(\boldsymbol{\Sigma}_k^{(it)}\right)^{-1}\left(\boldsymbol{\Sigma}_k^{(it)} - \left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(it)}\right)\left(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{(it)}\right)^{\mathsf{T}}\right)\left(\boldsymbol{\Sigma}_k^{(it)}\right)^{-1}$$

2.3. The λ−EM algorithm (Jacobian eigenvalue based acceleration)

According to Taylor series expansion for an appropriate function, $f(\boldsymbol{\theta})$, in the parameter space where it governs the transition between two successive parameter estimates, the following update rule holds valid:

$$\boldsymbol{\theta}^{(it+1)} - \boldsymbol{\theta}^{(it)} = \boldsymbol{J}^{(it)}\left(\boldsymbol{\theta}^{(it)} - \boldsymbol{\theta}^{(it-1)}\right) \tag{10}$$

where $\boldsymbol{J}^{(it)} = \frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(it)}}$. It turns out that the left side of the above equation tends for sufficiently high number of iterations, it, to the eigenvector associated to the largest eigenvalue of $\boldsymbol{J}$ [24]. In other words, as long as it is distinct, the largest eigenvalue of $\boldsymbol{J}$, for $it \to \infty$, dominates the convergence speed of this iterative algorithm. Based on this remark, authors in [25] proposed a step lengthening algorithm based on a multivariate form of the well-known Aitken acceleration approach to improve the convergence speed of the EM algorithm. As it is well-known for any iterative process where errors decrease proportionally through iterations, as it is the case for the EM algorithm, estimation errors between successive parameter estimates are proportionally linked such that:

$$\boldsymbol{\theta}^{(it)} - \boldsymbol{\theta}^{(it-1)} = \lambda\left(\boldsymbol{\theta}^{(it-1)} - \boldsymbol{\theta}^{(it-2)}\right) \tag{11}$$

For $\lambda < 1$, the above resembled fixed-point iterations are convergent [24]. Authors in [25] proposed to compute the step size λ at each iteration as a function of the current

and the previous two parameters estimates as follows:

$$\lambda^{(it)} = \frac{1}{Z}\sum_{i=1}^{Z}\frac{\left(\theta_i^{(it)} - \theta_i^{(it-1)}\right)}{\left(\theta_i^{(it-1)} - \theta_i^{(it-2)}\right)} \tag{12}$$

According to Eq. (11) and Eq. (12), authors proposed to use the following update rule of $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{(new)} = \boldsymbol{\theta}^{(it)} + \lambda^{(it)}\left(\boldsymbol{\theta}^{(it)} - \boldsymbol{\theta}^{(it-1)}\right) \tag{13}$$

It is noteworthy that $\boldsymbol{\theta}^{(new)}$ will replace $\boldsymbol{\theta}^{(it)}$ if the former increases the log-likelihood function being maximized [25]. This acceleration approach, when applied to accelerate the EM algorithm, is called the λ-EM method hereafter.

2.4. The Anti-Annealing based EM approach

The Anti-Annealing based EM (AAEM) approach is essentially inspired from the Annealing EM (AEM) one. As discussed previously, AEM is proposed in [19] as an efficient way to avoid local maxima during the optimization of the log-likelihood function for the EM algorithm. Recall that the key idea underlying the AEM approach is the parametrization of the posterior probability distribution by a temperature-related parameter, denoted here by β controlling the annealing process. Indeed, the Annealing scheme tracks the optimum of the log-likelihood function from high temperature wherein the log-likelihood is smoothed (*i.e.* it has one global optimum) to low temperature wherein the shape of the log-likelihood gradually approaches the one of the original log-likelihood. In this way, one guarantees a good initial guess through successive temperature parameters. In other words, the AEM algorithm modifies the posterior probabilities with the temperature-related parameter β in the E-step of the EM algorithm as follows:

$$h_k^{(it)}(n) = \frac{\left(\alpha_k^{(it)} p(\boldsymbol{x}_n|\boldsymbol{\mu}_k^{(it)}, \boldsymbol{\Sigma}_k^{(it)})\right)^{\beta}}{\sum_{r=1}^{K}\left(\alpha_r^{(it)} p(\boldsymbol{x}_n|\boldsymbol{\mu}_r^{(it)}, \boldsymbol{\Sigma}_r^{(it)})\right)^{\beta}} \tag{14}$$

In the M-step, the local vector of parameters $\boldsymbol{\theta}_k^{(it+1)}$ is updated using this posterior value as shown in Eq. (7). Typically, the AEM algorithm starts at $\beta_{\min} \simeq 0$ and

slowly increases towards one in such a way the initial guess of the vector $\boldsymbol{\theta}$, for a given β, in the EM algorithm is equal to its estimate computed under the previous β. Authors in [12] proposed a variant of the AEM approach called the AAEM method that considerably improves the convergence speed of the AEM and consequently the EM algorithm. Contrary to the AEM method wherein the temperature-related parameter β varies from very small value upwards to one, the AAEM algorithm applies a hybrid schedule, where it starts with $\beta_{\min} < 1$, then the parameter slowly increases upwards to $\beta_{\max} > 1$ and finally it is decreased downwards to $\beta = 1$. It is worth noting that the temperature-related parameter should be slow enough while dealing with complicated data with a large number of clusters [12].

Since the AAEM algorithm significantly outperforms the AEM one [19], only the AAEM algorithm is considered hereafter. Since both the gradient and the projection matrices, in the ECG algorithm, are basically computed using the posterior probability density as shown previously, applying the AA to ECG is straightforward and gives rise to the AAECG approach.

3. The proposed ELS scheme

As it is well-known, in case of GMM with unbalanced mixing coefficients (resp. overlapped clusters), the EM algorithm suffers from super linear convergence cycles, called "swamps" wherein the algorithm spends, for a given direction, a high number of iterations to get the final solution. To cope with this situation and inspired from the works in [21] and [26], an Exact Line Search (ELS) scheme is employed, giving rise to the ELS-EM algorithm. The latter is based on a linear interpolation of the unknown parameter ${\boldsymbol{\theta}_k}^{(new)}$ as follows:

$$ {\boldsymbol{\theta}_k}^{(new)} = {\boldsymbol{\theta}_k}^{(it-1)} + \operatorname{diag}\left(\boldsymbol{\rho}_k^{(it)}\right) \boldsymbol{G}_{\boldsymbol{\theta}_k}^{(it)} \tag{15} $$

where $\boldsymbol{\theta}_k^{(it-1)}$ denotes the estimation of $\boldsymbol{\theta}_k$ at the $(it-1)$-th iteration, $\boldsymbol{\rho}_k^{(it)} = \left[\rho_{\alpha_k^{(it)}}, \rho_{\boldsymbol{\mu}_k^{(it)}}, \rho_{\boldsymbol{\Sigma}_{\boldsymbol{k}}^{(\boldsymbol{it})}}\right]^{\mathsf{T}}$ stands for the vector of relaxation factors (step sizes) associated to GMM mixing coefficients, α_k, cluster means, μ_k, and covariance matrices, Σ_k, computed at the it-th iteration, respectively. $\boldsymbol{G}_{\boldsymbol{\theta}_k}^{(it)} = {\boldsymbol{\theta}_k}^{(it)} - {\boldsymbol{\theta}_k}^{(it-1)}$ denotes the given

search direction at the current iteration. Since $\boldsymbol{\theta}_k = [\alpha_k, \boldsymbol{\mu}_k, \mathrm{vec}\left(\boldsymbol{\Sigma}_k\right)]^{\mathsf{T}}$, we can write:

$$\begin{aligned}
\alpha_k^{(new)} &= \alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)} \\
\boldsymbol{\mu}_k^{(new)} &= \boldsymbol{\mu}_k^{(it-1)} + \rho_{\boldsymbol{\mu}_k^{(it)}} \boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)} \\
\boldsymbol{\Sigma}_k^{(new)} &= \boldsymbol{\Sigma}_k^{(it-1)} + \rho_{\boldsymbol{\Sigma}_k^{(it)}} \boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}
\end{aligned} \tag{16}$$

where $G_{\alpha_k}^{(it)} = \alpha_k^{(it)} - \alpha_k^{(it-1)}$, $\boldsymbol{G}_{\boldsymbol{\mu_k}}^{(it)} = \boldsymbol{\mu}_k^{(it)} - \boldsymbol{\mu}_k^{(it-1)}$ and $\boldsymbol{G}_{\boldsymbol{\Sigma_k}}^{(it)} = \boldsymbol{\Sigma}_k^{(it)} - \boldsymbol{\Sigma}_k^{(it-1)}$. Contrary to $\boldsymbol{\Sigma}_k^{(it)}$, $\boldsymbol{\Sigma}_k^{(new)}$ is not guaranteed to be positive semi-definite. Consequently, the semi-positive definiteness property of $\boldsymbol{\Sigma}_k^{(new)}$ should be verified at each iteration. If this property is violated, then $\boldsymbol{\Sigma}_k^{(new)}$ is set to $\boldsymbol{\Sigma}_k^{(it)}$ (in this situation, no further improvement can be expected for the covariance at the current iteration). The ELS scheme consists in exactly computing the step size vector, $\boldsymbol{\rho}_k^{(it)}, \forall k \in \{1, ..., K\}$ in an algebraic manner. This is by looking for the optimal step size $\boldsymbol{\rho}_k^{(it)}$ maximizing the expectation of the complete log-likelihood function, $Q\left(\boldsymbol{\theta}^{(new)} \left| \boldsymbol{\theta}^{(it)} \right.\right) = E\left[\sum_{n=1}^{N} \log\left(\alpha_k{}^{(new)} p\left(\boldsymbol{x}_n \left| \boldsymbol{\mu}_k^{(new)}, \boldsymbol{\Sigma}_k^{(new)} \right.\right)\right)\right]_{h_k^{(it)}(n)}$, under the posterior probability distribution, $h_k^{(it)}(n)$, such that:

$$\begin{aligned}
&\arg\max_{\boldsymbol{\rho}_k^{(it)}} \left\{ Q\left(\boldsymbol{\theta}^{(new)} \left| \boldsymbol{\theta}^{(it)} \right.\right) \right\} \\
&= \arg\max_{\boldsymbol{\rho}_k^{(it)}} \left\{ \sum_{n=1}^{N} \sum_{k=1}^{K} \left\{ \log\left(\alpha_k^{(new)} p(\boldsymbol{x}_n | \boldsymbol{\mu}_k^{(new)}, \boldsymbol{\Sigma}_k^{(new)}) \right) \right\} h_k^{(it)}(n) \right\} \\
&= \arg\max_{\boldsymbol{\rho}_k^{(it)}} \left\{ \sum_{n=1}^{N} \sum_{k=1}^{K} \left\{ \begin{array}{l}
\log\left(\alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)}\right) - \frac{D}{2}\log 2\pi \\
-\frac{1}{2}\log\det\left(\boldsymbol{\Sigma}_k^{(it-1)} + \rho_{\boldsymbol{\Sigma}_k^{(it)}} \boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right) \\
-\frac{1}{2}\left(\boldsymbol{x}_n - \left(\boldsymbol{\mu}_k^{(it-1)} + \rho_{\boldsymbol{\mu}_k^{(it)}} \boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right)^{\mathsf{T}} \\
\times\left(\boldsymbol{\Sigma}_k^{(it-1)} + \rho_{\boldsymbol{\Sigma}_k^{(it)}} \boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right)^{-1} \\
\times\left(\boldsymbol{x}_n - \left(\boldsymbol{\mu}_k^{(it-1)} + \rho_{\boldsymbol{\mu}_k^{(it)}} \boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right)
\end{array} \right\} h_k^{(it)}(n) \right\} \\
&s.t. \sum_{k=1}^{K} (\alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)}) = 1, \alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)} \geq 0.
\end{aligned} \tag{17}$$

The above optimization problem with respect to $\boldsymbol{\rho}_k^{(it)} = \left[\rho_{\alpha_k^{(it)}}, \rho_{\boldsymbol{\mu}_k^{(it)}}, \rho_{\boldsymbol{\Sigma}_k^{(it)}}\right]^{\mathsf{T}}$ with $\rho_{\alpha_k^{(it)}} \neq \rho_{\boldsymbol{\mu}_k^{(it)}} \neq \rho_{\boldsymbol{\Sigma}_k^{(it)}}$ is the optimal way to proceed. However, computing $\rho_{\boldsymbol{\mu}_k^{(it)}}$ and $\rho_{\boldsymbol{\Sigma}_k^{(it)}}$ ($\rho_{\boldsymbol{\mu}_k^{(it)}} \neq \rho_{\boldsymbol{\Sigma}_k^{(it)}}$) requires to solve a system of equations in $\rho_{\boldsymbol{\mu}_k^{(it)}}$ and $\rho_{\boldsymbol{\Sigma}_k^{(it)}}$ at each iteration which is relatively of high numerical complexity. To alleviate this issue,

an alternative suboptimal but feasible solution $\rho_{\boldsymbol{\mu}_k^{(it)}} = \rho_{\boldsymbol{\Sigma}_k^{(it)}} = \rho^{(it)}, \forall k \in \{1, ..., K\}$ is to be considered instead. Then, Eq. (17) becomes:

$$\begin{aligned}
&\arg\max_{\boldsymbol{\rho}_k^{(it)}} \left\{ Q\left(\boldsymbol{\theta}^{(new)} \middle| \boldsymbol{\theta}^{(it)}\right)\right\} \\
&= \arg \max_{\rho_{\alpha_k^{(it)}},\, \rho^{(it)}} \left\{ \sum_{n=1}^{N} \sum_{k=1}^{K} \left\{ \begin{array}{l} \log\left(\alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)}\right) - \frac{D}{2}\log 2\pi \\ -\frac{1}{2}\log\det\left(\boldsymbol{\Sigma}_k^{(it-1)} + \rho^{(it)} \boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right) \\ -\frac{1}{2}\left(\boldsymbol{x}_n - \left(\boldsymbol{\mu}_k^{(it-1)} + \rho^{(it)} \boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right)^{\mathsf{T}} \\ \times \left(\boldsymbol{\Sigma}_k^{(it-1)} + \rho^{(it)} \boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right)^{-1} \\ \times\left(\boldsymbol{x}_n - \left(\boldsymbol{\mu}_k^{(it-1)} + \rho^{(it)} \boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right) \end{array} \right\} h_k^{(it)}(n) \right\} \\
&s.t. \sum_{k=1}^{K} \alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)} = 1, \alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)} \geq 0.
\end{aligned} \tag{18}$$

The above optimization problem can be solved by alternating, at each iteration, between the following two optimization sub-problems:

$$\begin{aligned}
&\boldsymbol{P1}: \arg\max_{\rho_{\alpha_k^{(it)}}} \left\{ Q\left(\theta^{(new)} \middle| \theta^{(it)}\right)\right\} \\
&= \arg\max_{\rho_{\alpha_k^{(it)}}} \left\{ \sum_{n=1}^{N} \sum_{k=1}^{K} \left\{ \log\left(\alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)}\right)\right\} \times h_k^{(it)}(n) \right\} \\
&s.t. \sum_{k=1}^{K} \left(\alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)}\right) = 1,\ \alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)} \geq 0
\end{aligned} \tag{19}$$

and:

$$\begin{aligned}
&\boldsymbol{P2}: \arg\max_{\rho^{(it)}} \left\{ Q\left(\theta^{(new)} \middle| \theta^{(it)}\right)\right\} \\
&= \arg\max_{\rho^{(it)}} \left\{ \sum_{n=1}^{N} \sum_{k=1}^{K} \left\{ \begin{array}{l} -\frac{1}{2}\log\det\left(\boldsymbol{\Sigma}_k^{(it-1)} + \rho^{(it)} \boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right) \\ -\frac{1}{2}\left(\boldsymbol{x}_n - \left(\boldsymbol{\mu}_k^{(it-1)} + \rho^{(it)} \boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right)^{\mathsf{T}} \\ \times \left(\boldsymbol{\Sigma}_k^{(it-1)} + \rho^{(it)} \boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right)^{-1} \\ \times\left(\boldsymbol{x}_n - \left(\boldsymbol{\mu}_k^{(it-1)} + \rho^{(it)} \boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right) \end{array} \right\} h_k^{(it)}(n) \right\}
\end{aligned} \tag{20}$$

The constrained optimization problem $\boldsymbol{P1}$ is solved by maximizing the Lagrangian function associated to $\boldsymbol{P1}$ and given by:

$$\begin{aligned} L\left(\rho_{\alpha_k^{(it)}}, \xi\right) &= \sum_{n=1}^{N}\sum_{k=1}^{K}\left\{\log\left(\alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)}\right)\right\} h_k^{(it)}(n) \\ &+ \xi\left\{\sum_{k=1}^{K}(\alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)}) - 1\right\} \end{aligned} \tag{21}$$

where ξ stands for Lagrange multiplier. The solution of maximizing the above equation is given by:

$$\rho_{\alpha_k^{(it)}} = \left(\sum_{k=1}^{N}\frac{h_k^{(it)}(n)}{N} - \alpha_k^{(it-1)}\right)/G_{\alpha_k}^{(it)} \tag{22}$$

Details regarding the maximization of Eq. (21) are given in **Appendix A**. To solve $\boldsymbol{P2}$, the optimal relaxation factor is easily performed by setting $\rho^{(it)} = \rho_{\boldsymbol{\mu}_k^{(it)}} = \rho_{\boldsymbol{\Sigma}_k^{(it)}}$ and rooting the following second order polynomial in $\rho^{(it)}$:

$$y_2{}^{(it)}\left(\rho^{(it)}\right)^2 + y_1^{(it)}\rho^{(it)} + y_0^{(it)} = 0 \tag{23}$$

where the coefficients $y_2^{(it)}$, $y_1^{(it)}$ and $y_0^{(it)}$ are given in **Appendix B**.

It is noteworthy that the ELS scheme is inserted after the E-step once the computed $\boldsymbol{\theta}^{(new)}$ in Eq. (16) guarantees an increased log-likelihood function compared to its value at $\boldsymbol{\theta}^{(it)}$. Note that since both AAEM and AAECG algorithms are based on the computation of the posterior probability density of the latent variables given the observed data and the current estimate of the vector of parameters $\boldsymbol{\theta}^{(it)}$, their convergence speed can be also improved using the ELS scheme, giving rise to ELS-AAEM and ELS-AAECG methods. The performance of the two latter variants will be also considered in our numerical simulations. **Algorithm 1** provides a pseudo-code of the proposed ELS-EM, ELS-AAEM and ELS-AAECG algorithms. Regarding the numerical complexity per iteration of the considered methods in this section, it is given in Table 1 and expressed in numerical flop. Note that a numerical flop is defined as a multiplication followed by addition. But, since, in practice, the number of multiplications is often larger than the number of additions, only the number of multiplications is reported in Table 1.

Algorithm 1 ELS-EM, ELS-AAEM, ELS-AAECG algorithms

Repeat until the convergence or a maximum number of iterations is reached.

E-step: calculate $h_k^{(it)}(n)$ from $\boldsymbol{\theta}^{(it)}$ in Eq. (6) for ELS-EM, or in Eq. (14) for ELS-AAEM and ELS-AAECG.

ELS-step:

(a) Compute $\rho_{\alpha^{(it)}}$ from Eq. (22) and compute $\rho^{(it)}$ by rooting Eq. (23);

(b) Find $\boldsymbol{\theta}_k^{(new)}, \forall 1 \leq k \leq K$ from Eq. (16) and then set $\boldsymbol{\theta} = [\boldsymbol{\theta}_1{}^{\intercal}, \cdots, \boldsymbol{\theta}_K{}^{\intercal}]^{\intercal}$;

(c) **if** $L(\boldsymbol{\theta}^{(new)}) > L(\boldsymbol{\theta}^{(it)})$ **then**

Set $h_k^{(it)}(n) = h_k^{(new)}(n)$ computed from Eq. (6) for ELS-EM or from Eq. (14) for ELS-AAEM and ELS-AAECG

else

Go to the **M/CG-step.**

end

M/CG-step: update the model parameter vector $\boldsymbol{\theta}^{(it+1)}$ using Eq. (7) for ELS-EM and ELS-AAEM or using Eq. (9) for ELS-AAECG.

End

Table 1. The computational complexity is calculated per iteration for different methods, $T(E) = O(NK(2D^2 + 2))$, $T(M) = O(NK(2D^2 + D + 1))$, $T(CG) = O(NK(4D^2 + 1))$, $T(ELS) = O(NK(4D^2 + 2D + 2))$, where N is the number of data points, D is the dimension of each data point and K is the number of Gaussian distributions in GMM.

Method	Numerical Complexity
EM	$T(E) + T(M) = O(NK(4D^2 + D + 3))$
ECG	$T(E) + T(CG) = O(NK(6D^2 + 3))$
λ-EM	$2T(E) + T(M) = O(NK(6D^2 + D + 3))$
ELS-EM	$T(E) + T(ELS) + T(M) = O(NK(8D^2 + 2D + 5))$
AAEM	$T(E) + T(M) = O(NK(4D^2 + D + 3))$
AAECG	$T(E) + T(CG) = O(NK(6D^2 + 3))$
ELS-AAEM	$T(E) + T(ELS) + T(M) = O(NK(8D^2 + 2D + 5))$
ELS-AAECG	$T(E) + T(ELS) + T(CG) = O(NK(10D^2 + 3D + 5))$

4. Results

This section is devoted to show to what extent the proposed ELS scheme can improve the convergence speed of the conventional EM algorithm and also that of its variants, *i.e.* the AAEM and the AAECG methods. Besides, the proposed ELS scheme (when applied) is compared to two very efficient schemes accelerating the EM algorithm, namely the ECG method [15] and the λ-EM [25]. This comparative study is first conducted in the context of GMMs with unbalanced mixing coefficients

(two- and four-component GMMs are considered), poorly-separated clusters (only a two-component GMM is considered). Then, the efficiency of the proposed ELS scheme is investigated in the context of handwritten digits '4' and '8' classification using the MNIST dataset (available online http://yann.lecun.com/exdb/mnist).

The estimation quality of the GMM parameters is evaluated based on symmetric Kullback divergence between the true and the estimated GMM components as well as the average log-likelihood function, as follows [12]:

$$\begin{aligned} e^{(it)} &= \frac{1}{R}\sum_{r=1}^{R}\sum_{k=1}^{K} KL_S^{(r)}\left(\mathcal{N}\left(\boldsymbol{x}\left|\boldsymbol{\mu}_k^{(it)},\boldsymbol{\Sigma}_k^{(it)},\boldsymbol{\theta}_0^{(r)}\right.\right),\mathcal{N}\left(\boldsymbol{x}\left|\boldsymbol{\mu}_{\pi_k},\boldsymbol{\Sigma}_{\pi_k}\right.\right)\right) \\ \overline{L}^{(it)} &= \frac{1}{R}\sum_{r=1}^{R} L(\boldsymbol{\theta}^{(it)}|\boldsymbol{\theta}_0^{(r)}) \end{aligned} \tag{24}$$

where $e^{(it)}$ stands for the mean estimation error and $\overline{L}^{(it)}$ stands for the average log-likelihood at the it-th iteration, R denotes the number of random and independent initialization points $\boldsymbol{\theta}_0^{(r)}$, $\{\pi_k\}_{k=1}^{K}$ is the one-to-one mapping estimated by minimum weight bipartite graph matching [12], and $KL_S^{(r)}$ is the symmetric Kullback divergence when the r-th initial point is considered and defined as:

$$\begin{aligned} & KL_S^{(r)}\left(p^{(r)},q\right) = KL\left(p^{(r)},q\right) + KL\left(q,p^{(r)}\right),\ \forall 1\leq r\leq R \\ & \Rightarrow KL_S^{(r)}\left(\mathcal{N}\left(\boldsymbol{x}\left|\boldsymbol{\mu}_i^{(r)},\boldsymbol{\Sigma}_i^{(r)}\right.\right),\mathcal{N}\left(\boldsymbol{x}\left|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\right.\right)\right) \\ & = \frac{1}{2}\mathrm{Tr}\left(\left(\boldsymbol{\Sigma}_i^{(r)}\right)^{-1}\boldsymbol{\Sigma}_j + \boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_i^{(r)}\right) + \frac{1}{2}\left(\boldsymbol{\mu}_i^{(r)}-\boldsymbol{\mu}_j\right)^{\mathsf{T}}\left(\left(\boldsymbol{\Sigma}_i^{(r)}\right)^{-1}+\boldsymbol{\Sigma}_j^{-1}\right)\left(\boldsymbol{\mu}_i^{(r)}-\boldsymbol{\mu}_j\right) \\ & -D \end{aligned} \tag{25}$$

where $\mathrm{Tr}(.)$ is the trace of its matrix argument. Besides, considering the stop condition, the maximal number of iterations was set to 2000 for all the algorithms. As far as the value of the threshold τ in Eq. (8) was considered, it was set to 10^{-10} for all the methods except for the Anti-Annealing based ones where it was set to 10^{-6} as suggested in [12]. Indeed, the authors argued that the Anti-Annealing based approaches did not require a conservative tolerance since they were able to speed up convergence at later stages.

4.1. Initialization strategy

As indicated in [1, 19], the EM algorithm is highly sensitive to initialization. Indeed, too close initial guess can lead to slowness or to sub-estimation (*i.e.* trend to

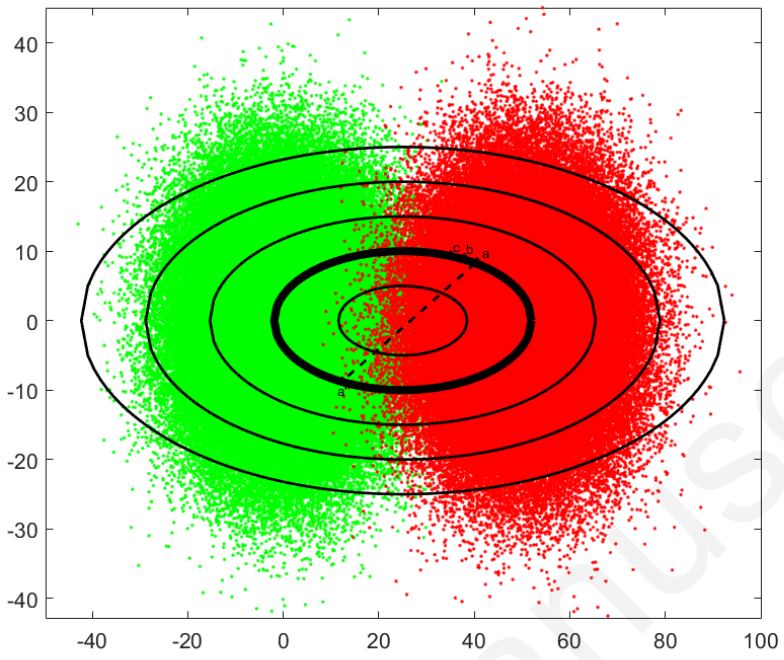

Fig. 1. Illustration of the proposed initialization strategy in the case of two-component GMM model with $N_1 = N_2 = 2 \times 10^5$. Ellipse associated to the data covariance matrix is shown in bold, the others being its dilated and contracted versions. Points $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ and $\boldsymbol{a}'$ are four possible initializations of the means $\boldsymbol{\mu}_1$ or $\boldsymbol{\mu}_2$. The impact of the initialization on the convergence of the EM algorithm is evaluated using three possible initializations couples of $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$: $(\boldsymbol{a}, \boldsymbol{b})$, $(\boldsymbol{a}, \boldsymbol{c})$ and $(\boldsymbol{a}, \boldsymbol{a}')$. Reported results show that the EM algorithm stops, respectively, after 3, 82 and 27 iterations with GMM estimation error of 12.19, 3.83×10^{-5} and 3.83×10^{-5}.

identify only one-component instead of K-component GMM model). For instance, let's consider the identification of a 2-D two-component GMM model shown in Fig. 1. For initialization, all mixing coefficients are set to $1/K$ ($K = 2$ in this example). Besides, initial values of the covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ can be chosen to be equal to the covariance matrix of the observed data points. Regarding the initial values of the means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, a wise selection strategy is to be used in order to avoid sub-estimation and slowness issues. To illustrate this fact, let $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ and $\boldsymbol{a}'$ be four different possible initial points for the two means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. The latter four points can be figured out on the same imaginary ellipse associated with the observed data covariance matrix, as shown in Fig. 1. When the chosen initial points for $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are too close (*i.e.* case of points $\boldsymbol{a}$ and $\boldsymbol{b}$), the EM algorithm suffers from sub-estimation as reported in our conducted

simulations. More precisely, the algorithm stops very early (only three iterations) with high estimation error $e = 12.19$. Now, this error is considerably decreased for relatively well separated initial points (*i.e.* cases of $(\boldsymbol{a}, \boldsymbol{c})$ and $(\boldsymbol{a}, \boldsymbol{a}')$). Indeed, while the EM algorithm requires, in the case of the $(\boldsymbol{a}, \boldsymbol{c})$, 82 iterations to converge with estimation error $e = 3.83 \times 10^{-5}$, it requires 27 iterations with estimation error $e = 3.83 \times 10^{-5}$ for the couple $(\boldsymbol{a}, \boldsymbol{a}')$. In order to span different well separated initial points for $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, the ellipse related to the computed observation covariance matrix is dilated three times with factors 1.5, 2 and 2.5, respectively and contracted once by a factor of 0.5. This leads to five center ellipses in the observation plan, as shown in Fig. 1. Next, 10 points are randomly selected in each ellipse. For each point on a given ellipse, its symmetrical point with respect to the center is taken. Consequently a set of 10 couples of possible well separated initial points for $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ is obtained. Finally, the behaviour of the considered algorithms was averaged over those 50 couples of possible initial points. The generalization of this strategy to the case of four-component GMM model considered in this study is, without a loss of generality, straightforward.

4.2. A two-component GMM

A two-component GMM is considered hereafter to evaluate the performance of the proposed ELS scheme when applied to the EM algorithm and its variants, the AAEM, the AAECG and the λ-EM approaches. Regarding the AA based approaches in this configuration, the temperature-related parameter β takes successively the following values $0, 8, 1.0, 1.2$ and 1.0 [12]. Three possible two-component GMM situations are investigated hereafter: (i) GMM with balanced and slightly overlapped components, (ii) GMM with unbalanced and slightly overlapped components and (iii) GMM with balanced and highly overlapped components.

4.2.1. Case of balanced and slightly overlapped components

The performance of the considered algorithms is evaluated here in the case of two-component GMM with components that are assumed to be balanced (*i.e.* $\alpha_1 = \alpha_2$) and slightly overlapped. This performance study is performed as a function of the size of observed data set with the assumption that the latter are equally divided between the

Table 2. Mean number of iterations $\pm$ standard deviation, mean error $\pm$ standard deviation computed at the mean number of iterations and mean elapsed CPU time per iteration $\pm$ standard deviation, over 50 randomly and independently chosen initial points, for the EM, ECG, λ-EM, AAEM, AAECG algorithms, the proposed ELS-EM, ELS-AAEM and ELS-AAECG algorithms in the case of two balanced-component GMM as a function of the number of data points with $N_2 = N_1$.

Method	Mean iteration$\pm$std			
	$N_1 = 2 \times 10^2$	$N_1 = 2 \times 10^3$	$N_1 = 2 \times 10^4$	$N_1 = 2 \times 10^5$
EM	53.9$\pm$39.2	77.7$\pm$116.8	87.1$\pm$158.5	123.0$\pm$297.6
ECG	92.7$\pm$36.8	111.9$\pm$148.8	119.6$\pm$191.4	182.1$\pm$342.1
λ-EM	35.8$\pm$24.1	50.9$\pm$72.3	56.7$\pm$97.8	78.3$\pm$184.7
ELS-EM	**24.8$\pm$9.4**	**30.0$\pm$35.5**	**33.9$\pm$51.3**	**46.6$\pm$111.5**
AAEM	72.0$\pm$20.8	30.3$\pm$25.4	27.0$\pm$4.7	27.6$\pm$3.0
AAECG	75.3$\pm$5.7	24.1$\pm$1.9	24.0$\pm$1.9	23.9$\pm$2.0
ELS-AAEM	**34.9$\pm$7.1**	17.2$\pm$1.4	17.3$\pm$1.3	17.3$\pm$0.9
ELS-AAECG	39.1$\pm$2.7	16.9$\pm$0.8	16.9$\pm$0.6	17.0$\pm$0.7

Method	Mean error$\pm$std			
	$N_1 = 2 \times 10^2$	$N_1 = 2 \times 10^3$	$N_1 = 2 \times 10^4$	$N_1 = 2 \times 10^5$
EM	2.8071$\pm$4.5388	2.5798$\pm$4.9074	2.5486$\pm$4.7803	2.4059$\pm$4.6275
ECG	0.7781$\pm$2.5774	2.5231$\pm$4.8344	2.5486$\pm$4.8656	2.6552$\pm$4.8460
λ-EM	2.6464$\pm$4.4556	2.5753$\pm$4.8995	2.4993$\pm$4.7792	2.3932$\pm$4.6095
ELS-EM	**0.7205$\pm$2.4813**	**2.2893$\pm$4.6325**	**2.2727$\pm$4.6057**	**2.0325$\pm$4.3945**
AAEM	0.2500$\pm$0.8664	9.6060$\pm$5.1515	10.1560$\pm$4.8070	11.4134$\pm$3.4001
AAECG	0.0458$\pm$0.0100	12.3095$\pm$0.0661	12.3731$\pm$0.0631	12.3979$\pm$0.0544
ELS-AAEM	**0.0428$\pm$0.0021**	9.6264$\pm$5.1618	10.1617$\pm$4.8095	11.4144$\pm$3.4005
ELS-AAECG	0.0429$\pm$0.0024	12.3281$\pm$0.0651	12.3892$\pm$0.0616	12.4153$\pm$0.0519

Method	Mean CPU time$\pm$std			
	$N_1 = 2 \times 10^2$	$N_1 = 2 \times 10^3$	$N_1 = 2 \times 10^4$	$N_1 = 2 \times 10^5$
EM	0.0008$\pm$0.0003	0.0018$\pm$0.0007	0.0108$\pm$0.0016	0.1654$\pm$0.0242
ECG	0.0017$\pm$0.0007	0.0025$\pm$0.0006	0.0123$\pm$0.0020	0.2014$\pm$0.0314
λ-EM	0.0018$\pm$0.0009	0.0029$\pm$0.0009	0.0160$\pm$0.0030	0.2506$\pm$0.0495
ELS-EM	0.0028$\pm$0.0013	0.0043$\pm$0.0014	0.0229$\pm$0.0045	0.3765$\pm$0.0714
AAEM	0.0008$\pm$0.0003	0.0016$\pm$0.0005	0.0099$\pm$0.0015	0.1533$\pm$0.0263
AAECG	0.0016$\pm$0.0006	0.0024$\pm$0.0008	0.0114$\pm$0.0020	0.1862$\pm$0.0336
ELS-AAEM	0.0023$\pm$0.0009	0.0030$\pm$0.0011	0.0169$\pm$0.0034	0.2706$\pm$0.0559
ELS-AAECG	0.0031$\pm$0.0014	0.0041$\pm$0.0014	0.0189$\pm$0.0032	0.3118$\pm$0.0513

two GMM components (*i.e.* $N_1 = N_2$). Therefore, a mixture of two 2-D Gaussian distributions $\mathcal{N}_i(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), i \in \{1, 2\}$ is generated with $\boldsymbol{\mu}_1 = [0, 0]^\mathsf{T}, \boldsymbol{\mu}_2 = [50, 0]^\mathsf{T}$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = [10^2 \quad 0; 0 \quad 10^2]$. Note that coefficients $\alpha_i, i \in \{1, 2\}$ are defined here as $\alpha_i = \frac{N_i}{N_1+N_2}$ such that the constraint $\alpha_1 + \alpha_2 = 1$ (see Eq. (3)) is respected. Under the assumption $N_1 = N_2$, we then have $\alpha_1 = \alpha_2 = 0.5$. Reported results in terms of (i) the mean number of iterations required by the algorithm to reach the final solution, (ii) the mean estimation error and (iii) the mean elapsed CPU time, are given in Table 2 for different sizes of data points (*i.e.* $N_1 = N_2 = 2 \times 10^2, 2 \times 10^3, 2 \times 10^4$ and 2×10^5).

According to the latter table, an increase in the convergence speed (expressed here in terms of the mean number of iterations required by the algorithm to reach the final solution) of the EM, the AAEM and the suggested AAECG algorithm is to be noticed when the proposed ELS scheme is employed, whatever the size of observed data points is. Besides, all considered algorithms except the AA-based algorithms require higher number of iterations to reach its respective final solutions as the number of data points increases. As shown in Table 2, employing the ELS scheme is still advantageous especially in difficult situations where the algorithm under study suffers from convergence issue. For instance, for $N_1 = N_2 = 2 \times 10^5$, the ELS scheme reduces dramatically (around 60%) the number of iterations required by the EM algorithm to reach its final solution. Besides, regarding the AA-based methods, the proposed ELS-AAEM and ELS-AAECG algorithms, globally outperform the AAEM and AAECG approaches when $N_1 = N_2 = 2 \times 10^2$. The aforementioned results are also confirmed in terms of the mean error taken at the computed mean number of iterations, as given in Table 2. Indeed, the proposed ELS-AAEM and ELS-ECG algorithms show lower mean error values compared to the AAEM and the AAECG ones for relatively small of observations size (*i.e.* $N_1 = N_2 = 2 \times 10^2$). However, a clear lack of convergence of the AA-based algorithms is to be noticed for higher data size as confirmed by the high error values depicted in Table 2. We note also from the latter table that the proposed ELS-EM outperforms the ECG, the λ-EM and the EM algorithms whatever the size of observed data points is.

Above mentioned results can be further confirmed as depicted in Figs. 2 and 3 where the former concerns the case of relatively small number of data points (*i.e.* $N_1 = N_2 = 2 \times 10^2$) and the latter is for relatively high observations size (*i.e.* $N_1 = N_2 = 2 \times 10^5$). Fig. 2 (b,c) and Fig. 3 (b,c) show that the ELS scheme, when employed, helps considerably in reducing the number of iterations and in providing better estimation quality as reflected by the values of the mean error corresponding to the obtained mean number of iterations. In addition to the superiority of the ELS-EM over the conventional EM method, higher performance of the former compared to the ECG and to the λ-EM methods are also shown in Fig. 2 (b) and Fig. 3 (b). Indeed, the proposed ELS-EM algorithm reaches faster its maximum log-likelihood

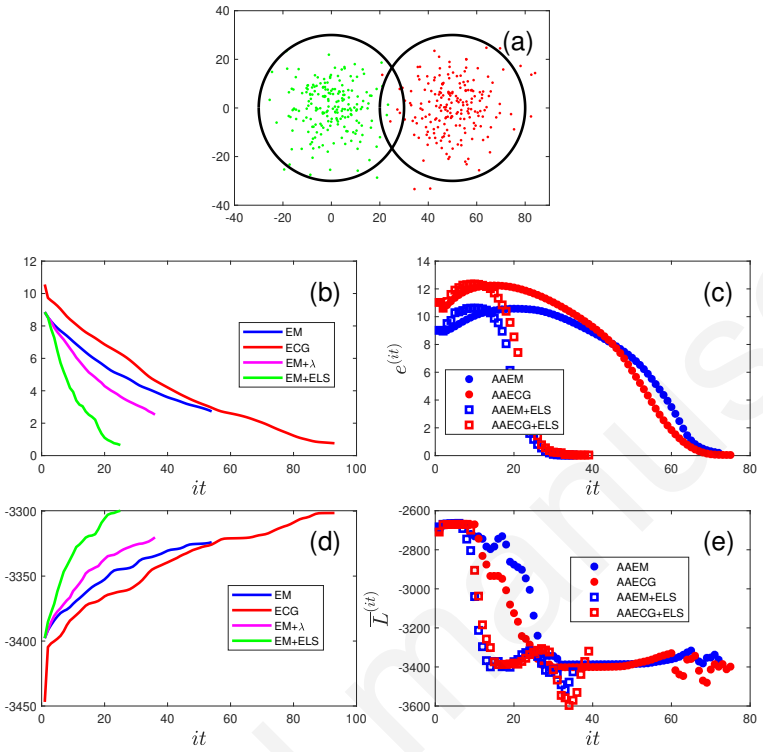

Fig. 2. Performance of the EM, ECG, λ-EM, AAEM, AAECG algorithms compared to the proposed ELS-EM, ELS-AAEM and ELS-AAECG ones in the case of $N_1 = 2 \times 10^2$. (a) A two-component GMM with small overlap, (b, c) mean estimation error, (d, e) averaged log-likelihood.

solution compared to the other considered methods in this study as shown in Fig. 2 (d) and Fig. 3 (d). As far as the performance of AA-based algorithms is concerned, a smaller number of iterations is generally required for the proposed ELS-AAEM and ELS-AAECG algorithms compared to the AAEM and AAECG ones to get the final solution. However, this fact holds true only in the case of relatively small data points (*i.e.* $N_1 = N_2 = 2 \times 10^2$), as shown in Fig. 2 (c). In fact, as mentioned previously, a lack of convergence of the AA-based methods is to be noticed for higher number of data points (*i.e.* $2 \times 10^3, 2 \times 10^4$ and 2×10^5). For the lack of space, only results for ($N_1 = N_2 = 2 \times 10^5$) are reported and shown in Fig. 3 (c). This lack of convergence is due to the fact that AA-based methods tend probably to underestimate the GMM

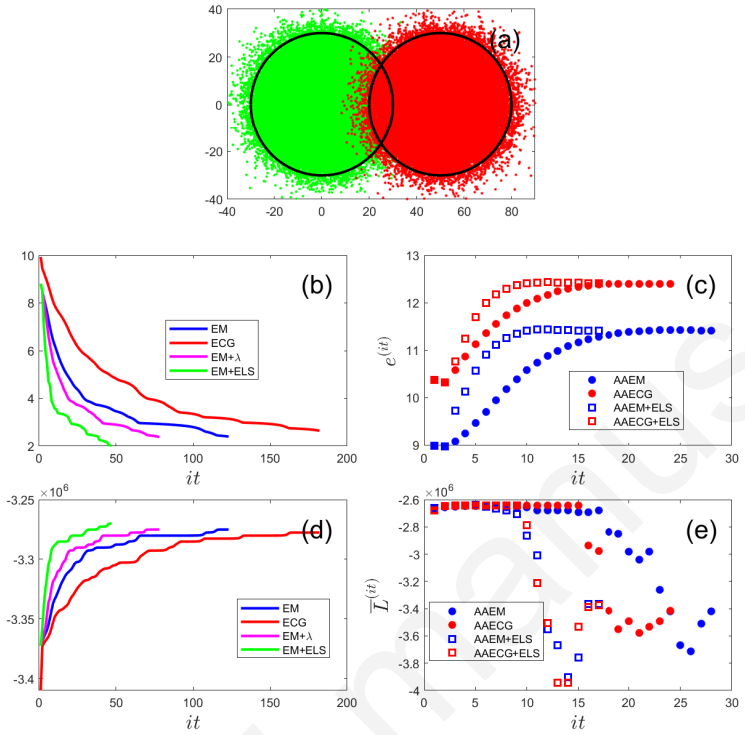

Fig. 3. Performance of the EM, ECG, λ-EM, AAEM, AAECG algorithms compared to the proposed ELS-EM, ELS-AAEM and ELS-AAECG ones in the case of $N_1 = 2 \times 10^5$. (a) A two-component GMM with small overlap, (b, c) mean estimation error, (d, e) averaged log-likelihood.

parameters and tend to identify only one Gaussian component instead of two. This is regardless the non-monotonic behaviour of the log-likelihood maximization using the AA-based approaches. Indeed, this non-monotonic behaviour is probably induced by the permanent change of $h_k^{(it)}(n)$ in Eq. (14) with the temperature-related parameter, β [12]. Regarding the mean CPU time per iteration, Table 2 shows, as expected, higher values for the ELS-EM, the ELS-AAEM and the ELS-AAECG compared to the EM, the AAEM and the AAECG, respectively. This is mainly due to the fact that the use of the supplementary exact line search steps has been employed in the latter methods (see **Algorithm 1**). However, this computation time is not crippling since less iterations and lower estimation error are expected when the ELS scheme is employed.

4.2.2. Case of unbalanced and slightly overlapped components

Hereafter, the performance of the considered algorithms is evaluated here in the case of unbalanced but slightly overlapped two Gaussian components. To this end, $N_1 = 2 \times 10^5$ data points were simulated from the first Gaussian component with mean $\boldsymbol{\mu}_1 = [0, 0]^\top$ and covariance matrix $\boldsymbol{\Sigma}_1 = [10^2 \quad 0; 0 \quad 10^2]$. Regarding the second 2-D Gaussian component, the mean vector is set to $\boldsymbol{\mu}_2 = [50, 0]^\top$ and the covariance matrix $\boldsymbol{\Sigma}_2$ is chosen such that $\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_1$. Furthermore, three values of N_2 (*e.g.* $N_2 \in \{2 \times 10^2, 2 \times 10^3, 2 \times 10^4\}$) were investigated.

Table 3. Mean number of iterations $\pm$ standard deviation, mean error $\pm$ standard deviation computed at the mean number of iterations and mean elapsed CPU time per iteration $\pm$ standard deviation, over 50 random and independent initial points, for the EM, the ECG, the λ-EM, the AAEM, the AAECG algorithms and the proposed ELS-EM, ELS-AAEM and ELS-AAECG ones in the case of two unbalanced-component GMM as a function of the number of data points, N_2 with $N_1 = 2 \times 10^5$.

Method	Mean iteration$\pm$std		
	$N_2 = 2 \times 10^2$	$N_2 = 2 \times 10^3$	$N_2 = 2 \times 10^4$
EM	905.6$\pm$19.4	125.6$\pm$2.0	42.3$\pm$1.5
ECG	504.2$\pm$24.3	89.7$\pm$**1.7**	36.6$\pm$**1.1**
λ-EM	560.2$\pm$12.7	78.1$\pm$2.7	28.4$\pm$1.3
ELS-EM	**442.4**$\pm$**7.2**	**71.4**$\pm$3.3	**25.6**$\pm$2.4
AAEM	78.6$\pm$2.8	131.5$\pm$1.9	54.6$\pm$1.4
AAECG	59.0$\pm$2.6	97.5$\pm$1.2	47.6$\pm$1.1
ELS-AAEM	49.6$\pm$3.9	74.4$\pm$2.0	36.5$\pm$2.0
ELS-AAECG	**39.3**$\pm$**2.0**	**66.5**$\pm$**1.6**	**36.3**$\pm$**2.1**

Method	Mean error$\pm$std		
	$N_2 = 2 \times 10^2$	$N_2 = 2 \times 10^3$	$N_2 = 2 \times 10^4$
EM	0.0718$\pm$0.2394	0.0029$\pm$0.0000	**0.0002$\pm$0.0000**
ECG	0.3518$\pm$0.9339	0.0029$\pm$0.0000	**0.0002$\pm$0.0000**
λ-EM	0.0865$\pm$0.2929	**0.0028$\pm$0.0001**	**0.0002$\pm$0.0000**
ELS-EM	**0.0061$\pm$0.0087**	**0.0028$\pm$0.0001**	**0.0002$\pm$0.0000**
AAEM	0.0554$\pm$0.2237	0.0079$\pm$0.0038	0.0003$\pm$0.0002
AAECG	0.3014$\pm$1.1736	0.0065$\pm$0.0023	0.0003$\pm$0.0002
ELS-AAEM	0.0942$\pm$0.3365	0.0097$\pm$0.0049	0.0005$\pm$0.0003
ELS-AAECG	0.4267$\pm$2.6767	0.0070$\pm$0.0036	0.0005$\pm$0.0003

Method	Mean CPU time$\pm$std		
	$N_2 = 2 \times 10^2$	$N_2 = 2 \times 10^3$	$N_2 = 2 \times 10^4$
EM	0.0646$\pm$0.0044	0.0671$\pm$0.0051	0.0705$\pm$0.0040
ECG	0.0815$\pm$0.0051	0.0833$\pm$0.0050	0.0887$\pm$0.0050
λ-EM	0.1072$\pm$0.0055	0.1097$\pm$0.0082	0.1127$\pm$0.0059
ELS-EM	0.1729$\pm$0.0097	0.1657$\pm$0.0091	0.1710$\pm$0.0098
AAEM	0.0635$\pm$0.0049	0.0671$\pm$0.0063	0.0700$\pm$0.0034
AAECG	0.0779$\pm$0.0062	0.0834$\pm$0.0058	0.0872$\pm$0.0055
ELS-AAEM	0.1423$\pm$0.0106	0.1525$\pm$0.0098	0.1480$\pm$0.0080
ELS-AAECG	0.1530$\pm$0.0122	0.1671$\pm$0.0123	0.1651$\pm$0.0110

Table 3 shows clearly an increase in the convergence speed of the EM, the AAEM and the suggested AAECG algorithms when the proposed ELS scheme is employed, whatever the value of N_2. In the most difficult situation, *i.e.* when $N_2 = 2 \times 10^2$, an increase in the convergence speed of more than $50\%, 35\%$ and 30% for the ELS-EM, the ELS-AAEM and the ELS-AAECG is reported in Table 3 compared to that of their conventional counterparts, namely the EM, the AAEM and the AAECG approaches, respectively. In addition, Table 3 shows the superiority of the proposed ELS-EM algorithm over the λ-EM one [25] which can be also seen as a line-search based approach. For example, when $N_2 = 2 \times 10^2$, the ELS-EM outperforms the λ-EM with an increase of 21% in the convergence speed. Regarding the estimation quality (*i.e.* the mean error computed at the mean number of iteration), the proposed ELS-EM algorithm shows generally the best performance compared with the other considered algorithms as shown in Table 3. Besides, the proposed ELS-AAECG and ELS-EM algorithms, globally outperform the other approaches in the case of unbalanced Gaussian components and for all considered N_2 values. As far as the mean CPU time per iteration is concerned, we note again that higher values of the latter are expected when the ELS scheme is employed. This fact is consistent with the numerical complexity per iteration given in Table 1 for all considered methods in this paper.

The aforementioned results are assessed in terms of the mean error, $e^{(it)}$ (as depicted in Fig. 4 (b, c)) and the averaged log-likelihood, $\overline{L}^{(it)}$ (as depicted in Fig. 4 (d, e)). In the case of highly unbalanced two-component GMM as considered here (Fig. 4 (a)), *e.g.* when $N_2 = 2 \times 10^2$, it is obvious that the ELS scheme, when employed, helps considerably in reducing the number of iterations that the EM, the AAEM and the AAECG are spending when stacking in swamps produced in such a situation. Thus, the proposed ELS-EM, ELS-AAEM and ELS-AAECG algorithms get their maximum log-likelihood solution in a relatively smaller number of iterations compared to the EM, ECG, λ-EM, AAEM and AAECG approaches, as depicted in Fig. 4 (d, e).

4.2.3. Case of balanced and overlapped Gaussian components

The convergence speed of the ELS-EM, ELS-AAEM, ELS-AAECG algorithms compared to the EM, AAEM, ECG, λ-EM and AAECG algorithms is evaluated as a

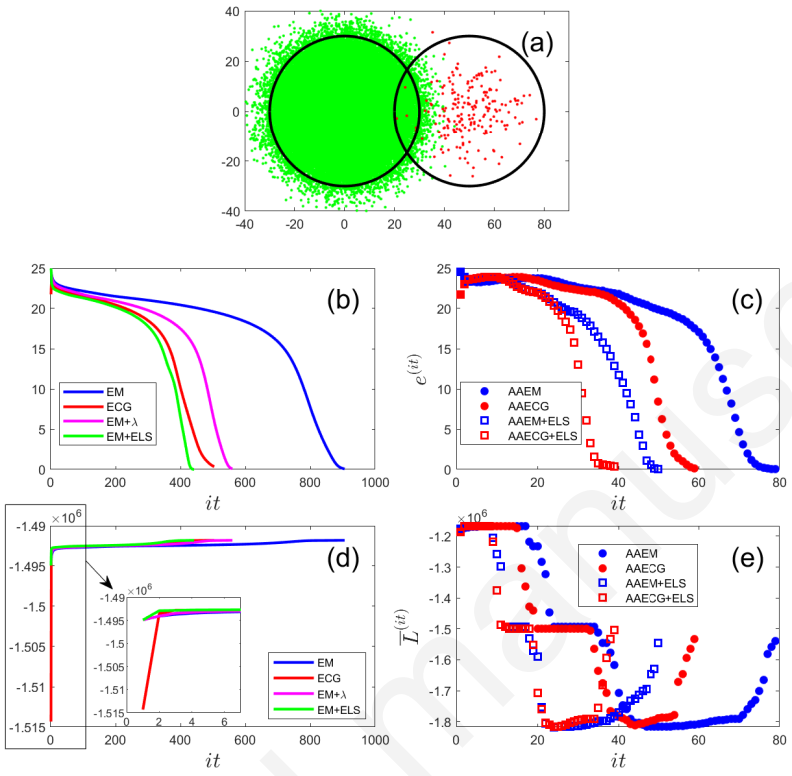

Fig. 4. Performance of the EM, ECG, λ-EM, AAEM, AAECG algorithms, the proposed ELS-EM, ELS-AAEM and ELS-AAECG algorithms in the case of two unbalanced-component GMM with $N_1 = 2 \times 10^5$, $N_2 = 2 \times 10^3$ and well separated clusters ($d = 50$). (a) A two-component GMM, (b, c) mean estimation error, (d, e) averaged log-likelihood.

function of the overlap between two balanced GMM components (*e.g.* $\alpha_1 = \alpha_2 = 0.5$). This overlap is expressed as the distance, denoted by d, between the two cluster centroids, *e.g.* $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. To this end, $N_1 = N_2 = 2 \times 10^5$ and $\boldsymbol{\mu}_1 = [0, 0]^\mathsf{T}$, $\boldsymbol{\mu}_2 = [d, 0]^\mathsf{T}$, with d varying from 10 (see Fig. 5 (a)) to 50 (see Fig. 3 (a)) by a step of 10. As far as the covariance matrices of the two Gaussian components are concerned, they are kept equal such that $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = [10^2 \quad 0; 0 \quad 10^2]$.

Table 4 confirms that the proposed ELS scheme, when applied, enhances the convergence speed of the EM algorithm and its variants towards the final solution. More precisely, for the most difficult case considered in this configuration, *e.g.* $d = 10$, the

Table 4. Mean number of iterations $\pm$ standard deviation, mean error $\pm$ standard deviation taken at the mean number of iterations and mean CPU time per iteration $\pm$ standard deviation, over 50 random and independent initial points, for the EM, ECG, λ-EM, AAEM, AAECG algorithms and the proposed ELS-EM, ELS-AAEM and ELS-AAECG algorithms in the case of overlapped but balanced (*i.e.* $N_1 = N_2 = 2 \times 10^5$) Gaussian components as a function of the distance, d, between the latter.

Method	Mean iteration$\pm$std				
	$d = 10$	$d = 20$	$d = 30$	$d = 40$	$d = 50$
EM	1061.9$\pm$498.0	392.3$\pm$368.5	278.3$\pm$543.5	251.9$\pm$505.5	123.0$\pm$297.6
ECG	1033.6$\pm$560.1	300.7$\pm$214.1	262.0$\pm$485.3	247.3$\pm$471.8	182.1$\pm$342.1
λ-EM	685.9$\pm$362.6	245.4$\pm$229.5	203.1$\pm$434.8	158.3$\pm$312.1	78.3$\pm$184.7
ELS-EM	**403.7$\pm$254.3**	**171.2$\pm$125.6**	**127.6$\pm$230.1**	**96.6$\pm$190.9**	**46.6$\pm$111.5**
AAEM	25.8$\pm$1.2	26.0$\pm$1.3	26.1$\pm$1.8	26.1$\pm$1.8	27.6$\pm$3.0
AAECG	23.0$\pm$1.4	23.0$\pm$1.3	23.0$\pm$1.2	23.0$\pm$1.3	23.9$\pm$2.0
ELS-AAEM	18.6$\pm$1.0	18.6$\pm$1.1	18.5$\pm$1.0	18.7$\pm$1.1	17.3$\pm$0.9
ELS-AAECG	18.6$\pm$0.5	18.6$\pm$0.5	18.6$\pm$0.6	18.7$\pm$0.6	17.0$\pm$0.7

Method	Mean error$\pm$std				
	$d = 10$	$d = 20$	$d = 30$	$d = 40$	$d = 50$
EM	0.0965$\pm$0.2070	0.2227$\pm$0.6105	0.6185$\pm$1.5501	1.3867$\pm$2.9957	2.4059$\pm$4.6275
ECG	0.1223$\pm$**0.2034**	0.4269$\pm$0.7373	0.7655$\pm$1.6585	1.7334$\pm$3.1738	2.6552$\pm$4.8460
λ-EM	0.0962$\pm$0.2064	0.2227$\pm$0.6105	0.6150$\pm$1.5416	1.3869$\pm$2.9960	2.3932$\pm$4.6095
ELS-EM	**0.0958**$\pm$0.2057	**0.2141$\pm$0.5879**	**0.6082$\pm$1.5250**	**1.2452$\pm$2.8835**	**2.0325$\pm$4.3945**
AAEM	0.4936$\pm$0.0026	1.9927$\pm$0.0043	4.4745$\pm$0.0090	7.9796$\pm$0.0149	11.4134$\pm$3.4001
AAECG	0.4938$\pm$0.0025	1.9919$\pm$0.0050	4.4737$\pm$0.0102	7.9773$\pm$0.0174	12.3979$\pm$0.0544
ELS-AAEM	0.4951$\pm$0.0020	1.9937$\pm$0.0049	4.4768$\pm$0.0099	7.9823$\pm$0.0160	11.4144$\pm$3.4005
ELS-AAECG	0.4945$\pm$0.0023	1.9933$\pm$0.0050	4.4762$\pm$0.0097	7.9829$\pm$0.0151	12.4153$\pm$0.0519

Method	Mean CPU time$\pm$std				
	$d = 10$	$d = 20$	$d = 30$	$d = 40$	$d = 50$
EM	0.1144$\pm$0.0524	0.1421$\pm$0.0305	0.1288$\pm$0.0287	0.1494$\pm$0.0215	0.1654$\pm$0.0242
ECG	0.1276$\pm$0.0221	0.1733$\pm$0.0363	0.1607$\pm$0.0356	0.1787$\pm$0.0216	0.2014$\pm$0.0314
λ-EM	0.1681$\pm$0.0374	0.2341$\pm$0.0493	0.2105$\pm$0.0466	0.2282$\pm$0.0297	0.2506$\pm$0.0495
ELS-EM	0.2935$\pm$0.1026	0.3689$\pm$0.0760	0.3287$\pm$0.0752	0.3569$\pm$0.0520	0.3765$\pm$0.0714
AAEM	0.0937$\pm$0.0211	0.1270$\pm$0.0254	0.1212$\pm$0.0282	0.1342$\pm$0.0204	0.1533$\pm$0.0263
AAECG	0.1112$\pm$0.0203	0.1533$\pm$0.0346	0.1407$\pm$0.0295	0.1557$\pm$0.0232	0.1862$\pm$0.0336
ELS-AAEM	0.1407$\pm$0.0310	0.1922$\pm$0.0432	0.1797$\pm$0.0407	0.1991$\pm$0.0304	0.2706$\pm$0.0559
ELS-AAECG	0.1612$\pm$0.0308	0.2218$\pm$0.0542	0.2049$\pm$0.0466	0.2266$\pm$0.0333	0.3118$\pm$0.0513

ELS-EM approach enjoy around 62% higher convergence speed than the EM algorithm. Furthermore, the ELS-EM shows around 61% and 41% higher convergence speed than the ECG and λ-EM methods, respectively. Similar behaviour can also be noted in cases of smaller overlaps. As far as the mean error is considered, Table 4 shows that our proposed ELS-EM algorithm outperforms the EM, the ECG and the λ-EM methods. As discussed previously, the AA-based methods suffer from a lack of convergence in the case of balanced-Gaussian components with high number of observed data points. This is reflected by the relatively high values of its associated mean error. Besides, reported results on averaged CPU time confirm again that the ELS scheme increases to some

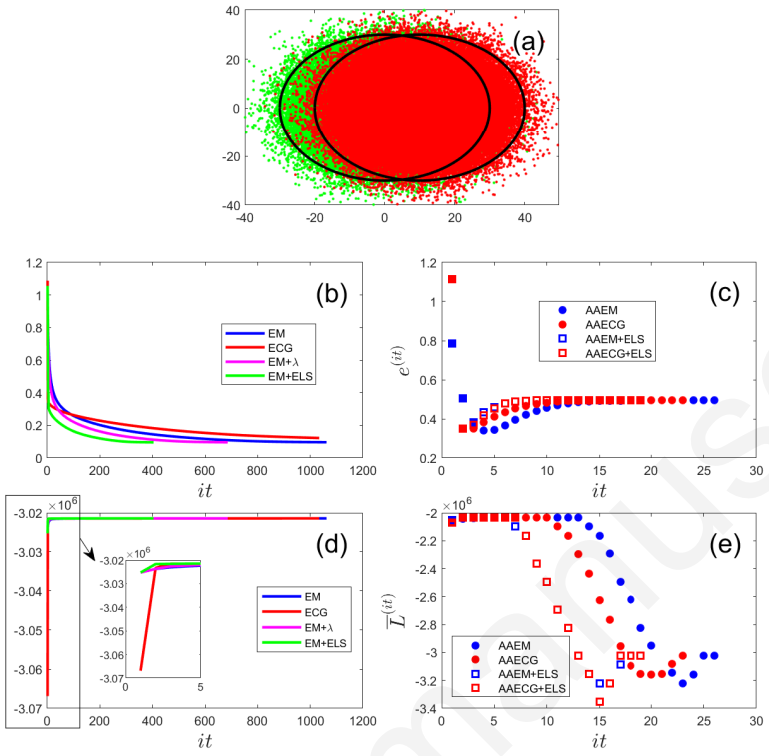

Fig. 5. Performance of the EM, ECG, λ-EM, AAEM, AAECG algorithms, the proposed ELS-EM, ELS-AAEM and ELS-AAECG algorithms in the case of poor-separated clusters ($d = 10$). (a) A two-component GMM, (b, c) mean estimation error, (d, e) averaged log-likelihood.

extent the execution time of the considered algorithms, as shown in Table 4. We stress again on the fact that such increase is not crippling since the latter scheme leads to a higher good identification quality in relatively smaller number of iterations.

The above mentioned results are highlighted in Figures 5 and 3 for which $d = 10$ and $d = 50$, respectively. More particularly, Fig. 5 (d) and Fig. 3 (d) show generally a faster increase in the log-likelihood towards the final solution for the ELS-EM approach compared to the conventional EM, ECG and λ-EM ones. As far as Fig. 5 (c, e) and Fig. 3 (c, e) are concerned, abnormal behaviour in both the mean estimation error and the log-likelihood maximization can be observed for the AA-based methods. Indeed an increase in the mean estimation error, $e^{(it)}$, through iterations is reported for all AA-based methods (Fig. 5 (c) and Fig. 3 (c)). A decrease followed by an increase in

the log-likelihood value is to be noticed for those AA-based approaches (Fig. 5 (e) and Fig. 3 (e)). The case with unbalanced and highly overlapped components has not been considered since according to our preliminary results no method can deal with such a challenging situation.

4.3. A four-component GMM

In this experiment, a four-component GMM is considered. For sake of clarity, only the case of unbalanced Gaussian mixtures is studied. Therefore, following [12], the number of data points simulated from the four Gaussian distributions is respectively equal to $N_1 = 1.5 \times 10^5$, $N_2 = 1 \times 10^5$, $N_3 = 5 \times 10^4$ and $N_4 = 1.5 \times 10^2$ (see Fig. 6 (a)). The four Gaussian components have the following parameters: $\boldsymbol{\mu}_1 = [75, 500]^\mathsf{T}$, $\boldsymbol{\mu}_2 = [50, 10]^\mathsf{T}$, $\boldsymbol{\mu}_3 = [700, 10]^\mathsf{T}$ and $\boldsymbol{\mu}_4 = [650, 500]^\mathsf{T}$; $\boldsymbol{\Sigma}_1 = [100^2 \quad 0; 0 \quad 70^2]$, $\boldsymbol{\Sigma}_2 = [85^2 \quad 0; 0 \quad 70^2]$, $\boldsymbol{\Sigma}_3 = [110^2 \quad 0; 0 \quad 90^2]$ and $\boldsymbol{\Sigma}_4 = [90^2 \quad 0; 0 \quad 90^2]$. Regarding the mixing coefficients vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]^\mathsf{T}$, its components are defined as $\alpha_i = \frac{N_i}{\sum_{i=1}^{4} N_i}, \forall 1 \leq i \leq 4$. As far as the AA-based approaches are concerned, the temperature-related parameter β takes successively the following values 0.2, 0.4, 0.6, 0.8, 1.0, 1.2 and 1.0 [12].

Table 5 shows the mean number of iterations as well as the standard deviation required for the EM, ECG, λ-EM, AAEM, ELS-EM, AAECG, ELS-AAEM and ELS-

Table 5. Mean number of iterations $\pm$ standard deviation, mean error $\pm$ standard deviation taken at the mean number of iterations and mean CPU time per iteration $\pm$ standard deviation, over 50 randomly and independently chosen initial points, for the EM, ECG, λ-EM, AAEM, AAECG algorithms, the proposed ELS-EM, ELS-AAEM and ELS-AAECG algorithms in the case of four-component GMM.

Method	Mean iteration$\pm$std	Mean error$\pm$std	Mean CPU time$\pm$std
EM	543.7$\pm$518.3	24.3877$\pm$30.5674	0.1325$\pm$0.0388
ECG	329.2$\pm$344.4	20.4339$\pm$30.1306	0.1671$\pm$0.0387
λ-EM	384.9$\pm$505.1	22.3899$\pm$30.0999	0.2166$\pm$0.0715
ELS-EM	**202.6$\pm$294.1**	**19.6285$\pm$28.4608**	0.3428$\pm$0.0894
AAEM	89.5$\pm$23.2	35.5152$\pm$30.5008	0.1253$\pm$0.0342
AAECG	80.1$\pm$13.5	46.8577$\pm$32.7895	0.1594$\pm$0.0362
ELS-AAEM	**58.3$\pm$12.3**	45.0653$\pm$51.9933	0.2670$\pm$0.0688
ELS-AAECG	**56.4$\pm$11.6**	47.3368$\pm$37.8952	0.3041$\pm$0.0709

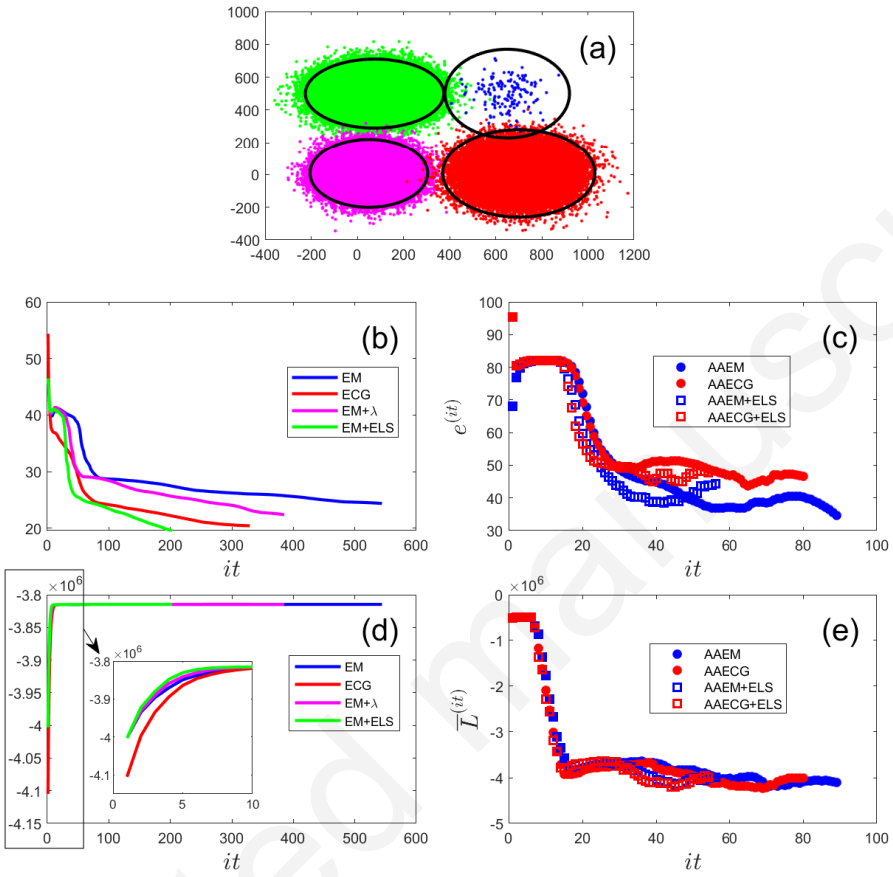

Fig. 6. Performance of the EM, ECG, λ-EM, AAEM, AAECG algorithms, the proposed ELS-EM, ELS-AAEM and ELS-AAECG algorithms in the case of four-component GMM. The four components are unbalanced with $N_1 = 1.5 \times 10^5$, $N_2 = 1 \times 10^5$, $N_3 = 5 \times 10^4$ and $N_4 = 1.5 \times 10^2$. (a) Four-component GMM, (b, c) mean estimation error, (d, e) averaged log-likelihood.

AAECG approaches to converge. Obviously, the proposed ELS-based methods (*e.g.* ELS-EM, ELS-AAEM and ELS-AAECG) outperform their conventional counterparts (*e.g.* EM, AAEM and AAECG). Indeed, an increase around 63%, 35% and 30% in the convergence speed is reported for the EM, AAEM and AAECG algorithms, respectively, when the proposed ELS-scheme is applied. Furthermore, the ELS-EM approach shows around 47% higher convergence speed compared to the λ-EM which can be seen as a

line search scheme. Besides, compared to the ECG approach, the ELS-EM shows around 38% higher convergence speed. As a result, compared to the rest of the considered algorithms in this study, the proposed ELS-EM provides the lowest mean estimation error associated to the obtained mean iteration count. This is despite of its relatively high execution time per iteration, as shown in Table 5.

The aforementioned results are assessed using Fig. 6 where the ELS scheme helps clearly in reducing the number of iterations required to get the final solution in a given search direction compared to the conventional EM, AAEM and AAECG algorithms. Consequently, the proposed ELS-EM, ELS-AAEM and ELS-AAECG methods get their maximum log-likelihood solutions in a relatively smaller number of iterations (even in the case of non-monotonic behaviour of the log-likelihood caused by the permanent change of $h_k^{(it)}(n)$, *i.e.* Eq. (14), with the temperature-related parameter [12]. The performance of the different techniques in terms of the mean error and average log-likelihood is depicted in Fig. 6 (b, c) and Fig. 6 (d, e), respectively. Obtained results confirm again how the ELS scheme when employed allows for a faster convergence of the considered algorithm towards the final solution of a given search direction. Also, as shown in Fig. 6 (b), the proposed strategy can lead to a better identification accuracy since it prevents the algorithms from stacking in swamps and consequently stops before reaching its final solution.

4.4. Real dataset

The behaviour of the different methods considered in this study was evaluated with the MNIST handwritten digits dataset (available online http://yann.lecun.com/exdb/mnist), which consists of 8-bit grayscale images of handwritten digits (0-9) where each image is of size (28×28). N_1 ($N_1 = 5000$) images of handwritten digit '4' and N_2 ($N_2 = 5000$) images of handwritten digit '8' from the training set were randomly selected. Then, these two sets of randomly chosen images were combined to build an observation matrix $\boldsymbol{X}$ of size ($(N_1 + N_2) \times 784$), whose n-th ($1 \leq n \leq (N_1 + N_2)$) row stands for the n-th normalized image. The Principal Component Analysis (PCA) was used next to reduce the dimensionality of the space of $\boldsymbol{X}$ by keeping only the two most informative principal components giving rise

Table 6. Mean number of iterations $\pm$ standard deviation, mean error $\pm$ standard deviation taken at the mean number of iterations and mean CPU time per iteration $\pm$ standard deviation, over 50 randomly and independently chosen initial points, for the EM, ECG, λ-EM, AAEM, AAECG algorithms, the proposed ELS-EM, ELS-AAEM and ELS-AAECG algorithms related to the MNIST digits '4' and '8' dataset.

Method	Mean iteration$\pm$std	Mean error$\pm$std	Mean CPU time$\pm$std
EM	229.6$\pm$18.3	2.7135$\pm$3.3686	0.0019$\pm$0.0002
ECG	173.3$\pm$**6.5**	**1.8639$\pm$1.7510**	0.0055$\pm$0.0007
λ-EM	143.9$\pm$11.6	2.7135$\pm$3.3686	0.0034$\pm$0.0004
ELS-EM	**113.8**$\pm$7.2	2.7131$\pm$3.3688	0.0052$\pm$0.0006
AAEM	199.6$\pm$2.7	**1.0727$\pm$0.0193**	0.0020$\pm$0.0005
AAECG	166.0$\pm$2.8	1.1354$\pm$0.0197	0.0025$\pm$0.0003
ELS-AAEM	**101.9$\pm$2.2**	1.0979$\pm$0.0272	0.0050$\pm$0.0008
ELS-AAECG	**98.6$\pm$2.5**	1.2382$\pm$0.0368	0.0052$\pm$0.0005

to the transposed matrix $\tilde{\boldsymbol{X}}$ of size $(2 \times (N_1 + N_2))$. Since the labels of data points in $\tilde{\boldsymbol{X}}$ were known, a new unbalanced data set denoted here by $\boldsymbol{Y}$ of size $(2 \times (\tilde{N}_1 + \tilde{N}_2))$, with $\tilde{N}_2 << \tilde{N}_1$, was generated. In fact $\tilde{N}_1$ ($\tilde{N}_1 = 5000$ for digit '4') and $\tilde{N}_2$ ($\tilde{N}_2 = 250$ for digit '8') images from the reduced data set $\tilde{X}$ were randomly chosen.

A two overlapped-component GMM as depicted in Fig. 7 (a) was used to approximate the density of the obtained dataset. Table 6 shows the mean number of iterations and the standard derivation for all algorithms considered in our comparative study. According to this table, the proposed ELS-based methods (*e.g.* ELS-EM, ELS-AAEM and ELS-AAECG) show higher convergence speed towards the final solution compared to their standard versions (*e.g.* EM, AAEM and AAECG). Indeed, in terms of number of iterations required to reach the final solution, the ELS-EM provides an acceleration around 34% compared to the ECG algorithm while an enhancement around 21% is noticed compared to the λ-EM algorithm. Regarding the mean error at the obtained mean number of iterations, the ELS-EM algorithm shows higher performance compared to the EM and the λ-EM algorithms. However, lower performance of the ELS-EM is to be noticed in this study compared to the ECG algorithm. In addition, regarding the AA-based algorithms, they outperform the EM, the ECG, the λ-EM and the ELS-EM algorithms in the case of unbalanced Gaussian components with small dataset. As expected, algorithms employing the ELS scheme require higher execution time compared to the conventional ones as shown in Table 6. This fact is also assessed

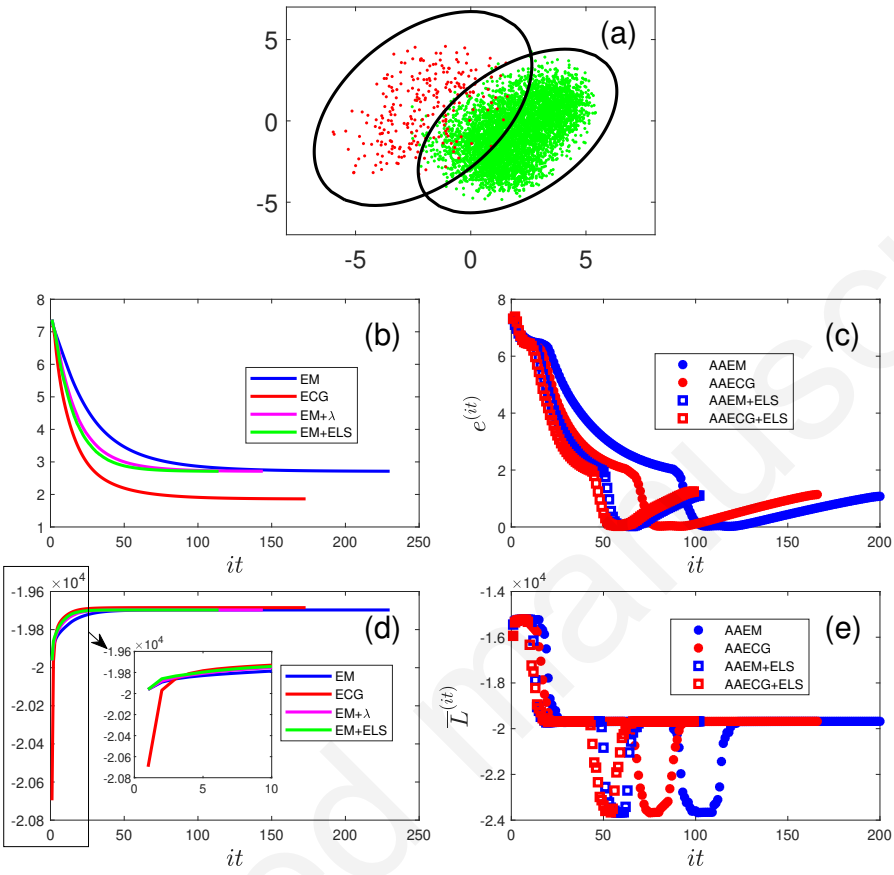

Fig. 7. Performance of the EM, ECG, λ-EM, AAEM, AAECG algorithms, the proposed ELS-EM, ELS-AAEM and ELS-AAECG algorithms in the case of MNIST digits '4' and '8' dataset. The two components are unbalanced with $\tilde{N}_1 = 5000$ and $\tilde{N}_2 = 250$. (a) Dataset $\boldsymbol{Y}$, (b, c) mean estimation error, (d, e) averaged log-likelihood.

using Fig. 7 (b, c) and Fig. 7 (d, e), which show the performance of the considered algorithms in terms of the mean error and the average log-likelihood value as functions of mean number of iterations, respectively.

5. Conclusion

In this paper, an exact line search scheme has been proposed to accelerate the convergence speed of the EM algorithm and its variants, the ECG, the AA-EM and the

AA-ECG methods. The ELS scheme is based on the exact computation, at each iteration, of the step size that should be used towards the final solution in a given direction of the linear search process. The computation of this exact step size is performed by simply rooting a second-order polynomial computed from the initial log-likelihood maximization problem. The proposed ELS scheme has been evaluated in the context of two and four-component GMMs and also in the context of MINST handwritten digit dataset. Its behaviour has been analyzed in case of balanced, unbalanced, well-separated and poorly separated clusters. The numerical results showed the noticeable improvement in the convergence speed of the aforementioned algorithm when the ELS scheme is employed. Furthermore, the ELS-based approaches, especially the ELS-EM, showed generally a higher performance than the conventional ECG and the λ-EM algorithms.

Acknowledgments

This work was supported by the CRIBs, the National Key R&D Program of China (2017YFC0107900), the Short-term Recruitment Program of Foreign Experts (WQ20163200398), the NSFC (31400842) and the China Scholarship Council (CSC).

Appendix A: solution of Eq. (19)

The optimal step size $\rho_{\alpha_k^{(it)}}$ maximizing the Lagrangian function, $L(\rho_{\alpha_k^{(it)}}, \xi)$, Eq. (21) associated to the $\boldsymbol{P1}$ problem in Eq. (19) is computed as follows:

$$\frac{\partial L\left(\rho_{\alpha_k^{(it)}},\xi\right)}{\partial \rho_{\alpha_k^{(t)}}} = \sum_{n=1}^{N} \frac{1}{\left(\alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)}\right)} \times h_k^{(it)}(n) \times G_{\alpha_k}^{(it)} + \xi \times G_{\alpha_k}^{(it)} \tag{26}$$

Then we set:

$$\frac{\partial L\left(\rho_{\alpha_k^{(it)}},\xi\right)}{\partial \rho_{\alpha_k^{(t)}}} = 0 \tag{27}$$

which implies:

$$\alpha_k^{(it-1)} + \rho_{\alpha_k^{(it)}} G_{\alpha_k}^{(it)} = -\sum_{n=1}^{N} \frac{1}{\xi} \times h_k^{(it)}(n) \tag{28}$$

On the other hand, we have:

$$\begin{aligned} &1=\sum_{k=1}^{K}\left(\alpha_k^{(it-1)}+\rho_{\alpha_k^{(it)}}G_{\alpha_k}^{(it)}\right)=-\sum_{k=1}^{K}\sum_{n=1}^{N}\tfrac{1}{\xi}\times h_k^{(it)}(n)=-\sum_{n=1}^{N}\tfrac{1}{\xi}=-\tfrac{N}{\xi}\\ &\Rightarrow \xi=-N \end{aligned} \tag{29}$$

so that:

$$\begin{aligned} &\alpha_k^{(it-1)}+\rho_{\alpha_k^{(it)}}G_{\alpha_k}^{(it)}=-\sum_{n=1}^{N}\tfrac{1}{\xi}\times h_k^{(it)}(n)=\sum_{n=1}^{N}\tfrac{h_k^{(it)}(n)}{N}\\ &\Rightarrow \rho_{\alpha_k^{(it)}}=\left(\sum_{n=1}^{N}\frac{h_k^{(it)}(n)}{N}-\alpha_k^{(it-1)}\right)/G_{\alpha_k}^{(it)} \end{aligned} \tag{30}$$

Appendix B: solution of Eq. (20)

In order to solve $\boldsymbol{P2}$ in Eq. (20), and based on the following statement: $(\boldsymbol{Q}+\sigma^2\boldsymbol{M})^{-1}\simeq\boldsymbol{Q}^{-1}-\sigma^2\boldsymbol{Q}^{-1}\boldsymbol{M}\boldsymbol{Q}^{-1}$ [28], we can write:

$$(\boldsymbol{\Sigma}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)})^{-1}\simeq\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}-\rho^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1} \tag{31}$$

Then, Eq. (20) can be rewritten as follows:

$$\begin{aligned} &\arg\max_{\rho^{(it)}}\left\{Q\left(\boldsymbol{\theta}^{(new)}\middle|\boldsymbol{\theta}^{(it)}\right)\right\}\\ &\simeq\arg\max_{\rho^{(it)}}\left\{\sum_{n=1}^{N}\sum_{k=1}^{K}\left\{\begin{array}{l} -\frac{1}{2}\log\det\left(\boldsymbol{\Sigma}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right)\\ -\frac{1}{2}\left(\boldsymbol{x_n}-\left(\boldsymbol{\mu}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right)^{\mathsf{T}}\\ \times\left(\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}-\rho^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\right)\\ \times\left(\boldsymbol{x_n}-\left(\boldsymbol{\mu}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right) \end{array}\right\}h_k^{(it)}(n)\right\} \end{aligned} \tag{32}$$

Then, the derivative of $Q\left(\boldsymbol{\theta}^{(new)}\middle|\boldsymbol{\theta}^{(it)}\right)$ with respect to $\rho^{(it)}$ is given by:

$$
\begin{aligned}
&\frac{\partial Q\left(\boldsymbol{\theta}^{(new)}\middle|\boldsymbol{\theta}^{(t)}\right)}{\partial \rho^{(it)}} \\
&\simeq \sum_{n=1}^{N}\sum_{k=1}^{K}\left\{\begin{array}{l}
-\frac{1}{2}\mathrm{Tr}\left(\left(\boldsymbol{\Sigma}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right) \\
+\frac{1}{2}\left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\mathsf{T}}\left(\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}-\rho^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\right) \\
\times\left(\boldsymbol{x_n}-\left(\boldsymbol{\mu}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right) \\
+\frac{1}{2}\left(\boldsymbol{x_n}-\left(\boldsymbol{\mu}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1} \\
\times\left(\boldsymbol{x_n}-\left(\boldsymbol{\mu}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right) \\
+\frac{1}{2}\left(\boldsymbol{x_n}-\left(\boldsymbol{\mu}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right)^{\mathsf{T}} \\
\times\left(\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}-\rho^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\right)\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}
\end{array}\right\}h_k^{(it)}(n) \\
&\simeq \sum_{n=1}^{N}\sum_{i=1}^{K}\left\{\begin{array}{l}
-\frac{1}{2}\mathrm{Tr}\left(\left(\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}-\rho^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\right)\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right) \\
+\left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\mathsf{T}}\left(\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}-\rho^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\right) \\
\times\left(\boldsymbol{x_n}-\left(\boldsymbol{\mu}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right) \\
+\frac{1}{2}\left(\boldsymbol{x_n}-\left(\boldsymbol{\mu}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1} \\
\times\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\left(\boldsymbol{x_n}-\left(\boldsymbol{\mu}_k^{(it-1)}+\rho^{(it)}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)\right)
\end{array}\right\}h_k^{(it)}(n)
\end{aligned}
\tag{33}
$$

Now this derivative can be written as a polynomial in $\rho^{(it)}$ as follows:

$$
\begin{aligned}
&\frac{\partial Q\left(\boldsymbol{\theta}^{(new)}\middle|\boldsymbol{\theta}^{(t)}\right)}{\partial \rho^{(it)}} \simeq \\
&\simeq \sum_{n=1}^{N}\sum_{k=1}^{K}\left\{\begin{array}{l}
-\frac{1}{2}\mathrm{Tr}\left(\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right) \\
-\frac{1}{2}\rho^{(it)}\mathrm{Tr}\left(\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right) \\
+\left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\left(\boldsymbol{x_n}-\boldsymbol{\mu}_k^{(it-1)}\right) \\
-\rho^{(it)}\left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\left(\boldsymbol{x_n}-\boldsymbol{\mu}_k^{(it-1)}\right) \\
-\rho^{(it)}\left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)} \\
+\left(\rho^{(it)}\right)^2\left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)} \\
+\frac{1}{2}\left(\boldsymbol{x_n}-\boldsymbol{\mu}_k^{(it-1)}\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\left(\boldsymbol{x_n}-\boldsymbol{\mu}_k^{(it-1)}\right) \\
-\rho^{(it)}\left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\left(\boldsymbol{x_n}-\boldsymbol{\mu}_k^{(it-1)}\right) \\
\frac{1}{2}\left(\rho^{(it)}\right)^2\left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\mathsf{T}}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}
\end{array}\right\}h_k^{(it)}(n) \\
&=-\frac{1}{2}\eta_1^{(it)}-\frac{1}{2}\rho^{(it)}\eta_2^{(it)}+\eta_3^{(it)}-\rho^{(it)}\eta_4^{(it)}+-\rho^{(it)}\eta_5^{(it)} \\
&+\left(\rho^{(it)}\right)^2\eta_6^{(it)}+\frac{1}{2}\eta_7^{(it)}-\rho^{(it)}\eta_4^{(it)}+\frac{1}{2}\left(\rho^{(it)}\right)^2\eta_6^{(it)} \\
&=y_2^{(it)}\left(\rho^{(it)}\right)^2+y_1^{(it)}\rho^{(it)}+y_0^{(it)}
\end{aligned}
\tag{34}
$$

where

$$
\begin{aligned}
\eta_1^{(it)} &= \sum_{n=1}^{N}\sum_{k=1}^{K} \mathrm{Tr}\left(\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right) h_k^{(it)}(n)\\
\eta_2^{(it)} &= \sum_{n=1}^{N}\sum_{k=1}^{K} \mathrm{Tr}\left(\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\right) h_k^{(it)}(n)\\
\eta_3^{(it)} &= \sum_{n=1}^{N}\sum_{k=1}^{K} \left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\top}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\left(\boldsymbol{x_n}-\boldsymbol{\mu}_k^{(it-1)}\right) h_k^{(it)}(n)\\
\eta_4^{(it)} &= \sum_{n=1}^{N}\sum_{k=1}^{K} \left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\top}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\left(\boldsymbol{x_n}-\boldsymbol{\mu}_k^{(it-1)}\right) h_k^{(it)}(n)\\
\eta_5^{(it)} &= \sum_{n=1}^{N}\sum_{k=1}^{K} \left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\top}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)} h_k^{(it)}(n)\\
\eta_6^{(it)} &= \sum_{n=1}^{N}\sum_{k=1}^{K} \left(\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)}\right)^{\top}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\mu}_k}^{(it)} h_k^{(it)}(n)\\
\eta_7^{(it)} &= \sum_{n=1}^{N}\sum_{k=1}^{K} \left(\boldsymbol{x_n}-\boldsymbol{\mu}_k^{(it-1)}\right)^{\top}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\boldsymbol{G}_{\boldsymbol{\Sigma}_k}^{(it)}\left(\boldsymbol{\Sigma}_k^{(it-1)}\right)^{-1}\left(\boldsymbol{x_n}-\boldsymbol{\mu}_k^{(it-1)}\right) h_k^{(it)}(n)\\
y_2^{(it)} &= \tfrac{3}{2}\eta_6^{(it)}\\
y_1^{(it)} &= \tfrac{1}{2}\eta_2^{(it)} - 2\eta_4^{(it)} - \eta_5^{(it)}\\
y_0^{(it)} &= -\tfrac{1}{2}\eta_1^{(it)} + \eta_3^{(it)} + \tfrac{1}{2}\eta_7^{(it)}
\end{aligned}
\tag{35}
$$

References

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

[2] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.

[3] A. K. Jain, R. P. W. Duin, J. Mao, Statistical pattern recognition: A review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4–37.

[4] J. G. Dy, C. E. Brodley, Feature selection for unsupervised learning, Journal of Machine Learning Research 5 (Aug) (2004) 845–889.

[5] D. Naik, P. Shah, A review on image segmentation clustering algorithms, International Journal of Computer Science and Information Technologies 5 (3) (2014) 3289–93.

[6] W. Wu, H. Xiong, S. Shekhar, Clustering and information retrieval, Vol. 11, Springer Science & Business Media, 2013.

[7] P. Berkhin, A survey of clustering data mining techniques, Grouping Multidimensional Data 25 (2006) 71.

[8] J. R. Montalvão Filho, B. Dorizzi, J. C. M. Mota, Channel estimation by symmetrical clustering, IEEE Transactions on Signal Processing 50 (6) (2002) 1459–1469.

[9] G. Amit, N. Gavriely, N. Intrator, Cluster analysis and classification of heart sounds, Biomedical Signal Processing and Control 4 (1) (2009) 26–36.

[10] X. Meng, D. Van Dyk, Fast EM-type implementations for mixed effects models, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 60 (3) (1998) 559–578.

[11] C. J. Wu, On the convergence properties of the EM algorithm, The Annals of Statistics (1983) 95–103.

[12] I. Naim, D. Gildea, Convergence of the EM algorithm for gaussian mixtures with unbalanced mixing coefficients, in: Proceedings of the 29th International Conference on Machine Learning (ICML-12), 2012, pp. 1655–1662.

[13] K. Lange, A gradient algorithm locally equivalent to the EM algorithm, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 52 (2) (1995) 425–437.

[14] M. Jamshidian, R. I. Jennrich, Acceleration of the EM algorithm by using quasi-newton methods, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 59 (3) (1997) 569–587.

[15] R. Salakhutdinov, S. Roweis, Z. Ghahramani, Expectation-conjugate gradient: An alternative to EM, IEEE Signal Processing Letters 11 (7) (2004).

[16] R. Salakhutdinov, S. Roweis, Z. Ghahramani, Optimization with EM and expectation-conjugate-gradient, in: Proceedings of the 20th International Conference on Machine Learning (ICML-03), 2003, pp. 672–679.

[17] S. E. Atkinson, The performance of standard and hybrid EM algorithms for ml estimates of the normal mixture model with censoring, Journal of Statistical Computation and Simulation 44 (1-2) (1992) 105–115.

[18] R. A. Redner, H. F. Walker, Mixture densities, maximum likelihood and the EM algorithm, SIAM Review 26 (2) (1984) 195–239.

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

[20] J. Nocedal, S. Wright, Numerical optimization, Springer, 1999.

[21] M. Rajih, P. Comon, R. A. Harshman, Enhanced line search: A novel method to accelerate PARAFAC, SIAM journal on Matrix Analysis and Applications 30 (3) (2008) 1128–1147.

[22] L. Sorber, I. Domanov, M. Van Barel, L. De Lathauwer, Exact line and plane search for tensor optimization, Computational Optimization and Applications 63 (1) (2016) 121–142.

[23] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, C. Faloutsos, Tensor decomposition for signal processing and machine learning, IEEE Transactions on Signal Processing 65 (13) (2017) 3551–3582.

[24] C. F. Gerald, Applied numerical analysis, Pearson Education India, 2004.

[25] N. Laird, N. Lange, D. Stram, Maximum likelihood computations with repeated measures: application of the EM algorithm, Journal of the American Statistical Association 82 (397) (1987) 97–105.

[26] A. Karfoul, L. Albera, L. De Lathauwer, Iterative methods for the canonical decomposition of multi-way arrays: Application to blind underdetermined mixture identification, Signal Processing 91 (8) (2011) 1789–1802.

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

[28] K. B. Petersen, M. S. Pedersen, The matrix cookbook, Technical University of Denmark 7 (2008) 15.