Comput Stat
DOI 10.1007/s00180-015-0565-y

ORIGINAL PAPER

Improving the vector ε acceleration for the EM algorithm using a re-starting procedure

Masahiro Kuroda · Zhi Geng · Michio Sakakihara

Received: 15 November 2013 / Accepted: 31 January 2015

Abstract The expectation–maximization (EM) algorithm is a popular algorithm for finding maximum likelihood estimates from incomplete data. However, the EM algorithm converges slowly when the proportion of missing data is large. Although many acceleration algorithms have been proposed, they require complex calculations. Kuroda and Sakakihara (Comput Stat Data Anal 51:1549–1561, 2006) developed the ε-accelerated EM algorithm which only uses the sequence of estimates obtained by the EM algorithm to get an accelerated sequence for the EM sequence but does not change the original EM sequence. We find that the accelerated sequence often has larger values of the likelihood than the current estimate obtained by the EM algorithm. Thus, in this paper, we try to re-start the EM iterations using the accelerated sequence and then generate a new EM sequence that increases its speed of convergence. This algorithm has another advantage of simple implementation since it only uses the EM iterations and re-starts the iterations by an estimate with a larger likelihood. The re-starting algorithm called the εR-accelerated EM algorithm can further improve the EM algorithm and the ε-accelerated EM algorithm in the sense of that it can reduces the number of iterations and computation time.

M. Kuroda (✉)
Department of Socio-Information, Okayama University of Science,
1-1 Ridaicho, Okayama 700-0005, Japan
e-mail: kuroda@soci.ous.ac.jp

Z. Geng
School of Mathematical Sciences, LMAM, Peking University, Beijing 100871, China

M. Sakakihara
Department of Information Science, Okayama University of Science,
1-1 Ridaicho, Okayama 700-0005, Japan

Published online: 10 February 2015

Keywords The vector ε algorithm · The EM algorithm · Re-starting procedure · Acceleration of convergence

1 Introduction

The expectation–maximization (EM) algorithm formulated by Dempster et al. (1977) is a general and popular algorithm for finding maximum likelihood estimates (MLEs) from incomplete data due to stability in convergence, simplicity in implementation and applicability in practice. However, the drawback of the EM algorithm is that its convergence is linear and very slow when the proportion of missing data is high.

In order to circumvent the problem of slow convergence of the EM algorithm, various acceleration algorithms incorporating optimization methods with faster convergence rate have been proposed. The optimization methods include the multivariate Aitken method of Louis (1982) and Laird et al. (1987), the conjugate gradient method of Jamshidian and Jennrich (1993) and the quasi-Newton method of Lange (1995) and Jamshidian and Jennrich (1997). However, they require the matrix computation such as matrix inversion or evaluation of Hessian and Jacobian matrices and a line search for step length optimization. Therefore their acceleration algorithms tend to lack one or more of the nice properties of the EM algorithm, although they converge faster than the EM algorithm.

Kuroda and Sakakihara (2006) developed the ε-accelerated EM algorithm for accelerating the convergence of the sequence of EM iterations using the vector ε algorithm of Wynn (1962). The algorithm consists two steps: The first step is the expectation and maximization steps (the EM step) of the EM algorithm and the second step is the acceleration step using the vector ε algorithm. The vector ε algorithm is a fairly simple computational procedure and its implementation is very easy. Moreover, its computational cost is much less than that of any optimization method. The merit of the ε-accelerated EM algorithm is that it requires only the sequence of EM iterations for acceleration and maintains the nice properties of the EM algorithm. In the numerical experiments, Kuroda and Sakakihara (2006) demonstrated that the ε-accelerated EM algorithm significantly accelerates the convergence of the sequence of EM iterations. Wang et al. (2008) provided theorems concerning convergence and acceleration of the ε-accelerated EM algorithm.

In order to further reduce the number of iterations and computation time, we improve the ε-accelerated EM algorithm using a re-starting procedure. The re-starting procedure embedding in the acceleration step finds an initial value for re-starting the EM step such that a newly generated sequence of EM iterations from the value moves quickly into a neighborhood of a stationary point. When applying the ε-accelerated EM algorithm to the newly generated sequence, its speed of convergence can be increased. Therefore the use of the re-starting procedure makes the ε-accelerated EM algorithm converge faster. We refer the ε-accelerated EM algorithm with a re-starting procedure to the εR-accelerated EM algorithm.

The paper is organized as follows. Section 2 describes the ε-accelerated EM algorithm. In Sect. 3, we provide the εR-accelerated EM algorithm and show some theoretical results concerning its acceleration. Section 4 presents numerical experiments to

illustrate the behavior of convergence of the εR-accelerated EM algorithm. In Sect. 5, we present our concluding remarks.

2 The ε-accelerated EM algorithm

Let $\mathbf{y}$ be the incompletely observed data in a sample space $\Omega_{\mathbf{y}}$ and $\mathbf{x}$ be the complete data augmented from $\mathbf{y}$ in a sample space $\Omega_{\mathbf{x}}$. Assume that there exists some function $h(\mathbf{x}) = \mathbf{y}$ relating $\mathbf{x}$ to $\mathbf{y}$. Let $f(\cdot|\theta)$ denote a density function with an unknown d-dimensional parameter vector $\theta = (\theta_1, \ldots, \theta_d)^\top$ in a parameter space Θ and $L(\theta) = \log f(\cdot|\theta)$ be the log-likelihood function of θ. We denote the log-likelihood function for observed data $\mathbf{y}$ by $L_o(\theta) = \log f(\mathbf{y}|\theta)$, and the log-likelihood function for complete data $\mathbf{x}$ by $L_c(\theta) = \log f(\mathbf{x}|\theta)$. Denote the conditional expectation of $L_c(\theta)$ given $\mathbf{y}$ and θ' by

$$Q(\theta|\theta') = \mathrm{E}[L_c(\theta)|\mathbf{y}, \theta'].$$

The EM algorithm iteratively finds the sequence of EM estimates by

$$\theta^{(t+1)} = \arg\max_{\theta\in\Theta} Q(\theta|\theta^{(t)})$$

at the tth iteration for $t = 0, 1, \ldots$.

First we describe the EM algorithm for an initial value $\theta^{(0)} \in \Theta$ as follows:

– **E-step:** Calculate the expectation

$$Q(\theta|\theta^{(t)}) = \mathrm{E}[L_c(\theta)|\mathbf{y}, \theta^{(t)}].$$

– **M-step:** Find

$$\theta^{(t+1)} = \arg\max_{\theta\in\Theta} Q(\theta|\theta^{(t)}).$$

The E- and M-steps are repeated to produce the sequence of estimates $\{\theta^{(t)}\}_{t\geq 0}$. We define a mapping $\theta \mapsto M(\theta)$ from Θ to Θ such that each iteration $\theta^{(t)} \to \theta^{(t+1)}$ is denoted by

$$\theta^{(t+1)} = M(\theta^{(t)}). \tag{1}$$

The EM algorithm has the property

$$L_o(\theta^{(t+1)}) \geq L_o(\theta^{(t)}), \tag{2}$$

and $\{\theta^{(t)}\}_{t\geq 0}$ converges to a stationary point θ^*.

Next we briefly introduce the vector ε algorithm presented by Wynn (1962) to accelerate the convergence of a slowly convergent vector sequence. The algorithm is very effective for linearly convergent sequences. Let $\{\theta^{(t)}\}_{t\geq 0}$ be a linearly convergent vector sequence generated by an iterative computational procedure and converge to a

stationary point θ^* as $t \to \infty$. Then the vector ε algorithm generates the accelerated sequence $\{\dot{\theta}^{(t)}\}_{t\geq 0}$ for $\{\theta^{(t)}\}_{t\geq 0}$ by

$$\dot{\theta}^{(t-1)} = \theta^{(t)} + \left[\left[\theta^{(t-1)} - \theta^{(t)}\right]^{-1} + \left[\theta^{(t+1)} - \theta^{(t)}\right]^{-1}\right]^{-1}, \tag{3}$$

where $[\theta]^{-1} = \theta/\|\theta\|^2$ and $\|\theta\|$ is the Euclidean norm of θ, see "Appendix 1" for details.

Below we describe the ε-accelerated EM algorithm proposed by Kuroda and Sakakihara (2006). Given an initial value $\theta^{(0)} \in \Theta$, the ε-accelerated EM algorithm repeats the following steps:

– **EM step:** Find

$$\theta^{(t+1)} = M(\theta^{(t)}).$$

– **ε-acceleration:** Use the EM sequence $(\theta^{(t+1)}, \theta^{(t)}, \theta^{(t-1)})$ to generate the accelerated sequence from

$$\dot{\theta}^{(t-1)} = \theta^{(t)} + \left[\left[\theta^{(t-1)} - \theta^{(t)}\right]^{-1} + \left[\theta^{(t+1)} - \theta^{(t)}\right]^{-1}\right]^{-1},$$

until

$$||\dot{\theta}^{(t-1)} - \dot{\theta}^{(t-2)}||^2 \leq \delta,$$

where δ is a desired accuracy.

The ε-accelerated EM algorithm generates two sequences: One is the EM sequence $\{\theta^{(t)}\}_{t\geq 0}$ at the EM step, and the other is the accelerated sequence $\{\dot{\theta}^{(t)}\}_{t\geq 0}$ at the ε-acceleration step. The accelerated sequence does not make any effect on the EM sequence. Wang et al. (2008) showed that $\{\dot{\theta}^{(t)}\}_{t\geq 0}$ converges to the same stationary point of $\{\theta^{(t)}\}_{t\geq 0}$ and it converges faster than $\{\theta^{(t)}\}_{t\geq 0}$. The ε-accelerated EM algorithm only uses the EM sequence, and thus it maintains the stability and simplicity of the EM algorithm.

3 Improvement of the ε-accelerated EM algorithm by using a re-starting procedure: the εR-accelerated EM algorithm

The ε-accelerated EM algorithm generates two parallel sequences, the accelerated sequence $\{\dot{\theta}^{(t)}\}_{t\geq 0}$ and the EM sequence $\{\theta^{(t)}\}_{t\geq 0}$. But at the ε-acceleration step, $\dot{\theta}^{(t-1)}$ may make the next EM estimate $M(\dot{\theta}^{(t-1)})$ have a larger likelihood value than the current EM estimate $\theta^{(t+1)}$, that is, $L_o(M(\dot{\theta}^{(t-1)})) > L_o(\theta^{(t+1)})$. Thus, when this occurs, we re-start the EM iterations with the initial value $\dot{\theta}^{(t-1)}$, stop the original EM sequence, and get $\dot{\theta}^{(t)}$ from $(\dot{\theta}^{(t-1)}, M(\dot{\theta}^{(t-1)}), M(M(\dot{\theta}^{(t-1)})))$. Notice that at the re-starting point, we still generate the EM sequence using three estimates obtained

from the same initial value $\dot{\theta}^{(t-1)}$. That is, we keep to always apply the ε-acceleration to a sequence obtained by the EM mapping $M()$ from the same initial value. This re-starting algorithm proposed here is called the εR-accelerated EM algorithm.

By our experiments, the re-starting procedure is performed almost every time only by the re-starting condition $L_o(M(\dot{\theta}^{(t-1)})) > L_o(\theta^{(t+1)})$, and it inefficiently takes much computation time. Thus we add one more condition for re-starting $\|\dot{\theta}^{(t-1)} - \dot{\theta}^{(t-2)}\|^2 \leq \delta_{Re} (> \delta)$, and we reset $\delta_{Re} = \delta_{Re}/10^k$ at each re-starting, where k is an integer, such as 1. By this condition, we control the re-starting frequency. For example, let $\delta = 10^{-12}$ for stopping condition, and initialize $\delta_{Re} = 1$ and $k = 1$. Then the re-starting procedure is performed at most 12 times. The conditions for re-starting are summarized as follows:

(i) $L_o(M(\dot{\theta}^{(t-1)})) > L_o(\theta^{(t+1)})$, and
(ii) $\|\dot{\theta}^{(t-1)} - \dot{\theta}^{(t-2)}\|^2 < \delta_{Re}$.

Condition (i) means that the likelihood can be increased by the re-starting. Condition (ii) is used to reduce the frequency of re-starting. This is the key idea of the re-starting procedure.

The εR-accelerated EM algorithm repeats the following steps:

– **EM step:** Find

$$\theta^{(t+1)} = M(\theta^{(t)}).$$

– ***ε*-acceleration step:** Use $(\theta^{(t+1)}, \theta^{(t)}, \theta^{(t-1)})$ to generate the accelerated sequence from

$$\dot{\theta}^{(t-1)} = \theta^{(t)} + \left[\left[\theta^{(t-1)} - \theta^{(t)}\right]^{-1} + \left[\theta^{(t+1)} - \theta^{(t)}\right]^{-1}\right]^{-1}.$$

– **re-starting step:** If $L_o(M(\dot{\theta}^{(t-1)})) > L_o(\theta^{(t+1)})$ and $\|\dot{\theta}^{(t-1)} - \dot{\theta}^{(t-2)}\|^2 < \delta_{Re}$, then set

$$\theta^{(t)} = \dot{\theta}^{(t-1)},$$

update

$$\theta^{(t+1)} = M(\dot{\theta}^{(t-1)}),$$

and reset $\delta_{Re} = \delta_{Re}/10^k$.

Set $t = t + 1$. Repeat the above steps until

$$||\dot{\theta}^{(t-1)} - \dot{\theta}^{(t-2)}|| \leq \delta.$$

The initial value δ_{Re} and the size of decrement 10^{-k} are related to the improvement of the computational efficiency of the ε-accelerated EM algorithm. When setting $\delta_{Re} = 1$ and the decrement of 10^0, the algorithm may re-start in every iteration after several

iterations. Then the computation time for obtaining the re-starting EM sequence is twice that for doing $\{\theta^{(t)}\}_{t\geq 0}$ in each iteration. In the case of setting the larger size of decrement such as 10^{-8} or 10^{-10} for $\delta = 10^{-12}$, the re-starting is performed a few times. In both cases, the computation time of the εR-accelerated EM algorithm may take longer than or the same as that of the ε-accelerated EM algorithm. Thus when the re-starting step effectively finds initial values, the εR-accelerated EM algorithm greatly reduces the number of iterations and computation time for convergence.

The advantage of the εR-accelerated EM algorithm over the ε-accelerated EM algorithm is that it re-starts the iterations of the EM algorithm at a better current estimate and also keeps that the likelihood increases in the iterations. We give the pseudo-code of the εR-accelerated EM algorithm in "Appendix 2".

We describe the sequences obtained by two algorithms at iterations:

The sequences of the ε-accelerated EM algorithm

$-$EM

$$\{\theta^{(t)}\}_{t\geq 0} : \theta^{(0)}\ \theta^{(1)}\ \theta^{(2)}\quad \ldots\quad \theta^{(t-1)}\ \theta^{(t)}\qquad \theta^{(t+1)}\qquad \theta^{(t+2)}\ \theta^{(t+3)}\ \ldots$$

$-\varepsilon$-acceleration

$$\{\dot{\theta}^{(t)}\}_{t\geq 0} : \dot{\theta}^{(0)}\ \ldots\ \dot{\theta}^{(t-3)}\ \dot{\theta}^{(t-2)}\ \dot{\theta}^{(t-1)}\ \dot{\theta}^{(t)}\qquad \dot{\theta}^{(t+1)}\qquad \ldots$$

The sequences of the εR-accelerated EM algorithm

$-$re-started EM

Re-start at the $(t+1)$th iteration

$$\{\tilde{\theta}^{(t)}\}_{t\geq 0} : \theta^{(0)}\ \theta^{(1)}\ \theta^{(2)}\quad \ldots\quad \theta^{(t-1)}\ \theta^{(t)}\ (= \dot{\theta}^{(t-1)})\ \Big|\ \tilde{\theta}^{(t+1)}\ (= M(\dot{\theta}^{(t-1)}))\qquad \tilde{\theta}^{(t+2)}\ \tilde{\theta}^{(t+3)}\ \ldots$$

$-$re-started ε-acceleration

$$\{\dot{\tilde{\theta}}^{(t)}\}_{t\geq 0} : \dot{\theta}^{(0)}\ \ldots\ \dot{\theta}^{(t-3)}\ \dot{\theta}^{(t-2)}\ \dot{\theta}^{(t-1)}\ \dot{\tilde{\theta}}^{(t)}\qquad \dot{\tilde{\theta}}^{(t+1)}\qquad \ldots$$

When the re-starting procedure is performed at the $(t+1)$th iteration, we obtain a new EM sequence $\{\tilde{\theta}^{(s)}\}_{s\geq t+1}$ starting from $\tilde{\theta}^{(t+1)} = M(\dot{\theta}^{(t-1)})$, and then the εR-accelerated EM algorithm generates $\{\dot{\tilde{\theta}}^{(s)}\}_{s\geq t+1}$ from the re-started EM sequence $\{\tilde{\theta}^{(s)}, \tilde{\theta}^{(s+1)}, \tilde{\theta}^{(s+2)}\}$. Note that $\dot{\tilde{\theta}}^{(t)}$ is calculated using $\{\theta^{(t)}, \tilde{\theta}^{(t+1)}, \tilde{\theta}^{(t+2)}\}$.

Now we discuss and compare the convergence of these sequences. We have from Meng and Rubin (1994)

$$\theta^{(t+1)} - \theta^* = DM(\theta^*)(\theta^{(t)} - \theta^*) + O(||\theta^{(t)} - \theta^*||^2),$$

where

$$DM(\theta) = \left(\frac{\partial M_j(\theta)}{\partial \theta_i}\right)$$

is the $d \times d$ Jacobian matrix for the mapping $M(\theta) = (M_1(\theta), \ldots, M_d(\theta))^\top$. We assume that t is sufficiently large and $\theta^{(t)}$ is in a neighborhood of θ^*. Then the EM algorithm is essentially a linear iteration

$$\theta^{(t+1)} - \theta^* = DM(\theta^*)(\theta^{(t)} - \theta^*) \tag{4}$$

with iteration matrix $DM(\theta^*)$. The largest eigenvalue of $DM(\theta^*)$ governs the rate of convergence of the EM algorithm and the linear iteration (4) has the same rate of convergence as the EM algorithm. Schafer (1997, p. 59) provides the approximation

$$\theta^{(t+1)} - \theta^* = \lambda(\theta^{(t)} - \theta^*), \tag{5}$$

where λ is the largest eigenvalue of $DM(\theta^*)$. By Eq. (5), we have

$$\|\theta^{(t+2)} - \theta^*\| = \lambda\|\theta^{(t+1)} - \theta^*\| = \lambda^2\|\theta^{(t)} - \theta^*\|$$

and therefore obtain

$$\|\theta^{(t+L)} - \theta^*\| = \lambda^L\|\theta^{(t)} - \theta^*\|. \tag{6}$$

Under some conditions (Little and Rubin 1987), the estimate of θ has asymptotically the normal distribution with mean θ^*. Then the closer to θ^* the estimate is, the larger the density is. By $L_o(M(\dot{\theta}^{(t-1)})) > L_o(\theta^{(t+1)})$, we have

$$\|\tilde{\theta}^{(t+1)} - \theta^*\| = \|M(\dot{\theta}^{(t-1)}) - \theta^*\| < \|\theta^{(t+1)} - \theta^*\|.$$

The mapping of $\tilde{\theta}^{(t)}$ is the same as the mapping of $\theta^{(t)}$, and they are different only by the parameter values. Thus we have

$$\tilde{\theta}^{(t+1)} - \theta^* = \lambda(\tilde{\theta}^{(t)} - \theta^*).$$

Similar to Eq. (6), we obtain

$$\|\tilde{\theta}^{(t+L)} - \theta^*\| = \lambda^L\|\tilde{\theta}^{(t)} - \theta^*\|. \tag{7}$$

Thus we have from Eqs. (6) and (7)

$$\|\tilde{\theta}^{(t+L)} - \theta^*\| < \|\theta^{(t+L)} - \theta^*\|. \tag{8}$$

When inequality (8) holds, we obtain the following result of the speed of convergence of $\{\dot{\theta}^{(t)}\}_{t\geq 0}$ and $\{\dot{\tilde{\theta}}^{(t)}\}_{t\geq 0}$.

Theorem 1 *Let $\{\dot{\tilde{\theta}}^{(t)}\}_{t\geq 0}$ be the sequence generated by the εR-accelerated EM algorithm and $\{\dot{\theta}^{(t)}\}_{t\geq 0}$ be the sequence by the ε-accelerated EM algorithm. Then $\{\dot{\tilde{\theta}}^{(t)}\}_{t\geq 0}$ converges to θ^* more quickly than $\{\dot{\theta}^{(t)}\}_{t>0}$.*

Proof First we provide the result that $\{\dot{\theta}^{(t)}\}_{t\geq 0}$ and $\{\dot{\tilde{\theta}}^{(t)}\}_{t\geq 0}$ converge to θ^* faster than $\{\theta^{(t)}\}_{t\geq 0}$ and $\{\tilde{\theta}^{(t)}\}$, respectively. Wang et al. (2008) showed the following lemma.

Lemma 1

$$\lim_{t\to\infty} \frac{||\dot{\theta}^{(t-1)} - \theta^*||}{||\theta^{(t+1)} - \theta^*||} = 0.$$

By Lemma 1, we have $||\dot{\theta}^{(t-1)} - \theta^*|| = o(||\theta^{(t+1)} - \theta^*||)$. Similarly applying the ε-acceleration to the re-starting sequence $\{\tilde{\theta}^{(t)}\}_{t\geq 0}$, we also have $||\dot{\tilde{\theta}}^{(t-1)} - \theta^*|| = o(||\tilde{\theta}^{(t+1)} - \theta^*||)$. That is, the ε-acceleration can speed up the convergence of the sequence $\{\tilde{\theta}^{(t)}\}_{t\geq 0}$.

Next we show that the speed of convergence of $\{\dot{\tilde{\theta}}^{(t)}\}_{t\geq 0}$ is faster than that of $\{\dot{\theta}^{(t)}\}_{t\geq 0}$. We set $\triangle\theta^{(t-1)} = \theta^{(t)} - \theta^{(t-1)}$ and $\eta^{(t)} = [[\triangle\theta^{(t)}]^{-1} - [\triangle\theta^{(t-1)}]^{-1}]^{-1}$. Then we have from Eq. (3)

$$\begin{aligned}\|\dot{\theta}^{(t)} - \theta^*\|^2 &= \|\theta^{(t)} - \theta^* + \eta^{(t)}\|^2 \\ &= \|\theta^{(t)} - \theta^*\|^2 + 2\langle\theta^{(t)} - \theta^*, \eta^{(t)}\rangle + \|\eta^{(t)}\|^2. \end{aligned} \tag{9}$$

Since

$$\triangle\theta^{(t)} = \theta^{(t+1)} - \theta^{(t)} = (\theta^{(t+1)} - \theta^*) - (\theta^{(t)} - \theta^*) = (\lambda - 1)(\theta^{(t)} - \theta^*)$$

for sufficiently large t and $\theta^{(t)}$ in a neighborhood of θ^*, we have

$$\|\triangle\theta^{(t)}\|^2 = (\lambda - 1)^2\|\theta^{(t)} - \theta^*\|^2$$

and

$$\|\triangle\theta^{(t)} - \triangle\theta^{(t-1)}\|^2 = (\lambda - 1)^4\|\theta^{(t-1)} - \theta^*\|^2.$$

We obtain from Wang et al. (2008) and the above equations

$$\begin{aligned}\eta^{(t)} &= \left[\frac{\triangle\theta^{(t)}}{\|\triangle\theta^{(t)}\|^2} - \frac{\triangle\theta^{(t-1)}}{\|\triangle\theta^{(t-1)}\|^2}\right]^{-1} \\ &= \frac{\|\triangle\theta^{(t)}\|^2\|\triangle\theta^{(t-1)}\|^2}{\|\triangle\theta^{(t)} - \triangle\theta^{(t-1)}\|^2}\left[\frac{\triangle\theta^{(t)}}{\|\triangle\theta^{(t)}\|^2} - \frac{\triangle\theta^{(t-1)}}{\|\triangle\theta^{(t-1)}\|^2}\right] \\ &= -\lambda(\theta^{(t-1)} - \theta^*).\end{aligned}$$

Then we have

$$\|\eta^{(t)}\|^2 = \lambda^2\|\theta^{(t-1)} - \theta^*\|^2$$

and

$$\langle\theta^{(t)} - \theta^*, \eta^{(t)}\rangle = -\lambda^2\|\theta^{(t-1)} - \theta^*\|^2.$$

Equation (9) becomes

$$\|\dot{\theta}^{(t)} - \theta^*\|^2 = \|\theta^{(t)} - \theta^*\|^2 - \lambda^2\|\theta^{(t-1)} - \theta^*\|^2.$$

In a similar way, we obtain

$$\|\dot{\tilde{\theta}}^{(t)} - \theta^*\|^2 = \|\tilde{\theta}^{(t)} - \theta^*\|^2 - \lambda^2\|\tilde{\theta}^{(t-1)} - \theta^*\|^2.$$

We have from Eqs. (6) and (7) and inequality (8)

$$\frac{\|\tilde{\theta}^{(t+L)} - \theta^*\|}{\|\theta^{(t+L)} - \theta^*\|} = \frac{\|\tilde{\theta}^{(t+L-1)} - \theta^*\|}{\|\theta^{(t+L-1)} - \theta^*\|} = \cdots = \frac{\|\tilde{\theta}^{(t+1)} - \theta^*\|}{\|\theta^{(t+1)} - \theta^*\|} = \gamma < 1.$$

By the above two equations, we get

$$\begin{aligned}\|\dot{\tilde{\theta}}^{(t+L)} - \theta^*\|^2 &= \|\tilde{\theta}^{(t+L)} - \theta^*\|^2 - \lambda^2\|\tilde{\theta}^{(t+L-1)} - \theta^*\|^2 \\ &= \gamma^2(\|\theta^{(t+L)} - \theta^*\|^2 - \lambda^2\|\theta^{(t+L-1)} - \theta^*\|^2),\end{aligned}$$

and thus obtain

$$\frac{\|\dot{\tilde{\theta}}^{(t+L)} - \theta^*\|^2}{\|\dot{\theta}^{(t+L)} - \theta^*\|^2} = \frac{\|\tilde{\theta}^{(t+L)} - \theta^*\|^2 - \lambda^2\|\tilde{\theta}^{(t+L-1)} - \theta^*\|^2}{\|\theta^{(t+L)} - \theta^*\|^2 - \lambda^2\|\theta^{(t+L-1)} - \theta^*\|^2} = \gamma^2 < 1,$$

which completes the proof of the theorem. □

4 Numerical experiments

In this section, we provide numerical experiments using linear models and Poisson and normal mixture models. Then we investigate how much faster the εR-accelerated EM algorithm converges than the EM and ε-accelerated EM algorithms and compare the performance of the εR-accelerated EM algorithm with that of the AEM algorithm proposed by Jamshidian and Jennrich (1993). The AEM algorithm is a conjugate gradient acceleration of the EM algorithm and is described in "Appendix 3". As the line search algorithm in the AEM algorithm, we use the gold-section method (`R` code is given in Jones et al. (2009)). All computations are performed with the statistical package `R` (R Development Core Team 2013) executing on Intel Core 2 Duo 2.4 GHz with 4 GB of memory. The CPU times (in seconds) are measured by the function `proc.time`.[1] For all experiments, we set $\delta = 10^{-12}$ for convergence of the algorithms, $\delta_{Re} = 1$ and $k = 1$ for the re-starting condition.

[1] Times are typically available to 10 msec.

4.1 Linear models

Consider the following linear model

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{e}, \tag{10}$$

where $\mathbf{y}$ is an $n \times 1$ vector of n observations, $\mathbf{X}$ is an $n \times (p+1)$ known design matrix, β is a vector of $p+1$ fixed parameters, and $\mathbf{e}$ is an $n \times 1$ random vector of errors with $e_j \sim N(0, \sigma^2)$ for $j = 1, \ldots, n$:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \mathbf{e} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}.$$

Thus we have

$$\mathbf{y} \sim N(\mathbf{X}\beta, \sigma^2 \mathbf{I}_n). \tag{11}$$

The analysis of variance and multiple linear regression belong to this model.

We consider the case that $\mathbf{y}$ is missing at random. Then $\mathbf{y}$ is partitioned into the observed part $\mathbf{y}_o$ and the missing part $\mathbf{y}_m$. Then the E-step for the $(t+1)$th estimate of $\mathbf{y}_m$ is given by

$$\mathbf{y}_m^{(t+1)} = \mathbf{E}[\mathbf{y}_m | \beta^{(t)}] = \mathbf{X}_m \beta^{(t)},$$

and the M-step for finding $\beta^{(t+1)}$ is given by

$$\beta^{(t+1)} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}\mathbf{y}^{(t+1)},$$

where $\mathbf{y}^{(t+1)} = (\mathbf{y}_o, \mathbf{y}_m^{(t+1)})$.

4.1.1 Two-way analysis of variance

We consider the two-way analysis of variance (two-way ANOVA) with two factors A and B having levels J and K, respectively. Then the two-way ANOVA model can be written as

$$y_{jk} = \lambda_0 + \lambda_j^A + \lambda_k^B + e_{jk}, \tag{12}$$

where the terms λ_j^A and λ_k^B correspond to main effects of A and B. For the parameters, we impose the sum-to-zero constraints such that

$$\sum_{j=1}^{J} \lambda_j^A = \sum_{k=1}^{K} \lambda_k^B = 0.$$

In this numerical experiment, $\mathbf{y}$ consists of A and B with $J = K = 4$ levels and is generated from the normal distribution $N(0, 100)$. We set the proportion of missing values of $\mathbf{y}$ to 50 %, and then find $\beta = (\lambda_0, \{\lambda_j^A\}_{1\le j\le 4}, \{\lambda_k^B\}_{1\le k\le 4})$ and σ^2. The procedure is replicated 1,000 times.

In order to evaluate the effect of the re-starting procedure, we examine the convergence behavior of the EM, ε-accelerated EM and εR-accelerated EM algorithms measured by $\log_{10}\|\beta^{(t)} - \beta^{(\infty)}\|^2$ (or $\log_{10}\|\dot{\beta}^{(t)} - \beta^{(\infty)}\|^2$), where $\beta^{(\infty)}$ is the MLE founded by the algorithm. Figure 1 illustrates the traces of these algorithms till the convergence of the εR-accelerated EM algorithm attains. The figure shows that the sequence generated by the εR-accelerated EM algorithm converges after 37 iterations with $\delta = 10^{-12}$, while the sequence of the ε-accelerated EM algorithm matches four-digit precision to the MLE. For the experiment, the EM algorithm converges after 217

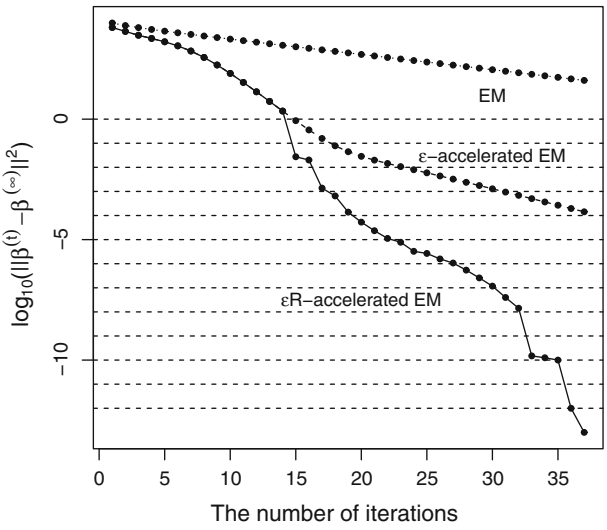

Fig. 1 Convergence behavior of the EM (*dotted line*), ε-accelerated EM (*dashed line*) and εR-accelerated EM (*solid line*) algorithms till the converge of the εR-accelerated EM algorithm attains

Table 1 Summary statistics of the numbers of iterations and CPU times of the EM, ε-accelerated EM (ε), εR-accelerated EM (εR) and AEM (CG) algorithms from 1,000 simulated data for the two-way ANOVA

	The number of iterations				CPU time			
	EM	ε	εR	CG	EM	ε	εR	CG
Min.	35.0	14.0	10.00	7.00	0.1000	0.0400	0.0400	0.0900
1st Qu.	109.0	45.0	29.00	17.00	0.3100	0.1400	0.1300	0.2200
Median	163.5	78.0	41.00	22.00	0.4700	0.2300	0.1600	0.2700
Mean	167.7	72.4	40.38	22.29	0.4814	0.2146	0.1615	0.2726
3rd Qu.	214.2	97.0	51.00	27.00	0.6100	0.2800	0.1900	0.3200
Max.	365.0	119.0	79.00	45.00	1.0300	0.3600	0.2800	0.5600

iterations and the ε-accelerated EM algorithm does after 86 iterations. We can see that the re-starting procedure works effectively to reduce the number of iterations.

Table 1 presents the summary statistics of the numbers of iterations and CPU times of these algorithms. The εR-accelerated EM algorithm requires about a quarter of the number of iterations and about one third of CPU time of the EM algorithm. The AEM algorithm greatly reduces the number of iterations but increases CPU time comparing with the other two acceleration algorithms.

4.1.2 Multiple linear regression

For the explanatory variables that are all continuous, $\mathbf{X}$ has the first column $(1, \ldots, 1)^\top$ corresponding to β_0 and the observed values of explanatory variables as other columns:

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{pmatrix}.$$

In this numerical experiment, we generate $(\mathbf{y}, \mathbf{X})$ for $n = 100$ and $p = 30$ from the multivariate normal distribution $N(\mathbf{0}, \Sigma_p)$, where Σ_p is a $p \times p$ covariance matrix and is randomly chosen using a Wishart random number generator. The procedure is replicated 1,000 times for every algorithm.

We show the results in Table 2. The ε-accelerated EM algorithm converges about 1.7 times faster than the EM algorithm in both of the number of iterations and CPU time. The εR-accelerated EM algorithm furthermore speeds up the convergence of the EM algorithm. The algorithm requires the number of iterations about 3.5 times smaller and CPU time about 2.8 times shorter than those of the EM algorithm. The AEM algorithm reduces the number of iterations as well as the εR-accelerated EM algorithm but not CPU time. The line search algorithm for finding a suitable α for each iteration may take a longer computation time.

Table 2 Summary statistics of the numbers of iterations and CPU times of the EM, ε-accelerated EM (ε), εR-accelerated EM (εR) and AEM (CG) algorithms from 1,000 simulated data for the multiple regression model

	The number of iterations				CPU time			
	EM	ε	εR	CG	EM	ε	εR	CG
Min.	91.0	58.0	23.00	25.00	0.820	0.530	0.3300	0.5300
1st Qu.	155.8	93.0	43.00	35.00	1.400	0.850	0.5200	0.6900
Median	181.0	108.0	52.00	39.00	1.630	0.980	0.5900	0.7600
Mean	189.9	110.9	53.95	42.84	1.711	1.007	0.6128	0.7775
3rd Qu.	218.0	125.0	63.00	47.00	1.950	1.140	0.7000	0.8400
Max.	432.0	217.0	136.00	140.00	3.850	1.940	1.3400	1.5800

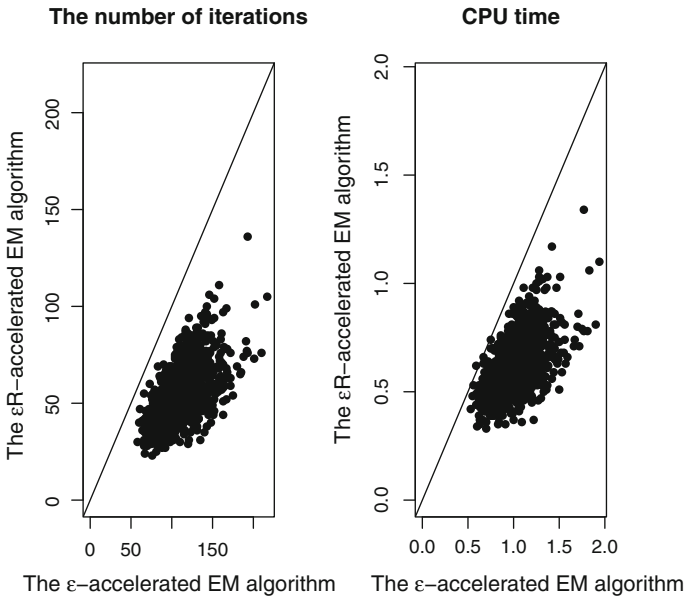

Fig. 2 Scatter plots of the εR-accelerated EM algorithm by the ε-accelerated EM algorithm for the number of iterations and CPU time from 1,000 initial values for the multiple regression model

We compare the performance of the εR-accelerated EM algorithm with that of the ε-accelerated EM algorithm. Figure 2 presents the scatter plots of the εR-accelerated EM algorithm by the ε-accelerated EM algorithm for the number of iterations and CPU time. We can see from the figure that the εR-accelerated EM algorithm converges in a smaller number of iterations than the ε-accelerated EM algorithm and well accelerates the convergence of $\{\beta^{(t)}\}_{t\geq 0}$ when the ε-accelerated EM algorithm requires more iterations and longer CPU time.

4.2 Linear mixed model

The linear mixed model is given by

$$\mathbf{y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{u}_i + \mathbf{e}_i,$$

for $i = 1, \ldots, m$, where $\mathbf{y}_i$ is an $n_i \times 1$ observed vector, $\mathbf{X}_i$ and $\mathbf{Z}_i$ are known $n_i \times p$ and $n_i \times q$ design matrices corresponding to the $p \times 1$ fixed effects vector β to be estimated and the $q \times 1$ random effects vector $\mathbf{u}_i$. We assume that $\mathbf{e}_i$ and $\mathbf{u}_i$ are independent of each other, and $\mathbf{e}_i$ is distributed $N(\mathbf{0}, \sigma_0^2\mathbf{R}_i)$ and $\mathbf{u}_i$ is $N(\mathbf{0}, \mathbf{D})$, where $\mathbf{R}_i$ is a known $n_i \times n_i$ matrix and $\mathbf{D}$ is an unknown $q \times q$ positive definite covariance matrix. We treat $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_m)$ as missing data. The log-likelihood of $\theta = (\beta, \sigma_0^2, \mathbf{D})$ given $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_m)$ is

$$L_o(\theta) = -\frac{1}{2}n\log(2\pi) - \frac{1}{2}\sum_{i=1}^{m}\left\{\log|\mathbf{V}_i| + (\mathbf{y}_i - \mathbf{X}_i\beta)^\top\mathbf{V}_i^{-1}(\mathbf{y}_i - \mathbf{X}_i\beta)\right\},$$

where $n = \sum_{i=1}^{m} n_i$ and $\mathbf{V}_i = \mathbf{Z}_i \mathbf{D} \mathbf{Z}_i^\top + \sigma_0^2 \mathbf{R}_i$. We denote the complete data vector by $\mathbf{w} = (\mathbf{w}_1^\top, \ldots, \mathbf{w}_m^\top)^\top$, where $\mathbf{w}_i = (\mathbf{y}_i^\top, \mathbf{u}_i^\top)^\top$. The log-likelihood function of θ for $\mathbf{w}$ is

$$L_c(\theta) = -\frac{1}{2} n \log(2\pi) - \frac{1}{2} \sum_{i=1}^{m} \left\{ \log |\Sigma_i| + (\mathbf{w}_i - \mu_i)^\top \Sigma_i^{-1} (\mathbf{w}_i - \mu_i) \right\},$$

where

$$\mu_i = \begin{pmatrix} \mathbf{X}_i \beta \\ \mathbf{0} \end{pmatrix}, \qquad \Sigma_i = \begin{pmatrix} \mathbf{Z}_i \mathbf{D} \mathbf{Z}_i^\top + \sigma_0^2 \mathbf{R}_i & \mathbf{Z}_i \mathbf{D} \\ \mathbf{D} \mathbf{Z}_i^\top & \mathbf{D} \end{pmatrix}.$$

At the $(t+1)$th iteration, the E-step computes

$$\mathbf{u}_i^{(t+1)} = \mathbf{E}[\mathbf{u}_i | \theta^{(t)}] = (\mathbf{Z}_i^\top \mathbf{R}_i^{-1} \mathbf{Z}_i + \sigma_0^{2(t)} (\mathbf{D}^{(t)})^{-1})^{-1} \mathbf{Z}_i^\top \mathbf{R}_i^{-1} (\mathbf{y}_i - \mathbf{X}_i \beta^{(t)})$$

and

$$\mathbf{S}_i^{(t+1)} = \mathbf{E}[\mathbf{u}_i \mathbf{u}_i^\top | \theta^{(t)}] = (\mathbf{Z}_i^\top \mathbf{R}_i^{-1} \mathbf{Z}_i / \sigma_0^{2(t)} + (\mathbf{D}^{(t)})^{-1})^{-1} + \mathbf{u}_i^{(t+1)} \mathbf{u}_i^{(t+1)\top},$$

and the M-step updates the estimate of θ by

$$\begin{aligned}
\beta^{(t+1)} &= \left(\sum_{i=1}^{m} \mathbf{X}_i^\top \mathbf{R}_i^{-1} \mathbf{X}_i \right)^{-1} \sum_{i=1}^{m} \mathbf{X}_i^\top \mathbf{R}_i^{-1} (\mathbf{y}_i - \mathbf{Z}_i \mathbf{u}_i^{(t+1)}), \\
\sigma_0^{2(t+1)} &= \frac{1}{n} \sum_{i=1}^{m} (\mathbf{y}_i - \mathbf{X}_i \beta^{(t+1)} - \mathbf{Z}_i \mathbf{u}_i^{(t+1)})^\top \mathbf{R}_i^{-1} (\mathbf{y}_i - \mathbf{X}_i \beta^{(t+1)} - \mathbf{Z}_i \mathbf{u}_i^{(t+1)}) \\
&\quad + \frac{1}{n} \sum_{i=1}^{m} \mathrm{tr} \left[\mathbf{Z}_i^\top \mathbf{R}_i^{-1} \mathbf{Z}_i (\mathbf{Z}_i^\top \mathbf{R}_i^{-1} \mathbf{Z}_i / \sigma_0^{2(t)} + (\mathbf{D}^{(t)})^{-1})^{-1} \right], \\
\mathbf{D}^{(t+1)} &= \frac{1}{n} \sum_{i=1}^{m} \mathbf{S}_i^{(t+1)}.
\end{aligned}$$

The data shown in Table 3 are the average daily gain of two pigs of each litter in pounds (Snedecor and Cochran 1967). The experiment was designed so that each sire ($i = 1, \ldots, 5$) is mated to a random group of dams ($j = 1, 2$) and each mating producing a litter in which two pigs are chosen ($k = 1, 2$). The gain in weight of those two pigs is the criterion. The considered model is the linear mixed model with one random effect

$$\mathbf{y}_{ij} = \mathbf{X}_{ij} \beta + \mathbf{Z}_{ij} u_{ij} + \mathbf{e}_{ij},$$

where $\mathbf{y}_{ij}$ is the observed average gain of two pigs by day in pounds produced by the ith sire and jth dam, $\beta = (\beta_0, \beta_1, \ldots, \beta_4)^\top$ is a sire effect, u_{ij} is a random effect associated with the ith sire and the jth dam and $\mathbf{e}_{ij}$ is a random term. We assume that $\mathbf{e}_{ij}$ and

Table 3 Average daily gain of two pigs of each litter in pounds (Snedecor and Cochran 1967)

Sire	Dam	Gain
1	1	2.77
1	1	2.38
1	2	2.58
1	2	2.94
2	1	2.28
2	1	2.22
2	2	3.01
2	2	2.61
3	1	2.36
3	1	2.71
3	2	2.72
3	2	2.74
4	1	2.87
4	1	2.46
4	2	2.31
4	2	2.24
5	1	2.74
5	1	2.56
5	2	2.50
5	2	2.48

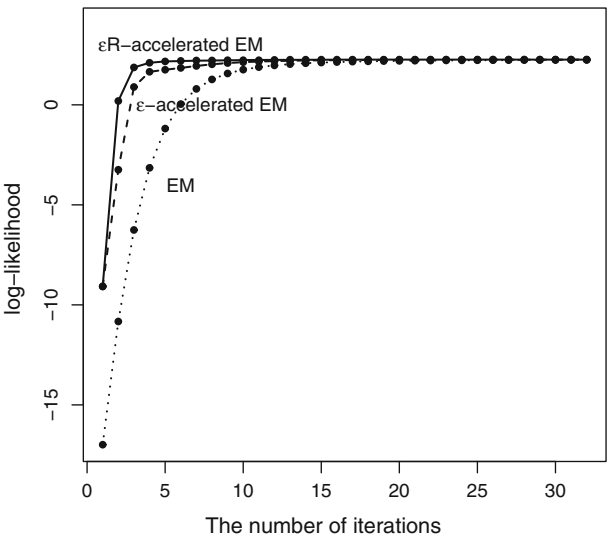

Fig. 3 Traces of the log-likelihoods of the EM (*dotted line*), ε-accelerated EM (*dashed line*) and εR-accelerated EM (*solid line*) algorithms till the converge of the εR-accelerated EM algorithm attains

Table 4 Summary statistics of the numbers of iterations and CPU times of the EM, ε-accelerated EM (ε) and εR-accelerated EM (εR) algorithms from 1,000 simulated data for the linear mixed model

	The number of iterations			CPU time		
	EM	ε	εR	EM	ε	εR
Min.	100.0	30.0	7	0.0400	0.0200	0.0100
1st Qu.	159.0	89.0	31	0.0900	0.0500	0.0300
Median	160.0	90.0	32	0.0900	0.0600	0.0300
Mean	159.1	89.7	31	0.0952	0.0552	0.0376
3rd Qu.	160.0	91.0	32	0.1000	0.0600	0.0500
Max.	315.0	245.0	113	0.1800	0.1400	0.1100

u_{ij} are distributed $N(\mathbf{0}, \sigma_0^2\mathbf{I}_2)$ and $N(0, \sigma_1^2)$, respectively. Thus the parameters to be estimated are $\theta = (\beta, \sigma_0^2, \sigma_1^2)$. We start with $\sigma_0^{2(0)}$ and $\sigma_1^{2(0)}$ from $U(0.001, 1.5)$ and with $\beta^{(0)}$ fixed to the value of mean vector $(2.57, 0.098, -0.040, 0.063, -0.010)$.

Figure 3 presents the traces of the log-likelihoods of the EM, ε-accelerated EM and εR-accelerated EM algorithms till the converge of the εR-accelerated EM algorithm attains. The figure indicates that the εR-accelerated EM and ε-accelerated EM algorithms increase the log-likelihood $L_o(\theta)$ much more than the EM algorithm. It is obvious the faster convergence of these acceleration algorithms over the EM algorithm. Table 4 is the results on 1,000 replications. The εR-accelerated EM algorithm is about 5 times faster than the EM algorithm and also is about 3 times faster than the ε-accelerated EM algorithm in the number of iterations. We see that the εR-accelerated EM algorithm can greatly improve the speed of convergence of the ε-accelerated EM algorithm.

4.3 Mixture models

Mixture models become increasingly popular due to the modeling flexibility and are one of the most interesting application areas of the EM algorithm. McLachlan and Peel (2000) provided a comprehensive book of the theory and applications of mixture models.

We consider two-component Poisson and normal mixture models. A two-component mixture model for density of an observation $\mathbf{y}$ has the form

$$f(\mathbf{y}|\theta) = \lambda f_1(\mathbf{y}|\theta_1) + (1-\lambda) f_2(\mathbf{y}|\theta_2),$$

where λ is an unknown mixing proportion ($0 < \lambda < 1$), and $f_1(\mathbf{y}|\theta_1)$ and $f_2(\mathbf{y}|\theta_2)$ are the component density functions with parameters θ_1 and θ_2, respectively. The log-likelihood function of $\theta = (\lambda, \theta_1, \theta_2)$ given observations $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ is

$$L_o(\theta) = \sum_{i=1}^{n} \log \{\lambda f_1(\mathbf{y}_i|\theta_1) + (1-\lambda) f_2(\mathbf{y}_i|\theta_2)\}.$$

In the setting of the EM algorithm, we regard $\mathbf{y}$ as incomplete data and introduce latent variables $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_n)$, where $\mathbf{z}_i = (z_{i1}, z_{i2})$ is a binary vector defined as $z_{ik} = 1$ if observation $\mathbf{y}_i$ arises from the kth component of the mixture model, and $z_{ik} = 0$ otherwise. Then the complete data $\mathbf{x}$ are given by $\mathbf{x} = (\mathbf{y}, \mathbf{z})^\top$. The density of $\mathbf{x}$ can be written as

$$f(\mathbf{x}|\theta) = \prod_{i=1}^{n} \prod_{k=1,2} \{p(z_{ik}|\lambda) f_k(\mathbf{y}_i|\theta_k)\}^{z_{ik}},$$

where $p(z_{i1} = 1|\lambda) = \lambda$ and $p(z_{i2} = 1|\lambda) = 1 - \lambda$. The log-likelihood function of θ given $\mathbf{x}$ is

$$L_c(\theta) = \sum_{i=1}^{n} z_{i1} \log \lambda f_1(\mathbf{y}_i|\theta_1) + \sum_{i=1}^{n} z_{i2} \log(1-\lambda) f_2(\mathbf{y}_i|\theta_2).$$

Then, the E-step computes the conditional expectations $w(\mathbf{y}_i, \theta')$ of z_{i1} given θ' and $\mathbf{y}_i$ for $i = 1, \ldots, n$ and the M-step finds the MLEs of $\theta = (\lambda, \theta_1, \theta_2)$.

4.3.1 Poisson mixture model

We consider a mixture model of two Poisson distributions with $\theta = (\lambda, \mu_1, \mu_2)$ given by

$$f(y|\theta) = \lambda \exp[-\mu_1] \frac{\mu_1^y}{y!} + (1-\lambda) \exp[-\mu_2] \frac{\mu_2^y}{y!}.$$

The log-likelihood function of θ given observed data $y = (y_1, \ldots, y_n)$ is

$$L_o(\theta) = \sum_{i=1}^{n} \log \left\{ \lambda \exp[-\mu_1] \frac{\mu_1^{y_i}}{y_i!} + (1-\lambda) \exp[-\mu_2] \frac{\mu_2^{y_i}}{y_i!} \right\}. \tag{13}$$

Let c_j be the frequency of $Y = y_i$ for $j = 0, 1, 2, \ldots$. Then Eq. (13) can be rewritten as

$$L_o(\theta) = \sum_{j=0}^{\infty} c_j \log \left\{ \lambda \exp[-\mu_1] \frac{\mu_1^j}{j!} + (1-\lambda) \exp[-\mu_2] \frac{\mu_2^j}{j!} \right\}.$$

Then the EM estimates for the $(t+1)$th iteration are given by

$$\lambda^{(t+1)} = \sum_{j=0}^{\infty} c_j w(j, \theta^{(t)}) \Big/ \sum_{j=0}^{\infty} c_j,$$

$$\mu_1^{(t+1)} = \sum_{j=0}^{\infty} j c_j w(j, \theta^{(t)}) \Big/ \sum_{j=0}^{\infty} c_j w(j, \theta^{(t)}).,$$
$$\mu_2^{(t+1)} = \sum_{j=0}^{\infty} j c_j (1 - w(j, \theta^{(t)})) \Big/ \sum_{j=0}^{\infty} c_j (1 - w(j, \theta^{(t)})) \ ,$$

where

$$w(j, \theta) = \frac{\lambda \exp[-\mu_1]\mu_1^j / j!}{\lambda \exp[-\mu_1]\mu_1^j / j! + (1 - \lambda) \exp[-\mu_2]\mu_2^j / j!}.$$

In this numerical experiment, **y** is generated from the mixture of two Poisson distributions with $\theta = (\lambda, \mu_1, \mu_2) = (0.4, 3, 5)$ and is given in Table 5. Then we generate 1,000 initial values of λ from the uniform distribution $U(0.05, 0.95)$, and those of μ_1 and μ_2 from $U(1, 15)$ under restriction $\mu_1^{(0)} < \mu_2^{(0)}$.

We compare the MLEs θ^{MLE} from the EM and three acceleration algorithms with the true parameter values $\theta^{true} = (0.4, 3, 5)$. The mean values of θ^{MLE} from the EM algorithms are $(0.268, 3.05, 4.78)$. Each acceleration algorithm can also find the same values as those from the EM algorithm. The standard errors of θ^{MLE} from each algorithm are less than 10^{-4}. Thus it can be seen that the MLEs of (μ_1, μ_2) are closed to the true values. The values of $L_o(\theta^{MLE})$ from the algorithms are -1091.30 and are larger than $L_o(\theta^{true}) = -1092.74$ that is the value of the log-likelihood of θ^{true}.

We see from the mean values in Table 6 that the ε-accelerated EM algorithm reduces 1/1.6 of the number of iterations and 1/1.3 of CPU time of the EM algorithm. The AEM algorithm converges in a substantially smaller number of iterations than those of the other acceleration algorithms but takes longer CPU time. We note that the εR-accelerated EM algorithm considerably converges faster than the EM algorithm in both of the number of iterations and CPU time. As shown in the mean values of the table, the algorithm requires about 1/10 of the number of iterations and about 1/7 of CPU time of the EM algorithm. Figure 4 is the scatter plots of the ε-accelerated EM and εR-accelerated EM algorithms by the EM algorithm for the number of iterations and CPU time. The figure illustrates that the ε-accelerated EM algorithm increases linearly with the number of iterations and CPU time as the EM algorithm takes a larger number of iterations, while there is little variation in those for the εR-accelerated EM algorithm.

Table 5 Random data generated from the Poisson mixture distribution with parameter $(\lambda, \mu_1, \mu_2) = (0.4, 3, 5)$

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13
c_j	12	40	72	92	87	80	53	29	15	11	6	2	0	1

Table 6 Summary statistics of the numbers of iterations and CPU times of the EM, ε-accelerated EM (ε), εR-accelerated EM (εR) and AEM (CG) algorithms from 1,000 simulated data for the Poisson mixture model

	The number of iterations				CPU time			
	EM	ε	εR	CG	EM	ε	εR	CG
Min.	1,800	291	63.0	15.00	0.0800	0.0200	0.0100	0.0100
1st Qu.	3,772	2,263	152.0	34.00	0.1900	0.1400	0.0200	0.0800
Median	4,158	2,649	374.0	42.00	0.2200	0.1700	0.0300	0.1100
Mean	3,987	2,479	390.2	44.85	0.2086	0.1606	0.0295	0.1138
3rd Qu.	4,349	2,840	541.0	52.00	0.2300	0.1900	0.0400	0.1400
Max.	4,509	3,000	1174.0	100.00	0.2800	0.2400	0.1000	0.3000

4.3.2 Univariate normal mixture model

Next we consider a mixture model of two univariate normal distributions. Let $\phi_k(y|\mu_k, \sigma_k^2)$ be the kth normal density function with a mean μ_k and a variance σ_k^2. The density of an observation y is

$$f(y|\theta) = \lambda\phi_1(y|\mu_1, \sigma_1^2) + (1-\lambda)\phi_2(y|\mu_2, \sigma_2^2),$$

where $\theta = (\lambda, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$. Then the E-step for the $(t+1)$th iteration compute the conditional expectations $w(y_i, \theta) = \lambda\phi_1(y_i|\mu_1, \sigma_1^2)/f(y_i|\theta)$ $(i = 1, \ldots, n)$ and the M-step estimates $\theta^{(t+1)}$ by

$$\begin{aligned}
\lambda^{(t+1)} &= \sum_{i=1}^{n} w(y_i, \theta^{(t)}) \Big/ n, \\
\mu_1^{(t+1)} &= \sum_{i=1}^{n} w(y_i, \theta^{(t)}) y_i \Big/ \sum_{i=1}^{n} w(y_i, \theta^{(t)}) \ , \\
\mu_2^{(t+1)} &= \sum_{i=1}^{n} (1 - w(y_i, \theta^{(t)})) y_i \Big/ \sum_{i=1}^{n} (1 - w(y_i, \theta^{(t)})) \ , \\
\sigma_1^{2(t+1)} &= \sum_{i=1}^{n} w(y_i, \theta^{(t)})(y_i - \mu_1^{(t+1)})^2 \Big/ \sum_{i=1}^{n} w(y_i, \theta^{(t)}), \\
\sigma_2^{2(t+1)} &= \sum_{i=1}^{n} (1 - w(y_i, \theta^{(t)}))(y_i - \mu_2^{(t+1)})^2 \Big/ \sum_{i=1}^{n} (1 - w(y_i, \theta^{(t)})) \ .
\end{aligned}$$

We generate $\mathbf{y}$ of size $n = 1{,}000$ from the mixture of two univariate normal distributions with $\theta = (\lambda, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = (0.6, -1, 3, 8, 6)$. In setting 1,000 initial values of θ, $\lambda^{(0)}$ is generated from the beta distribution $Be(100\lambda, 100(1-\lambda))$, $\mu_k^{(0)}$ is from $N(\mu_k, 3)$ and $\sigma_k^{2(0)}$ is from $U(\sigma_k^2/2, 2\sigma_k^2)$ for $k = 1, 2$. We also impose $\mu_1^{(0)} < \mu_2^{(0)}$.

Table 7 Summary statistics of the numbers of iterations and CPU times of the EM, ε-accelerated EM (ε), εR-accelerated EM (εR) and AEM (CG) algorithms from 1,000 simulated data for the univariate normal mixture model

	The number of iterations				CPU time			
	EM	ε	εR	CG	EM	ε	εR	CG
Min.	336	139	29.0	22.00	0.140	0.060	0.030	0.410
1st Qu.	22,950	5,256	687.8	67.00	9.328	2.230	0.300	1.370
Median	26,910	9,413	770.0	90.00	10.940	3.980	0.340	1.890
Mean	25,950	8,800	833.9	98.05	10.540	3.721	0.368	2.051
3rd Qu.	30,140	12,260	886.2	118.00	12.210	5.170	0.390	2.502
Max.	40,780	21,960	3515.0	331.00	16.550	9.220	1.500	7.070

We obtain from 1,000 simulation runs of the EM and its acceleration algorithms that the mean values of θ^{MLE} are $(0.599, -0.43, 2.59.8.96, 7.27)$, and the standard error for the MLE of μ is 0.001 and the errors of other parameters are 0.01. We see that the MLEs of (λ, μ_2) are closed to the true values and, for other parameters, the errors between the MLEs and the true values are about ± 1. We obtain $L_o(\theta^{true}) = -2599.45$ and $L_o(\theta^{MLE}) = -2596.74$ for all θ^{MLE} from the algorithms.

Table 7 reports the results of the number of iterations and CPU time. The EM algorithm converges very slowly and takes more than 20000 iterations for convergence. The number of iterations and CPU time of the ε-accelerated EM algorithm reduces about one third of those of the EM algorithm. The AEM algorithm takes much fewer iterations than the ε-accelerated EM and εR-accelerated EM algorithms and its CPU time is shorter than that of the ε-accelerated EM algorithm. The εR-accelerated EM algorithm converges 30 times faster than the EM algorithm for both the number of iterations and CPU time. In terms of the performance of acceleration of the EM algorithm, the εR-accelerated EM algorithm is clearly the best. Figure 5 shows the scatter plots of the ε-accelerated EM and εR-accelerated EM algorithms by the EM algorithm for the number of iterations and CPU time. We obtain from the figure the same result as in the experiment of the Poisson mixture model. They show that the re-starting procedure can work effectively to reduce the number of iterations and CPU time for slowly convergent EM sequences.

4.3.3 Bivariate normal mixture model

Finally we consider a mixture model of two bivariate normal distributions. Let $\phi_k(\mathbf{y}|\mu_k, \Sigma_k)$ be the kth bivariate normal density function with a mean vector μ_k and a covariance matrix Σ_k. For the two-component mixture model, the density of an observation $\mathbf{y}$ is given by

$$f(\mathbf{y}|\theta) = \lambda\phi_1(\mathbf{y}|\mu_1, \Sigma_1) + (1-\lambda)\phi_2(\mathbf{y}|\mu_2, \Sigma_2),$$

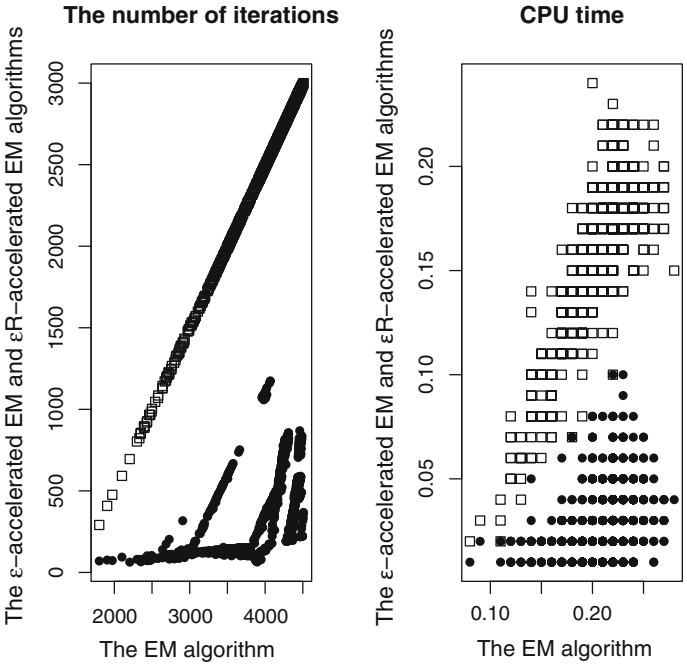

Fig. 4 Scatter plots of the ε-accelerated EM (*square*) and εR-accelerated EM (*circle*) algorithms by the EM algorithm for the number of iterations and CPU time from 1,000 initial values for the Poisson mixture model

where $\theta = (\lambda, \mu_1, \mu_2, \Sigma_1, \Sigma_2)$. Then we update the EM estimates for the $(t+1)$th iteration by

$$\lambda^{(t+1)} = \sum_{i=1}^{n} w(\mathbf{y}_i, \theta^{(t)}) \Big/ n,$$

$$\mu_1^{(t+1)} = \sum_{i=1}^{n} w(\mathbf{y}_i, \theta^{(t)})\mathbf{y}_i \Big/ \sum_{i=1}^{n} w(\mathbf{y}_i, \theta^{(t)}) \ ,$$

$$\mu_2^{(t+1)} = \sum_{i=1}^{n} (1 - w(\mathbf{y}_i, \theta^{(t)}))\mathbf{y}_i \Big/ \sum_{i=1}^{n} (1 - w(\mathbf{y}_i, \theta^{(t)})) \ ,$$

$$\Sigma_1^{(t+1)} = \sum_{i=1}^{n} w(\mathbf{y}_i, \theta^{(t)})(\mathbf{y}_i - \mu_1^{(t+1)})(\mathbf{y}_i - \mu_1^{(t+1)})^\top \Big/ \sum_{i=1}^{n} w(\mathbf{y}_i, \theta^{(t)})$$

$$\Sigma_2^{(t+1)} = \sum_{i=1}^{n} (1 - w(\mathbf{y}_i, \theta^{(t)}))(\mathbf{y}_i - \mu_2^{(t+1)})(\mathbf{y}_i - \mu_2^{(t+1)})^\top \Big/ \sum_{i=1}^{n} (1 - w(\mathbf{y}_i, \theta^{(t)})) \ ,$$

where $w(\mathbf{y}_i, \theta) = \lambda\phi_1(\mathbf{y}_i|\mu_1, \Sigma_1)/f(\mathbf{y}_i|\theta)$.

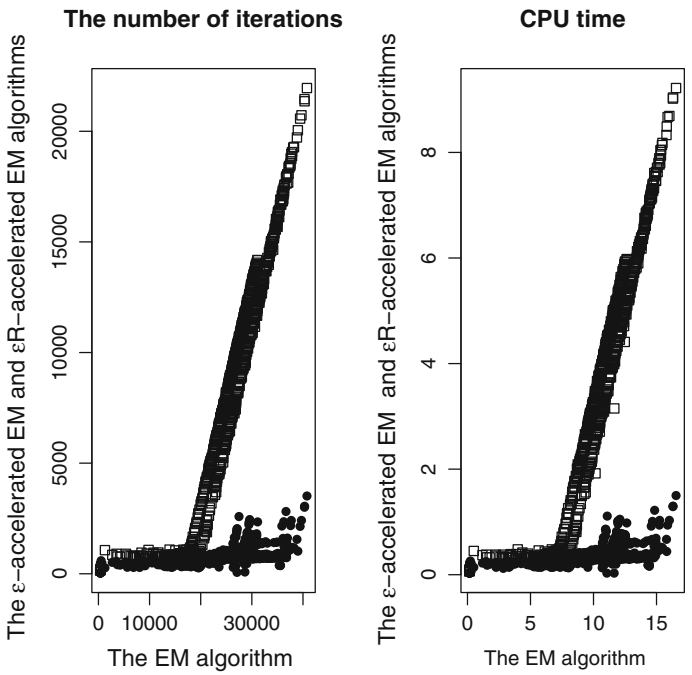

Fig. 5 Scatter plots of the ε-accelerated EM (*square*) and εR-accelerated EM (*circle*) algorithms by the EM algorithm for the number of iterations and CPU time from 1,000 initial values for the univariate normal mixture model

In this numerical experiment, we generate $\mathbf{y}$ of size $n = 1{,}000$ from the mixture of two bivariate normal distributions with

$$\lambda = 0.60, \mu_1 = \begin{pmatrix} -1 \\ -2 \end{pmatrix}, \mu_2 = \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 8 & 2 \\ 2 & 16 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 6 & 2 \\ 2 & 12 \end{pmatrix}.$$

In choosing 1,000 initial values of θ, $\lambda^{(0)}$ is generated from $Be(100\lambda, 100(1-\lambda))$, $\mu_k^{(0)}$ is from the bivariate normal distribution $N_2(\mu_k, \mathrm{diag}(3))$ and $\Sigma_k^{(0)}$ is from the Wishart distribution $W_2(\Sigma_k)$ for $k = 1, 2$.

Table 8 shows the mean values of the MLEs from the EM algorithm and their standard errors. The values from the ε-accelerated and εR-accelerated EM algorithms are also same in the table. We have the similar results for the errors between the MLEs and the true values but the larger standard errors in the experiments of the Poisson and univariate mixture models. The values of $L_o(\theta^{MLE})$ from the algorithms are -5350.29, while $L_o(\theta^{true}) = -5352.81$.

Table 9 indicates that the EM algorithm requires many iterations and takes long computation time for each convergence, and thus it is very valuable to accelerate the convergence of the algorithm. The ε-accelerated EM algorithm takes a half of the number of iterations and CPU time of the EM algorithm. We see that the εR-accelerated EM algorithm greatly reduce the number of iterations and CPU time and

Table 8 The mean values of the MLEs of the EM algorithm from 1,000 simulated data for the bivariate normal mixture model

	λ	μ_1		μ_2		
True value	0.60	−1	−2	3	0	
MLE	0.547	−0.35	−1.35	1.85	−0.48	
	(0.005)	(0.05)	(0.03)	(0.05)	(0.05)	
	Σ_1			Σ_2		
True value	8	2	16	6	2	12
MLE	7.65	3.37	16.26	6.77	2.67	13.02
	(0.05)	(0.05)	(0.13)	(0.06)	(0.07)	(0.11)

The values in parenthesis are the standard errors of the MLEs of the EM algorithm

Table 9 Summary statistics of the numbers of iterations and CPU times of the EM, ε-accelerated EM (ε) and εR-accelerated EM (εR) algorithms from 1,000 simulated data for the bivariate normal mixture model

	The number of iterations			CPU time		
	EM	ε	εR	EM	ε	εR
Min.	31	21	26.0	0.040	0.030	0.040
1st Qu.	5,874	2,185	366.0	9.098	3.365	0.590
Median	7,150	3,478	451.5	11.040	5.320	0.720
Mean	6,552	3,261	534.3	10.120	5.004	0.847
3rd Qu.	8,046	4,351	640.2	12.410	6.660	1.010
Max.	10,400	6,705	2,624.0	15.940	11.000	4.100

Table 10 The total numbers of iterations and total CPU times of the EM, ε-accelerated EM (ε) and εR-accelerated EM (εR) algorithms from 1,000 simulated data for the bivariate normal mixture model

	The number of iterations			CPU time		
	EM	ε	εR	EM	ε	εR
Total	655,1682	326,0672	534,348	10,119.62	5,004.50	846.50

its speed of convergence is 12 times faster than that of the EM algorithm. The results illustrate that the εR-accelerated EM algorithm can greatly improve the computational efficiency of the EM algorithm more than the ε-accelerated EM algorithm.

Table 10 gives the total number of iterations and total CPU time for each algorithm. The EM algorithm requires 65×10^5 iterations and its computation time is 2.8 h, while the εR-accelerated EM only takes 5.3×10^5 iterations and 14 min and reduces 1/12 of those of the EM algorithm. The choice of initial values is of great importance for finding the highest likelihood in mixture models due to the local convergence of the

EM algorithm. Several different initial values are employed to ensure that the global maximum is obtained, see Biernacki et al. (2003) and Karlis and Xekalaki (2003). The results show the possibility that the εR-accelerated EM algorithm greatly contributes to shorten the total number of iterations and the CPU time required.

5 Concluding remarks

In this paper, we provided the εR-accelerated EM algorithm for increasing the speed of convergence of the ε-accelerated EM algorithm by embedding the re-starting procedure in the ε-acceleration step. The re-starting step is a very simple rule, and moreover is performed with increasing a little bit of computation time for each iteration. When the re-starting step finds $\dot{\theta}^{(t-1)}$ satisfying that $L_o(M(\dot{\theta}^{(t-1)})) > L_o(\theta^{(t+1)})$ and $\|\dot{\theta}^{(t-1)} - \dot{\theta}^{(t-2)}\|^2$ is smaller than a threshold, we re-start the EM iterations using $M(\dot{\theta}^{(t-1)})$. We showed under some conditions that the εR-accelerated EM algorithm accelerates the convergence of the EM sequence more than the ε-accelerated EM algorithm.

Numerical experiments demonstrate that the εR-accelerated EM algorithm generates a faster convergent sequence than the ε-accelerated EM algorithm. We see that the speed of convergence of the εR-accelerated EM algorithm is much faster than that of the ε-accelerated EM algorithm when the EM algorithm requires many iterations for convergence. Then the re-starting procedure can work effectively to reduce greatly the number of iterations and computation time of the ε-accelerated EM algorithm. We also compared the performance of the εR-accelerated EM and AEM algorithms. Although the εR-accelerated EM algorithm requires a larger number of iterations than the AEM algorithm, its CPU time is much shorter than that of the AEM algorithm. On the other hand, the AEM algorithm needs to evaluate the gradient of likelihood and find α using a line search algorithm, while the εR-accelerated EM algorithm does not. Thus the εR-accelerated EM algorithm has more advantageous than the AEM algorithm in terms of the computational efficiency.

Mixture models are increasingly interest and popularity with the numerous developments and the frequent applications, see Lee and Scott (2012), Pynea et al. (2009), Lee et al. (2011) and Lin (2009). The EM algorithm is largely used for the maximum likelihood estimation of mixture models but its convergence tends to be slow. Improvement of convergence of the EM algorithm is an important topic. The results from the experiments show that the εR-accelerated EM algorithm is useful due to its fast speed of convergence.

Acknowledgments The authors would like to thank the editor and two referees for their valuable comments and helpful suggestions. This research is supported by the Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for Scientific Research (C), No. 24500353.

Appendix 1: The vector ε algorithm

Let $\theta^{(t)}$ denote a vector of dimensionality d that converges to a vector $\theta^{(\infty)}$ as $t \to \infty$. Let the inverse $[\theta]^{-1}$ of a vector θ be defined by

$$[\theta]^{-1} = \frac{\theta}{\|\theta\|^2},$$

where $||\theta||$ is the Euclidean norm of θ.

In general, the vector ε algorithm for a sequence $\{\theta^{(t)}\}_{t\geq 0}$ starts with

$$\varepsilon^{(t,-1)} = 0, \qquad \varepsilon^{(t,0)} = \theta^{(t)},$$

and then generates a vector $\varepsilon^{(t,k+1)}$ by

$$\varepsilon^{(t,k+1)} = \varepsilon^{(t+1,k-1)} + \left[\varepsilon^{(t+1,k)} - \varepsilon^{(t,k)}\right]^{-1}, \qquad k = 0, 1, 2, \ldots. \tag{14}$$

For practical implementation, we apply the vector ε algorithm for $k = 1$ to accelerate the convergence of $\{\theta^{(t)}\}_{t\geq 0}$. From Eq. (14), we have

$$\begin{aligned}
\varepsilon^{(t,2)} &= \varepsilon^{(t+1,0)} + \left[\varepsilon^{(t+1,1)} - \varepsilon^{(t,1)}\right]^{-1} \quad \text{for } k = 1,\\
\varepsilon^{(t,1)} &= \varepsilon^{(t+1,-1)} + \left[\varepsilon^{(t+1,0)} - \varepsilon^{(t,0)}\right]^{-1} = \left[\varepsilon^{(t+1,0)} - \varepsilon^{(t,0)}\right]^{-1} \quad \text{for } k = 0.
\end{aligned}$$

Then the vector $\varepsilon^{(t,2)}$ becomes as follows:

$$\begin{aligned}
\varepsilon^{(t,2)} &= \varepsilon^{(t+1,0)} + \left[\left[\varepsilon^{(t,0)} - \varepsilon^{(t+1,0)}\right]^{-1} + \left[\varepsilon^{(t+2,0)} - \varepsilon^{(t+1,0)}\right]^{-1}\right]^{-1}\\
&= \theta^{(t+1)} + \left[\left[\theta^{(t)} - \theta^{(t+1)}\right]^{-1} + \left[\theta^{(t+2)} - \theta^{(t+1)}\right]^{-1}\right]^{-1}.
\end{aligned}$$

Appendix 2: Pseudo-code of the εR-accelerated EM algorithm

Initialization

We set the initial value of the EM step θ_0, the desired precision δ, the threshold $\delta_{Re}(> \delta)$ and the size of decrement 10^{-k} and determine the maximum number of iterations ($itrmax$).

Iterations

$\theta_1 \leftarrow M(\theta_0)$
$\dot{\theta}_{old} \leftarrow \theta_1$
$itr \leftarrow 0$
repeat
  $itr \leftarrow itr + 1$
  $\theta_2 \leftarrow M(\theta_1)$
  `# The` ε`-acceleration step`
  $\dot{\theta}_{new} \leftarrow \theta_1 + \left[[\theta_0 - \theta_1]^{-1} + [\theta_2 - \theta_1]^{-1}\right]^{-1}$
  `# The re-starting procedure`
  if $\|\dot{\theta}_{new} - \dot{\theta}_{old}\|^2 < \delta_{Re}$ **then**

```
    if ‖θ̇_new − θ̇_old‖² < δ or itr > itrmax then
      Termination of iterations
    end if
    θ_tmp ← M(θ̇_new)
    if L_o(θ_tmp) > L_o(θ_2) then
      θ_2 ← θ_tmp
      θ_1 ← θ̇_new
      δ_Re ← δ_Re × 10^{-k}
    end if
  end if
  θ̇_old ← θ̇_new
  θ_0 ← θ_1
  θ_1 ← θ_2
end repeat
```

Appendix 3: The AEM algorithm of Jamshidian and Jennrich (1993)

We briefly introduce the AEM algorithm, which applies the generalized conjugate gradient (CG) algorithm to accelerate the EM algorithm. The idea of the AEM algorithm is that the change in θ' after an EM iteration $\tilde{\mathbf{g}}(\theta') = M(\theta') - \theta'$ can be viewed approximately as a generalized gradient. Thus the AEM algorithm treats the EM step as a generalized gradient and uses the generalized CG algorithm as an EM accelerator. The generalized gradient $\tilde{\mathbf{g}}(\theta')$ is given by

$$\tilde{\mathbf{g}}(\theta') = M(\theta') - \theta' \approx -\left.\frac{\partial^2 Q(\theta|\theta')}{\partial\theta\partial\theta^\top}^{-1}\right|_{\theta=\theta'} \left.\frac{\partial L_o(\theta)}{\partial\theta}\right|_{\theta=\theta'}.$$

In the AEM algorithm, first the EM algorithm runs until the difference between $2L_o(\theta^{(t)})$ and $2L_o(\theta^{(t-1)})$ falls below one. Then the CG accelerator updates the EM estimate $\theta^{(t)}$ for obtaining $\theta^{(t+1)}$:

1. Set $t = 0$ and $\mathbf{d}^{(t)} = \tilde{\mathbf{g}}(\theta^{(t)})$.
2. Find $\alpha^{(t)}$ to maximize $L_o(\theta^{(t)} + \alpha\mathbf{d}^{(t)})$ using a line search algorithm.
3. Update $\theta^{(t+1)} = \theta^{(t)} + \alpha^{(t)}\mathbf{d}^{(t)}$.
4. Compute

$$\tilde{\mathbf{g}}(\theta^{(t+1)}) = M(\theta^{(t+1)}) - \theta^{(t+1)},$$
$$\gamma^{(t)} = \frac{\{\mathbf{g}(\theta^{(t+1)}) - \mathbf{g}(\theta^{(t)})\}^\top \tilde{\mathbf{g}}(\theta^{(t+1)})}{\{\mathbf{g}(\theta^{(t+1)}) - \mathbf{g}(\theta^{(t)})\}^\top \mathbf{d}^{(t)}},$$

   where $\mathbf{g}(\theta)$ is the gradient of $L_o(\theta)$.
5. Update $\mathbf{d}^{(t+1)} = \tilde{\mathbf{g}}(\theta^{(t+1)}) - \gamma^{(t)}\mathbf{d}^{(t)}$ and then $t = t + 1$.
6. Repeat Steps 2 to 5 until $||\tilde{\mathbf{g}}(\theta^{(t)})||^2 \leq \delta$.

References

Biernacki C, Celeux G, Govaert G (2003) Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. Comput Stat Data Anal 41:561–575

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B 39:1–22

Jamshidian M, Jennrich RI (1993) Conjugate gradient acceleration of the EM algorithm. J Am Stat Assoc 88:221–228

Jamshidian M, Jennrich RI (1997) Acceleration of the EM algorithm by using quasi-Newton methods. J R Stat Soc Ser B 59:569–587

Jones O, Maillardet R, Robinson A (2009) Introduction to scientific programming and simulation using R. Chapman & Hall/CRC, Boca Raton

Karlis D, Xekalaki E (2003) Choosing initial values for the EM algorithm for finite mixtures. Comput Stat Data Anal 41:577–590

Kuroda M, Sakakihara M (2006) Accelerating the convergence of the EM algorithm using the vector ε algorithm. Comput Stat Data Anal 51:1549–1561

Laird NM, Lange K, Stram DO (1987) Maximum likelihood computations with repeated measures: application of the EM algorithm. J Am Stat Assoc 82:97–105

Lange K (1995) A quasi Newton acceleration of the EM algorithm. Stat Sin 5:1–18

Lee G, Scott C (2012) EM algorithms for multivariate Gaussian mixture models with truncated and censored data. Comput Stat Data Anal 56:2816–2829

Lee G, Finn W, Scott C (2011) Statistical file matching of flow cytometry data. J Biomed Inform 44:663–676

Lin TI (2009) Maximum likelihood estimation for multivariate skew normal mixture models. J Multivar Anal 100:257–265

Little RJA, Rubin DB (1987) Statistical analysis with missing data. Wiley, New York

Louis TA (1982) Finding the observed information matrix when using the EM algorithm. J R Stat Soc Ser B 44:226–233

McLachlan GJ, Peel D (2000) Finite mixture models. Wiley, New York

Meng XL, Rubin DB (1994) On the global and componentwise rates of convergence of the EM algorithm. Linear Algebra Appl 199:413–425

Pynea S, Hua X, Wangb K, Rossina E, Linc T, Maiera LM, Baecher-Alland C, McLachlan GJ, Tamayoa P, Haflera DA, De Jagera PL, Mesirova JP (2009) Automated high-dimensional flow cytometry data analysis. Proc Natl Acad Sci USA 106:8519–8524

R Development Core Team (2013) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. http://www.R-project.org

Schafer JL (1997) Analysis of incomplete multivariate data. Chapman & Hall/CRC, London

Snedecor GW, Cochran WC (1967) Statistical methods. Iowa State University Press, Iowa

Wang M, Kuroda M, Sakakihara M, Geng Z (2008) Acceleration of the EM algorithm using the vector epsilon algorithm. Comput Stat 23:469–486

Wynn P (1962) Acceleration techniques for iterated vector and matrix problems. Math Comp 16:301–322