Chapter

Fast Computation of the EM Algorithm for Mixture Models

Masahiro Kuroda

Abstract

Mixture models become increasingly popular due to their modeling flexibility and are applied to the clustering and classification of heterogeneous data. The EM algorithm is largely used for the maximum likelihood estimation of mixture models because the algorithm is stable in convergence and simple in implementation. Despite such advantages, it is pointed out that the EM algorithm is local and has slow convergence as the main drawback. To avoid the local convergence of the EM algorithm, multiple runs from several different initial values are usually used. Then the algorithm may take a large number of iterations and long computation time to find the maximum likelihood estimates. The speedup of computation of the EM algorithm is available for these problems. We give the algorithms to accelerate the convergence of the EM algorithm and apply them to mixture model estimation. Numerical experiments examine the performance of the acceleration algorithms in terms of the number of iterations and computation time.

Keywords: the EM algorithm, normal mixture models, acceleration of convergence, the vector ε algorithm, restarting procedure, initial value selection, the emEM algorithm

1. Introduction

Mixture models become increasingly popular due to their modeling flexibility and are applied to the clustering and classification of heterogeneous data, see [1–3]. The EM algorithm [4] is largely used for the maximum likelihood estimation of mixture models because the algorithm is stable in convergence and simple in implementation. Despite such advantages, it is pointed out that the EM algorithm is local and has slow convergence as the main drawback.

To circumvent the problem of slow convergence of the EM algorithm, various acceleration algorithms incorporating optimization methods are proposed. The optimization methods include the multivariate Aitken method [5], the conjugate gradient method [6], and the quasi-Newton method [7, 8]. However, these methods require 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 may converge faster than the EM algorithm.

As another approach, the ε-accelerated EM algorithm [9] is proposed to accelerate the convergence of the EM algorithm by using the vector ε (vε) algorithm [10] that is a vector extrapolation algorithm [11, 12]. The vε algorithm can accelerate the convergence of the sequence of estimates from the EM algorithm, and therefore, the ε-accelerated EM algorithm does not require any modification of the E- and M-steps of the EM algorithm. This point is the advantage of the ε-accelerated EM algorithm over other acceleration algorithms using the optimization methods. To reduce the number of iterations and computation time of the ε-accelerated EM algorithm, the εR-accelerated EM algorithm [13] is developed. The algorithm improves the computation speed of the ε-accelerated EM algorithm by embedding a restarting procedure. Then the restarting procedure finds a value for restarting the EM iterations such that a newly generated sequence of EM iterations from the value moves quickly into a neighborhood of a stationary point. We use the ε-accelerated EM and εR-accelerated EM algorithms for parameter estimation.

In application of the EM algorithm to mixture models, the algorithm is sensitive to the choice of the initial value and may find estimates at a local maximum of the log-likelihood function. Several strategies are proposed to efficiently initiate the EM algorithm for getting the global maximum of the log-likelihood function, see [14–17]. We use the emEM algorithm [14] for the mixture model estimation and improve its computation speed by the ε-accelerated EM and εR-accelerated EM algorithms.

The chapter is organized as follows. Section 2 describes the EM algorithm for normal mixture models. In Section 3, we introduce the ε-accelerated EM and εR-accelerated EM algorithms. Section 4 presents numerical experiments to evaluate the performance of these acceleration algorithms. In Section 5, we provide an acceleration algorithm that applies the ε-accelerated EM and εR-accelerated EM algorithms to the emEM algorithm. Numerical experiments in Section 6 study the effects of these acceleration algorithms on the emEM algorithm. In Section 7, we present our concluding remarks.

2. The EM algorithm for normal mixture models

Let $\mathbf{Y}_1, \dots, \mathbf{Y}_n$ be p-dimensional random vectors. Assume that an observed data vector $\mathbf{y}_i$ of $\mathbf{Y}_i$ arises from a mixture distribution of G components with density

$$f(\mathbf{y}_i|\boldsymbol{\theta}) = \sum_{k=1}^{G} \lambda_k \phi(\mathbf{y}_i|\boldsymbol{\mu}_k, \Sigma_k), \tag{1}$$

where $\phi(\mathbf{y}_i|\boldsymbol{\mu}_k, \Sigma_k)$ is the k-th component density of a p-variate normal distribution $N_p(\boldsymbol{\mu}_k, \Sigma_k)$ with mean vector $\boldsymbol{\mu}_k$, variance–covariance matrix Σ_k, λ_k is the k-th mixing proportion such that $0 < \lambda_k < 1$ and $\sum_{k=1}^{G} \lambda_k = 1$, and $\boldsymbol{\theta} = [\lambda_1, \dots, \lambda_G, \boldsymbol{\mu}_1^\top, \dots, \boldsymbol{\mu}_G^\top, \mathrm{vec}\Sigma_1^\top, \dots, \mathrm{vec}\Sigma_G^\top]^\top$. Here $\mathrm{vec}\Sigma_k$ is the vectorization of Σ_k. The log-likelihood function of $\boldsymbol{\theta}$ for $\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]$ is

$$\ell_o(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(\mathbf{y}_i|\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{G} \lambda_k \phi(\mathbf{y}_i|\boldsymbol{\mu}_k, \Sigma_k) \right\}. \tag{2}$$

Direct maximization of the function (2) is complicated, and then the maximum likelihood estimate (MLE) of $\boldsymbol{\theta}$ is usually found via the EM algorithm [4].

In the setting of the EM algorithm, we regard $\mathbf{y}_i$ as incomplete data and introduce the component-label vector $\mathbf{Z}_i = [Z_{i1}, \dots, Z_{iG}]^\top$ of zero–one indicator variables such that $Z_{ik} = 1$ indicates that $\mathbf{y}_i$ arises from the k-th component of the mixture model and $Z_{ik} = 0$ otherwise. Assume that $\mathbf{Z}_i$ has a multinomial distribution $Mu(1, \boldsymbol{\lambda})$ with parameter $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_G]^\top$. In the mixture model, the complete data vector is $\mathbf{x}_i = \left[\mathbf{y}_i^\top, \mathbf{z}_i^\top\right]^\top$, where $\mathbf{y}_i$ is the observed vector and $\mathbf{z}_i$ is the unobserved vector of $\mathbf{Z}_i$. Then $\mathbf{x}_i$ has a mixture distribution with density

$$f(\mathbf{x}_i|\boldsymbol{\theta}) = \prod_{k=1}^{G} \left\{\lambda_k \phi\left(\mathbf{y}_i|\boldsymbol{\mu}_k, \Sigma_k\right)\right\}^{z_{ik}}. \tag{3}$$

Given $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$, the log-likelihood function of $\boldsymbol{\theta}$ is

$$\ell_c(\theta) = \sum_{i=1}^{n}\sum_{k=1}^{G} z_{ik} \log \lambda_k \phi\left(\mathbf{y}_i|\boldsymbol{\mu}_k, \Sigma_k\right), \tag{4}$$

and the MLE $\hat{\boldsymbol{\theta}}$ of the function (4) is obtained from

$$\hat{\lambda}_k = \sum_{i=1}^{n} z_{ik}/n, \tag{5}$$

$$\hat{\boldsymbol{\mu}}_k = \sum_{i=1}^{n} z_{ik}\mathbf{x}_i \Big/ \sum_{i=1}^{n} z_{ik}, \tag{6}$$

$$\hat{\Sigma}_k = \sum_{i=1}^{n} z_{ik}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^{\mathrm{T}} \Big/ \sum_{i=1}^{n} z_{ik} \tag{7}$$

for $k = 1, \dots, G$. The EM algorithm finds $\hat{\boldsymbol{\theta}}$ by iterating the expectation step (E-step) and the maximization step (M-step). Let $\boldsymbol{\theta}^{(t)}$ be the t-th estimate of $\boldsymbol{\theta}$ in parameter space Θ. The E-step calculates the Q function that is the conditional expectation of $\ell_c(\boldsymbol{\theta})$ given $\mathbf{y}$ and $\boldsymbol{\theta}^{(t)}$ and is written as

$$Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}\right) = E\left[\ell_c(\boldsymbol{\theta})|\mathbf{y}, \boldsymbol{\theta}^{(t)}\right]. \tag{8}$$

Mixture models treat $\mathbf{z} = [\mathbf{z}_1, \dots, \mathbf{z}_n]$ as missing data. The E-step calculates the conditional expectation of Z_{ik} given $\mathbf{y}$ and $\boldsymbol{\theta}^{(t)}$:

$$\begin{aligned} \tau_{ik}^{(t+1)} &= E\left[Z_{ik}|\mathbf{y}, \boldsymbol{\theta}^{(t)}\right] = \Pr\left(Z_{ik}|\mathbf{y}, \boldsymbol{\theta}^{(t)}\right) \\ &= \lambda_k^{(t)} \phi\left(\mathbf{y}_i|\boldsymbol{\mu}_k^{(t)}, \Sigma_k^{(t)}\right) \Big/ \sum_{k=1}^{G} \lambda_k^{(t)} \phi\left(\mathbf{y}_i|\boldsymbol{\mu}_k^{(t)}, \Sigma_k^{(t)}\right). \end{aligned} \tag{9}$$

The quantity $\tau_{ik}^{(t)}$ is the posterior probability that $\mathbf{y}_i$ belongs to the k-th component of the mixture. From Eq. (9), the Q function (8) is given by

$$Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}\right) = \sum_{i=1}^{n}\sum_{k=1}^{G} \tau_{ik}^{(t+1)} \log \lambda_k \phi\left(\mathbf{y}_i|\boldsymbol{\mu}_k, \Sigma_k\right). \tag{10}$$

The M-step finds $\boldsymbol{\theta}^{(t+1)}$ maximizing the Q function (10) with respect to $\boldsymbol{\theta}$ over Θ given $\boldsymbol{\theta}^{(t)}$:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}\in\Theta} Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}\right). \tag{11}$$

When replacing z_{ik} in Eq. (5) with $\tau_{ik}^{(t+1)}$ in the E-step, we obtain

$$\lambda_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n}\tau_{ik}^{(t+1)}. \tag{12}$$

From Eqs. (6) and (7), we also have

$$\boldsymbol{\mu}_k^{(t+1)} = \sum_{i=1}^{n}\tau_{ik}^{(t+1)}\mathbf{x}_i \Big/ \sum_{i=1}^{n}\tau_{ik}^{(t+1)}, \tag{13}$$

$$\hat{\Sigma}_k^{(t+1)} = \sum_{i=1}^{n}\tau_{ik}^{(t+1)}\left(\mathbf{x}_i - \boldsymbol{\mu}_k^{(t+1)}\right)\left(\mathbf{x}_i - \boldsymbol{\mu}_k^{(t+1)}\right)^{\mathrm{T}} \Big/ \sum_{i=1}^{n}\tau_{ik}^{(t+1)}. \tag{14}$$

We describe the EM algorithm for the normal mixture model in Algorithm 1.

Algorithm 1: The EM algorithm.

E-step: Calculate $\boldsymbol{\tau}_k^{(t+1)} = \left[\tau_{i1}^{(t+1)}, \dots, \tau_{iG}^{(t+1)}\right]^{\mathrm{T}}$ using Eq. (9) and update $\boldsymbol{\tau}^{(t+1)} = \left[\boldsymbol{\tau}_1^{(t+1)}, \dots, \boldsymbol{\tau}_n^{(t+1)}\right]$.

M-step: Estimate $\boldsymbol{\theta}^{(t+1)}$ from Eqs. (12)–(14).

3. Acceleration of the EM algorithm

In order to accelerate the convergence of the EM algorithm, we can use the ε-accelerated EM algorithm [9] and the εR-accelerated EM algorithm [13]. The ε-accelerated EM algorithm incorporates the vector ε (vε) algorithm [10] in the EM algorithm. The εR-accelerated EM algorithm improves the computation speed of the ε-accelerated EM algorithm by adding a restarting procedure.

We briefly introduce the vε algorithm. Let $\left\{\boldsymbol{\theta}^{(t)}\right\}_{t\geq 0}$ be a linearly convergent vector sequence from an iterative computational procedure and converge to a stationary point $\hat{\boldsymbol{\theta}}$ as $t\to\infty$. Then the vε algorithm generates a sequence $\left\{\boldsymbol{\psi}^{(t)}\right\}_{t\geq 0}$ that converges to $\hat{\boldsymbol{\theta}}$ faster than $\left\{\theta^{(t)}\right\}_{t\geq 0}$ by using

$$\psi^{(t-1)} = \boldsymbol{\theta}^{(t)} + \left[\left[\Delta\boldsymbol{\theta}^{(t)}\right]^{-1} - \left[\Delta\boldsymbol{\theta}^{(t-1)}\right]^{-1}\right]^{-1}, \tag{15}$$

where $\Delta\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}$ and $[\boldsymbol{\theta}]^{-1} = \boldsymbol{\theta}/\|\boldsymbol{\theta}\|^2 = \boldsymbol{\theta}/\boldsymbol{\theta}^{\top}\boldsymbol{\theta}$, see Appendix A for details. The algorithm enables accelerating the convergence of a slowly convergent vector sequence and is very effective for linearly convergent sequences.

We define the EM algorithm as a mapping $\boldsymbol{\theta}\mapsto M(\boldsymbol{\theta})$ from Θ to Θ such that each iteration $\boldsymbol{\theta}^{(t)}\to\boldsymbol{\theta}^{(t+1)}$ is denoted by

$$\boldsymbol{\theta}^{(t+1)} = M\left(\boldsymbol{\theta}^{(t)}\right). \tag{16}$$

Algorithm 2: The ε-accelerated EM algorithm.

```
E-step: Estimate $\boldsymbol{\theta}^{(t+1)}$ from Eq. (16).
ε acceleration step Calculate $\psi^{(t-1)}$ from $\left\{\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)}\right\}$ using Eq. (15).
```

The ε-accelerated EM algorithm is shown in Algorithm 2. Given a convergence criterion δ, the ε-accelerated EM algorithm iterates until

$$\|\psi^{(t-1)} - \psi^{(t-2)}\|^2 < \delta. \tag{17}$$

Assume that the sequence $\left\{\boldsymbol{\theta}^{(t)}\right\}_{t\geq 0}$ from the EM algorithm converges to a stationary point $\hat{\boldsymbol{\theta}}$. The εR-accelerated EM algorithm generates $\left\{\psi^{(t)}\right\}_{t\geq 0}$ converging to $\hat{\boldsymbol{\theta}}$ faster than $\left\{\boldsymbol{\theta}^{(t)}\right\}_{t\geq 0}$ and provides $\hat{\boldsymbol{\theta}}$ from the final value of $\left\{\psi^{(t)}\right\}_{t\geq 0}$ when the algorithm terminates.

The theorems with the convergence and acceleration of the algorithm are given in [18].

As shown in Algorithm 2, the ε-accelerated EM algorithm generates two parallel sequences, $\left\{\psi^{(t)}\right\}_{t\geq 0}$ in the ε acceleration step and $\left\{\boldsymbol{\theta}^{(t)}\right\}_{t\geq 0}$ in the EM step. At the ε acceleration step, the EM estimate $M\left(\psi^{(t-1)}\right)$ from $\psi^{(t-1)}$ may have a larger log-likelihood function than the current EM estimate $\boldsymbol{\theta}^{(t+1)}$, that is,

$$\ell_o\left(M\left(\psi^{(t-1)}\right)\right) > \ell_o\left(\boldsymbol{\theta}^{(t+1)}\right). \tag{18}$$

When this occurs, the EM step is restarted with $M\left(\psi^{(t-1)}\right)$ as the initial value, and the ε acceleration step gets $\psi^{(t)}$ from $\left\{\psi^{(t-1)}, M\left(\psi^{(t-1)}\right), M\left(M\left(\psi^{(t-1)}\right)\right)\right\}$. Notice that at the restarting point, we still generate the EM sequence using three estimates obtained from the same initial value $\psi^{(t-1)}$. By this manner, we keep to always apply the ε-acceleration to a sequence obtained by the EM mapping M from the same initial value.

By our experiments, the restarting procedure is performed almost every time when we only use the restarting condition $\ell_o\left(M\left(\psi^{(t-1)}\right)\right) > \ell_o\left(\boldsymbol{\theta}^{(t+1)}\right)$, and then it inefficiently takes much computation time. As one more condition for restarting the EM step, we give $\|\psi^{(t-1)} - \psi^{(t-2)}\|^2 \leq \delta_{Re}(\ \ > \delta)$ and reset $\delta_{Re} = \delta_{Re}/10^k$ at each restarting, where k is an integer, such as one. By adding this condition, we can control the restarting frequency. For example, set $\delta = 10^{-12}$, and initialize $\delta_{Re} = 1$ and $k = 1$. Then the restarting procedure is performed at most 12 times.

The restarting conditions are summarized as follows:

i. $\ell_o\left(M\left(\psi^{(t-1)}\right)\right) > \ell_o\left(\boldsymbol{\theta}^{(t+1)}\right)$, and

ii. $\|\psi^{(t-1)} - \psi^{(t-2)}\|^2 < \delta_{Re}$.

Condition (i) means that the log-likelihood function can be increased by restarting. Condition (ii) is used to reduce the frequency of restarting. This is the key idea of the restarting procedure. The εR-accelerated EM algorithm is the ε-accelerated EM algorithm with the restarting procedure using conditions (i) and (ii) and is given in Algorithm 3.

Algorithm 3: The εR-accelerated EM algorithm.

EM step: Estimate $\boldsymbol{\theta}^{(t+1)}$ from Eq. (16).

ε **acceleration step:** Calculate $\psi^{(t-1)}$ from $\left\{\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t-1)}\right\}$ using Eq. (15).

Restarting step: If $\ell_o\left(M\left(\psi^{(t-1)}\right)\right) > \ell_o\left(\boldsymbol{\theta}^{(t+1)}\right)$ and $\|\psi^{(t-1)} - \psi^{(t-2)}\|^2 < \delta_{Re}$, then set

$$\boldsymbol{\theta}^{(t)} = \psi^{(t-1)}, \tag{19}$$

update

$$\boldsymbol{\theta}^{(t+1)} = M\left(\psi^{(t-1)}\right), \tag{20}$$

and reset

$$\delta_{Re} = \delta_{Re}/10^k. \tag{21}$$

The εR-accelerated EM algorithm also gives $\hat{\boldsymbol{\theta}}$ from the final value of $\left\{\psi^{(t)}\right\}_{t\geq 0}$. When the restarting step effectively finds values for restating the EM step, 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 restarts the EM step at a better current estimate and also keeps that the log-likelihood function increases in the iterations.

Theoretical results of convergence and speed of convergence of the εR-accelerated EM algorithm are given in [13].

4. Numerical experiments for the acceleration of the EM algorithm

We investigate how much faster the ε-accelerated EM and εR-accelerated EM algorithms converge than the EM algorithm. All computations are performed with the statistical package R [19] executing on Windows, Intel Core i5 3.00 GHz with 8 GB of memory.

The R package `MixSim` [17, 20] is used to simulate a random data matrix $\mathbf{y}$ having a p-variate normal mixture distribution of G components. We generate $\mathbf{y} = \left[\mathbf{y}_1, \dots, \mathbf{y}_{1000}\right]$ and find the MLE of $\boldsymbol{\theta}$ using the EM, ε-accelerated EM, and εR-accelerated EM algorithms. The procedure is replicated 100 times. Here, we consider $p = 2, 3, 4, 5, 6$ and $G = 4$. For all experiments, we set $\delta = 10^{-12}$ for convergence of the algorithms, $\delta_{Re} = 1$ and $k = 1$ for the restarting condition of the εR-accelerated EM algorithm. Initial values of the algorithms are obtained from the k-means method using the R function `kmeans`.

Tables 1 and **2** report the results of the number of iterations and CPU time of these algorithms for each p. The CPU times (in seconds) are measured by the R function `proc.time` that times are typically available to 10 milliseconds. For all computations, the acceleration algorithms found the same MLEs as those from the EM algorithm. We see from the tables that the EM algorithm requires a large number of iterations for convergence, whereas two acceleration algorithms converge a smaller number of iterations than the EM algorithm. Then the εR-accelerated EM algorithm can greatly reduce both the number of iterations and CPU time.

To measure the speed of convergence of the EM and two acceleration algorithms, we calculate iteration and CPU time speedups. The iteration speedup of an acceleration algorithm for the EM algorithm is defined by

$$\frac{\text{The number of iterations of the EM algorithm}}{\text{The number of iterations of an acceleration algorithm}}.$$

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
$p = 2$	EM	172.00	467.25	771.00	1069.48	1302.25	10852.00
	ε	133.00	308.50	445.00	697.74	706.50	8090.00
	εR	83.00	182.50	253.50	424.22	396.50	4967.00
$p = 3$	EM	210.00	403.50	628.50	716.33	946.75	1973.00
	ε	121.00	276.75	400.50	484.83	604.75	1566.00
	εR	68.00	167.50	244.50	307.99	359.75	1183.00
$p = 4$	EM	166.00	372.75	468.50	618.63	755.75	2193.00
	ε	120.00	248.75	331.50	400.00	461.50	1452.00
	εR	58.00	139.00	194.50	241.25	291.25	884.00
$p = 5$	EM	141.00	334.75	492.50	879.35	783.00	24886.00
	ε	101.00	235.50	351.00	687.31	516.00	24756.00
	εR	57.00	144.00	226.00	431.55	336.50	14288.00
$p = 6$	EM	193.00	361.25	499.00	655.80	647.75	5910.00
	ε	144.00	252.00	323.50	454.45	473.75	5825.00
	εR	99.00	163.75	230.50	302.13	299.00	4771.00

Table 1.
Summary statistics of the number of iterations of the EM, ε-accelerated EM (ε) and εR-accelerated EM (εR) algorithms for 100 simulated random data. Each data is generated from a p-variate normal mixture distribution of four components.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
$p = 2$	EM	0.39	1.04	1.68	2.31	2.80	22.73
	ε	0.30	0.75	1.08	1.66	1.66	19.18
	εR	0.22	0.49	0.66	1.11	1.04	13.21
$p = 3$	EM	0.75	1.40	2.07	2.64	3.30	8.53
	ε	0.45	1.01	1.46	1.99	2.52	7.60
	εR	0.35	0.68	1.00	1.44	1.68	8.26
$p = 4$	EM	0.42	0.93	1.16	1.53	1.86	5.34
	ε	0.28	0.65	0.86	1.06	1.24	3.80
	εR	0.20	0.44	0.59	0.71	0.86	2.39
$p = 5$	EM	0.25	0.64	0.92	1.65	1.50	46.11
	ε	0.22	0.49	0.72	1.42	1.08	50.36
	εR	0.16	0.35	0.51	0.95	0.80	29.07
$p = 6$	EM	0.51	1.02	1.42	1.84	1.88	17.86
	ε	0.43	0.75	1.02	1.37	1.47	17.75
	εR	0.32	0.54	0.76	0.99	1.00	14.29

Table 2.
Summary statistics of CPU time of the EM, ε-accelerated EM (ε) and εR-accelerated EM (εR) algorithms for 100 random data. Each data is generated from a p-variate normal mixture distribution of four components.

The CPU time speedup is also calculated similarly to the iteration speedup. **Tables 3** and **4** show the results of the iteration and CPU time speedups of two acceleration algorithms. We compare the mean values of the iteration and CPU time speedups of the algorithms. The ε-accelerated EM algorithm is about 1.5 times and 1.4 times faster than the EM algorithm in the number of iterations and CPU time, respectively. Then the εR-accelerated EM algorithm is more than twice as fast as the EM algorithm in both the number of iterations and CPU time. The boxplots of **Figures 1** and **2** also show that the εR-accelerated EM algorithm is obviously much faster than the ε-accelerated EM algorithm. **Table 3** and **Figure 1** indicate that in 75 out of 100 replications, the number of iterations of the εR-accelerated EM algorithm is less than half as small as that of the EM algorithm. For CPU time of **Table 4** and **Figure 2**, the εR-accelerated EM algorithm is more than twice as fast as the EM algorithm in 50 out of 100 replications.

Figure 3 shows the boxplots of the iteration and CPU time speedups of the εR-accelerated EM algorithm for $p = 6$. Here, "more" ("less") means that the number of iterations of the EM algorithm is more (less) than the median in **Tables 1** and **2**.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
$p = 2$	ε	1.05	1.34	1.54	1.61	1.77	3.58
	εR	1.15	2.08	2.73	3.03	3.48	11.36
$p = 3$	ε	1.07	1.32	1.52	1.52	1.68	2.15
	εR	1.20	1.97	2.57	2.58	2.98	6.08
$p = 4$	ε	1.13	1.32	1.48	1.51	1.62	2.33
	εR	1.45	2.09	2.42	2.60	2.94	9.04
$p = 5$	ε	1.01	1.30	1.46	1.47	1.63	2.06
	εR	1.33	1.84	2.23	2.32	2.67	4.32
$p = 6$	ε	1.01	1.28	1.46	1.49	1.65	2.33
	εR	1.24	1.86	2.17	2.37	2.59	6.75

Table 3.
Summary statistics of the iteration speedup of the ε-accelerated EM (ε) and εR-accelerated EM (εR) algorithms for 100 random data. Each data is generated from a p-variate normal mixture distribution of four components.

		Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
$p = 2$	ε	0.97	1.22	1.45	1.47	1.67	3.37
	εR	1.05	1.71	2.24	2.50	2.85	8.60
$p = 3$	ε	0.85	1.21	1.39	1.40	1.56	2.07
	εR	0.78	1.61	2.04	2.08	2.40	4.48
$p = 4$	ε	1.02	1.27	1.39	1.43	1.53	2.11
	εR	1.20	1.70	2.03	2.17	2.43	7.48
$p = 5$	ε	0.92	1.17	1.33	1.34	1.50	2.06
	εR	1.12	1.48	1.76	1.86	2.12	3.21
$p = 6$	ε	0.84	1.18	1.39	1.39	1.55	2.21
	εR	1.00	1.57	1.77	1.98	2.24	5.47

Table 4.
Summary statistics of the CPU time speedup of the ε-accelerated EM (ε) and εR-accelerated EM (εR) algorithms for 100 random data. Each data is generated from p-variate normal mixture distributions of four components.

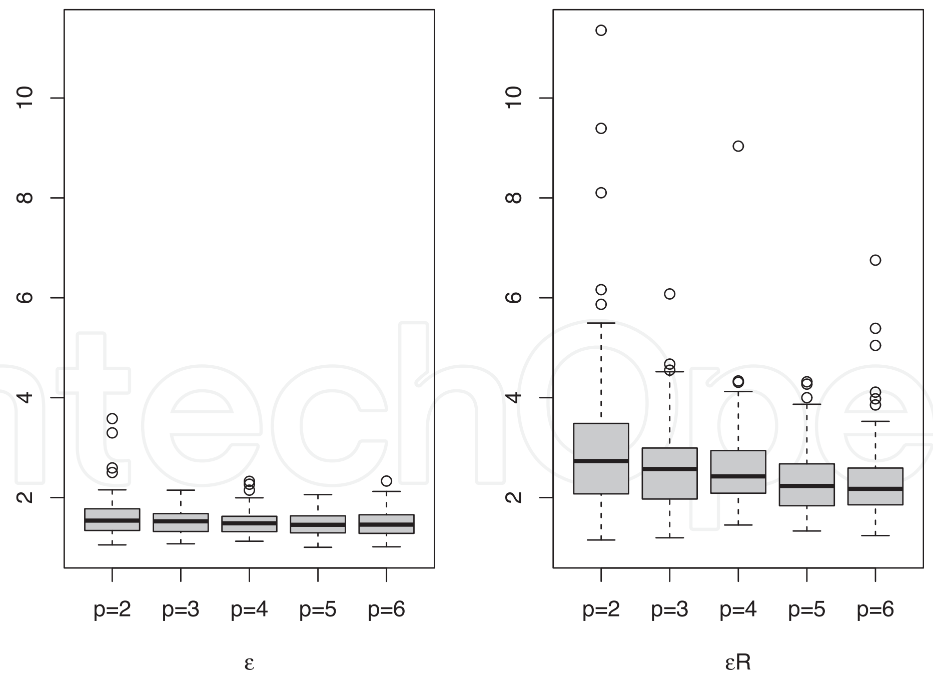

Figure 1.
Boxplots of the iteration speedup of the ε-accelerated EM (ε) and εR-accelerated EM (εR) algorithms for 100 random data generated from a p-variate normal mixture distribution of four components.

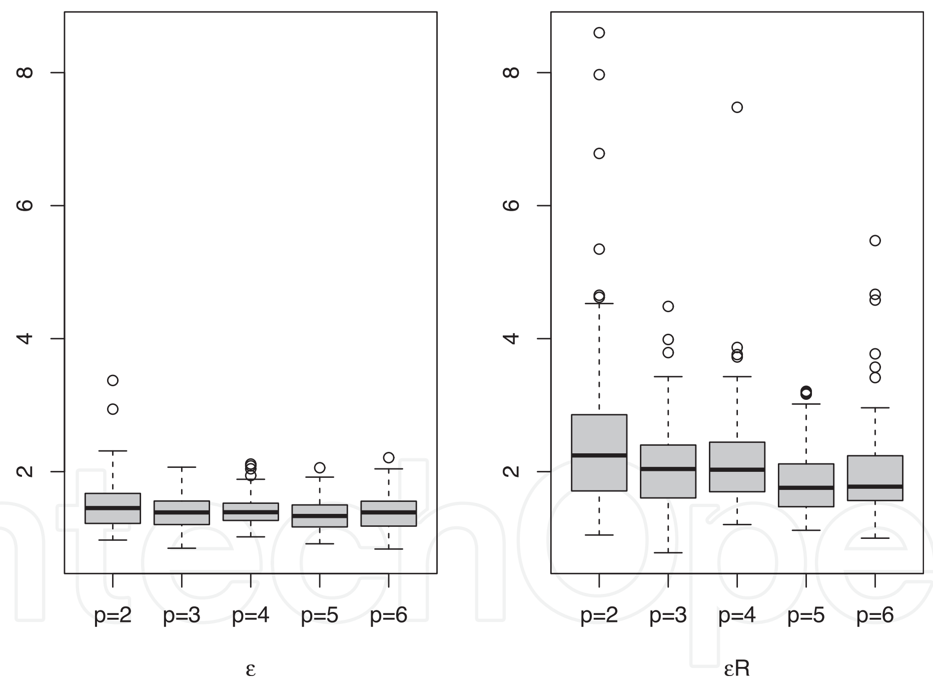

Figure 2.
Boxplots of the CPU time speedup of the ε-accelerated EM (ε) and εR-accelerated EM (εR) algorithms for 100 random data. Each data is generated from a p-variate normal mixture distribution of four components.

We can see from the figure that, for the larger number of iterations of the EM algorithm ("more"), the εR-accelerated EM algorithm works well to speed up the convergence of $\{\psi^{(t)}\}_{t\geq 0}$. We observed a similar result for other p. Therefore, the algorithm is more powerful when the EM algorithm takes a larger number of iterations.

The results from the tables and figures demonstrate that the restarting step in the εR-accelerated EM algorithm enables a significant increase in the computation speed with less computational effort.

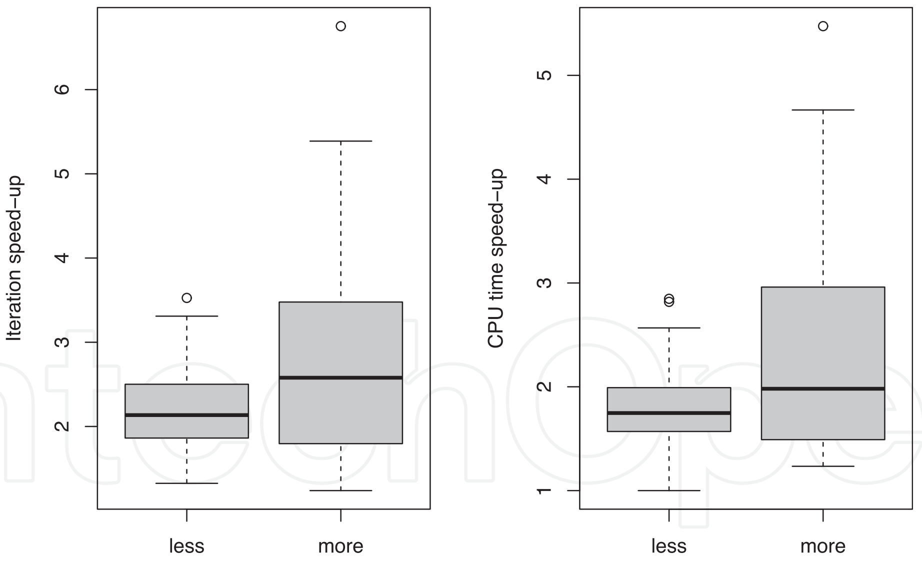

Figure 3.
*Boxplots of the iteration and CPU time speedups of the εR-accelerated EM algorithms for 100 random data. Each data is generated from a six-variate normal mixture distribution of four components. The label "less" ("more") means that the number of iterations of the EM algorithm is less (more) than the median in **Tables 1** and 2.*

5. Initial value selection for normal mixture models

It is well known that the log-likelihood function (2) may have numerous maximums. The EM algorithm does not guarantee to obtain the global maximum of the log-likelihood function due to its local convergence. Thus, the initial value of $\boldsymbol{\theta}$ deeply depends on the performance of the EM algorithm. Several methods for selecting the initial value are proposed; for example, see [14–17]. These methods are based on the multiple runs of the EM algorithm using different initial values and find $\hat{\boldsymbol{\theta}}$ for getting the global maximum of the log-likelihood function.

We apply the emEM algorithm [14] to the mixture model estimation. The algorithm is a popular one and usually provides excellent results when the number of components is not large [21]. The emEM algorithm selects an initial value in the em step that is several short runs of the EM algorithm using different initial values and a lax convergence criterion and obtains $\hat{\boldsymbol{\theta}}$ from the EM step that runs the EM algorithm starting from the initial value with a strict convergence criterion.

The em step consists of three steps. The first step generates J initial values of $\boldsymbol{\theta}$. The second step runs the EM algorithm from these initial values with a lax convergence criterion. Hence, we do not wait for convergence of the EM algorithm and stop the iterations. The third step selects the value giving the largest log-likelihood function among J trials.

Let δ_{ini} be a convergence criterion and T_{max} the maximum number of iterations. We present the emEM algorithm in Algorithm 4.

Algorithm 4: The emEM algorithm.

```
em step: Select $\boldsymbol{\theta}^{(0)}$ of the EM step.
    Random initialization step: Draw $J$ initial values $\left\{\boldsymbol{\theta}^{(0,j)}\right\}_{j=1,\dots,J}$.
    Short running step: Repeat the following computation for $j = 1, \dots, J$:
    Generate $\left\{\boldsymbol{\theta}^{(t_j,j)}\right\}_{t_j \geq 0}$ by iterating the EM algorithm from $\boldsymbol{\theta}^{(0,j)}$ and stop the
    iterations at the $t_j$-iteration if
```

$$\frac{\ell_o\left(\boldsymbol{\theta}^{(t_j,j)}\right)-\ell_o\left(\boldsymbol{\theta}^{(t_j-1,j)}\right)}{\ell_o\left(\boldsymbol{\theta}^{(t_j,j)}\right)-\ell_o\left(\boldsymbol{\theta}^{(0,j)}\right)}<\delta_{ini}, \quad \text{or} \quad t_j>T_{max}. \tag{22}$$

Obtain $\boldsymbol{\theta}^{(*,j)}=\boldsymbol{\theta}^{(t_j,j)}$.

Selection step: From J candidate initial values $\left\{\boldsymbol{\theta}^{(*,j)}\right\}_{j=1,\dots,J}$, find

$$\boldsymbol{\theta}^{(0)}=\underset{\left\{\boldsymbol{\theta}^{(*,j)}\right\}_{j=1,\dots,J}}{\arg\max}\left\{\ell_o\left(\boldsymbol{\theta}^{(*,j)}\right)\right\}_{j=1,\dots,J}. \tag{23}$$

EM step: Given $\boldsymbol{\theta}^{(0)}$ in the em step, find $\hat{\boldsymbol{\theta}}$ using the EM algorithm.

The em step performs multiple runs of the EM algorithm, and then its computation may be time-consuming. We replace the EM algorithm with the ε-accelerated EM algorithm in the em step and use the εR-accelerated EM algorithm to obtain $\hat{\theta}$ in the EM step. By applying these acceleration algorithms to the emEM algorithm, it is possible to reduce the number of iterations and CPU time. The acceleration of the emEM algorithm is referred as to the εem-εREM algorithm and is shown in Algorithm 5.

Algorithm 5: the εem-εREM algorithm.

ε-**em step**: Select $\boldsymbol{\theta}^{(0)}$ of the εR-EM step.

Random initialization step: Draw J initial values $\left\{\boldsymbol{\theta}^{(0,j)}\right\}_{j=1,\dots,J}$.

Short running step: Repeat the following computation for $j=1,\dots,J$:
Generate $\left\{\psi^{(t_j,j)}\right\}_{t_j\geq 0}$ by iterating the ε-accelerated EM algorithm from $\boldsymbol{\theta}^{(0,j)}$ and stop the iterations at the t_j-iteration if

$$\frac{\ell_o\left(\psi^{(t_j,j)}\right)-\ell_o\left(\psi^{(t_j-1,j)}\right)}{\ell_o\left(\psi^{(t_j,j)}\right)-\ell_o\left(\psi^{(0,j)}\right)}<\delta_{ini}, \quad \text{or} \quad t_j>T_{max}. \tag{24}$$

Obtain $\boldsymbol{\theta}^{(*,j)}=\psi^{(t_j,j)}$.

Selection step: From J candidate initial values $\left\{\boldsymbol{\theta}^{(*,j)}\right\}_{j=1,\dots,J}$, find

$$\boldsymbol{\theta}^{(0)}=\underset{\left\{\boldsymbol{\theta}^{(*,j)}\right\}_{j=1,\dots,J}}{\arg\max}\left\{\ell_o\left(\boldsymbol{\theta}^{(*,j)}\right)\right\}_{j=1,\dots,J}. \tag{25}$$

ε-**R-EM step**: Given $\boldsymbol{\theta}^{(0)}$ in the em step, find $\hat{\theta}$ using the εR-accelerated EM algorithm.

6. Numerical experiments for the initial value selection

We evaluate the performance of the ε-accelerated EM and εR-accelerated EM algorithms in application to the emEM algorithm.

	emEM			εem-εREM		
	em	**EM**	**total**	**ε-em**	**εR-EM**	**total**
$p = 2$	1912	3834	5746	1415	1429	2844
$p = 3$	1995	1490	3485	925	354	1279
$p = 4$	2352	725	3077	997	451	1448
$p = 5$	3344	885	4229	1516	397	1913
$p = 6$	2641	957	3598	1234	435	1669

Table 5.
The numbers of iterations of the emEM and εem-εREM algorithms. The em and ε-em steps generate 50 random initial values.

	emEM			εem-εREM		
	em	**EM**	**total**	**ε-em**	**εR-EM**	**total**
$p = 2$	6.04	7.37	13.41	4.67	3.22	7.89
$p = 3$	6.36	3.14	9.50	3.23	1.00	4.23
$p = 4$	8.81	1.61	10.42	3.98	1.86	5.84
$p = 5$	12.55	2.33	14.88	6.04	1.19	7.23
$p = 6$	11.01	2.44	13.45	5.35	1.43	6.78

Table 6.
CPU times of the emEM and εem-εREM algorithms. The em and ε-em steps generate 50 random initial values.

By using MixSim, we simulate $\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_{1000}]$ having the p-variate normal mixture distribution of six components for $p = 2, 3, 4, 5, 6$. The values of δ, δ_{Re}, and k are the same as in the experiments of Section 1.4. Assume that the probability of not finding the global maximum of the log-likelihood function in a single run is 0.80 for safety. Then the probability of finding the global maximum at least once is $1 - 0.80^{50} > 0.9999$. In the em and ε-em steps, we draw 50 initial values $\left\{\boldsymbol{\theta}^{(0,j)}\right\}_{j=1,\dots,50}$ from kmeans and set $\delta_{ini} = 0.001$ and $T_{max} = 1000$.

Tables 5 and **6** present the number of iterations and CPU time for each p. We see from **Table 5** that the number of iterations of the ε-em step is much smaller than that of the em step. The ε-accelerated EM algorithm effectively improves the computation speed of the em step. We compare the number of iterations and CPU time of the εem-εREM algorithm with those of the emEM algorithm. Then these values of the εem-εREM algorithm are about less than half of those of the emEM algorithm. The results illustrate that the ε-accelerated EM and εR-accelerated EM algorithms can sufficiently accelerate the convergence of the emEM algorithm.

7. Concluding remarks

In this chapter, we introduced the ε-accelerated EM and εR-accelerated EM algorithms. Both algorithms are given by very simple computational procedures and are executed with a little bit of computation for each iteration, while they well accelerate the convergence of the EM algorithm.

When the EM algorithm is applied to normal mixture models, the algorithm may converge slowly and be heavily dependent on the initial value. The first problem is solved by the acceleration of the EM algorithm. The numerical experiments indicated the availability of the ε-accelerated EM and εR-accelerated EM algorithms. For the second problem, the initial value selection is useful to initiate the EM algorithm. We applied the emEM algorithm to normal mixture model estimation and developed the εem-εREM algorithm to speed up the computation of the emEM algorithm. Then the ε-accelerated EM algorithm is used in the em step, and the εR-accelerated EM algorithm is in the EM step. Numerical experiments showed that the εem-εREM algorithm can converge in a smaller number of iterations and shorter CPU time than the emEM algorithm.

The ε-accelerated EM and εR-accelerated EM algorithms accelerate the convergence of the EM algorithm without any modification of the E- and M-steps of the algorithm. This means that these algorithms do not require to derive the acceleration formula for every statistical model. Thus, these algorithms are applied to several mixture models—mixtures of factor analyzers, mixtures of multivariate t-distributions, mixtures of generalized hyperbolic distributions, and parsimonious Gaussian mixture models. We expect that the convergence of the EM algorithms used in these mixture models tends to be slow. The results from the experiments show that the εR-accelerated EM and εR-accelerated EM algorithms are useful due to their fast speed of convergence and ease of use.

Appendix: the vector ε algorithm

Let $\boldsymbol{\theta}^{(t)}$ denote a d-dimensional vector that converges to a vector $\hat{\boldsymbol{\theta}}$ as $t \to \infty$. We define $[\boldsymbol{\theta}]^{-1} = \boldsymbol{\theta}/\|\boldsymbol{\theta}\|^2 = \boldsymbol{\theta}/\boldsymbol{\theta}^\top\boldsymbol{\theta}$. In general, the v$\varepsilon$ algorithm for a sequence $\left\{\boldsymbol{\theta}^{(t)}\right\}_{t\geq 0}$ starts with

$$\varepsilon^{(t,-1)} = \mathbf{0}, \qquad \varepsilon^{(t,0)} = \boldsymbol{\theta}^{(t)} \tag{26}$$

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

$$\begin{aligned} \varepsilon^{(t,k+1)} &= \varepsilon^{(t+1,k-1)} + \left[\varepsilon^{(t+1,k)} - \varepsilon^{(t,k)}\right] \\ &= \varepsilon^{(t+1,k-1)} + \left[\Delta\varepsilon^{(t,k)}\right]^{-1}, \qquad k = 0, 1, 2, \dots . \end{aligned} \tag{27}$$

For practical implementation, we apply the vε algorithm for $k = 1$ to accelerate the convergence of $\left\{\boldsymbol{\theta}^{(t)}\right\}_{t\geq 0}$. From the above equation, we have

$$\varepsilon^{(t,2)} = \varepsilon^{(t+1,0)} + \left[\Delta\varepsilon^{(t,1)}\right]^{-1} \quad \text{for } k = 1, \tag{28}$$

$$\varepsilon^{(t,1)} = \varepsilon^{(t+1,-1)} + \left[\Delta\varepsilon^{(t,0)}\right]^{-1} = \left[\Delta\varepsilon^{(t,0)}\right]^{-1} \quad \text{for } k = 0. \tag{29}$$

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

$$\begin{aligned} \varepsilon^{(t,2)} &= \varepsilon^{(t+1,0)} + \left[\left[\Delta\varepsilon^{(t+1,0)}\right]^{-1} - \left[\Delta\varepsilon^{(t,0)}\right]^{-1}\right]^{-1} \\ &= \boldsymbol{\theta}^{(t+1)} + \left[\left[\Delta\boldsymbol{\theta}^{(t+1)}\right]^{-1} - \left[\Delta\boldsymbol{\theta}^{(t)}\right]^{-1}\right]^{-1}. \end{aligned} \tag{30}$$

When setting $\psi^{(t)} = \varepsilon^{(t,2)}$, we obtain Eq. (15).

Author details

Masahiro Kuroda
Okayama University of Science, Okayama City, Japan

*Address all correspondence to: kuroda@mgt.ous.ac.jp

References

[1] Bouveyron C, Celeux G, Murphy TB, Raftery AE. Model-Based Clustering and Classification for Data Science with Applications in R. Cambridge: Cambridge University Press; 2019

[2] McLachlan G, Peel D. Finite Mixture Models. New York: Wiley; 2000

[3] McNicholas PD. Mixture Model-Based Classification. Boca Raton Chapman & Hall/CRC Press; 2016

[4] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. With discussion. Journal of the Royal Statistical Society Series B. 1977;**39**:1-38

[5] Louis TA. Finding the observed information matrix when using the EM algorithm. Journal of the Royal Statistical Society, Series B. 1982;**44**: 226-233

[6] Jamshidian M, Jennrich RI. Conjugate gradient acceleration of the EM algorithm. Journal of the American Statistical Association. 1993; **88**:221-228

[7] Jamshidian M, Jennrich RI. Acceleration of the EM algorithm by using quasi-Newton methods. Journal of the Royal Statistical Society, Series B. 1997;**59**:569-587

[8] Lange K. A quasi Newton acceleration of the EM algorithm. Statistica Sinica. 1995;**5**:1-18

[9] Kuroda M, Sakakihara M. Accelerating the convergence of the EM algorithm using the vector ε algorithm. Computational Statistics & Data Analysis. 2006;**51**:1549-1561

[10] Wynn P. Acceleration techniques for iterated vector and matrix problems. Mathematics of Computation. 1962;**16**: 301-322

[11] Brezinski C, Redivo-Zaglia M. Extrapolation Methods: Theory and Practice. Amsterdam: North-Holland; 1991

[12] Smith DA, Ford F, Sidi A. Extrapolation methods for vector sequences. SIAM Review. 1987;**29**: 199-233

[13] Kuroda M, Geng Z, Sakakihara M. Improving the vector ε acceleration for the EM algorithm using a re-starting procedure. Computational Statistics. 2015;**30**:1051-1077

[14] Biernacki C, Celeux G, Govaert G. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. Computational Statistics & Data Analysis. 2003;**41**: 561-575

[15] Kwedlo W. A new random approach for initialization of the multiple restart EM algorithm for Gaussian model-based clustering. Pattern Analysis and Applications. 2015;**18**:757-770

[16] Maitra R. Initializing optimization partitioning algorithms. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2009;**6**: 144-157

[17] Melnykov V, Chen W, Maitra R. MixSim: An R package for simulating data to study performance of clustering algorithms. Journal of Statistical Software. 2012;**51**:1

[18] Wang M, Kuroda M, Sakakihara M, Geng Z. Acceleration of the EM algorithm using the vector epsilon algorithm. Computational Statistics. 2008;**23**:469-486

[19] R Core Team. R. A Language and Environment for Statistical Computing.

Vienna, Austria: R Foundation for Statistical Computing; 2021; Available from: https://www.R-project.org/

[20] Maitra R, Melnykov V. Simulating data to study performance of finite mixture modeling and clustering algorithms. Journal of Computational and Graphical Statistics. 2010;**19**: 354-376

[21] Michael S, Melnykov V. An effective strategy for initializing the EM algorithm in finite mixture models. Advances in Data Analysis and Classification. 2016;**10**:563-583