On convergence and parameter selection of the EM and DA-EM algorithms for Gaussian mixtures

Jian Yu*, Chaomu Chaomurilige, Miin-Shen Yang

* Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing, China
** Department of Applied Mathematics, Chung Yuan Christian University, Chung-Li 32023, Taiwan

A R T I C L E   I N F O

Article history:
Received 13 November 2016
Revised 6 December 2017
Accepted 17 December 2017
Available online 19 December 2017

Keywords:
Expectation & maximization (EM) algorithm
Deterministic annealing EM (DA-EM)
GAUSSIAN mixtures
Self-annealing
Convergence
Parameter selection

A B S T R A C T

The expectation & maximization (EM) for Gaussian mixtures is popular as a clustering algorithm. However, the EM algorithm is sensitive to initial values, and so Ueda and Nakano [4] proposed the deterministic annealing EM (DA-EM) algorithm to improve it. In this paper, we investigate theoretical behaviors of the EM and DA-EM algorithms. We first derive a general Jacobian matrix of the DA-EM algorithm with respect to posterior probabilities. We then propose a theoretical lower bound for initialization of the annealing parameter in the DA-EM algorithm. On the other hand, some researches mentioned that the EM algorithm exhibits a self-annealing behavior, that is, the equal posterior probability with small random perturbations can avoid the EM algorithm to output the mass center for Gaussian mixtures. However, there is no theoretical analysis on this self-annealing property. Since the DA-EM will become the EM when the annealing parameter is 1, according to the Jacobian matrix of the DA-EM, we can prove the self-annealing property of the EM algorithm for Gaussian mixtures. Based on these results, we give not only convergence behaviors of the equal posterior probabilities and initialization lower bound of the temperature parameter of the DA-EM, but also a theoretical explanation why the EM algorithm for Gaussian mixtures exhibits a self-annealing behavior.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

Since Dempster et al. [1] proposed the EM algorithm to deal with incomplete data, the EM and its extensions have been widely studied and applied in various areas (see [2,3]). In the literature, there are many researches in developing the EM algorithm and its variants for Gaussian mixtures, such as Ueda and Nakano [4], Figueiredo and Jain [5], Zhang et al. [6], Yang et al. [7], Gkalelis et al. [8], Tao et al. [9], Gao et al. [10]. It is always interesting to study the convergence properties of the EM algorithm. Boyles [11] first claimed that the generalized EM sequence will converge to a compact connected set of local maxima of the likelihood function. In other words, the EM may converge to a local maximum or a saddle point of the log likelihood function. Furthermore, Meng and Rubin [13] considered a supplemented EM (SEM) algorithm and then used the SEM as a tool for monitoring whether the EM has converged to a (local) maximum. To analyze the convergence rate of the EM algorithm for Gaussian mixtures, Xu and Jordan [14] and Ma et al. [15] presented the Hessian matrix of the log-likelihood function for Gaussian mixtures with respect to the collection of mixture parameters, Ma and Fu [16] proved that, if the initial parameters are set within the neighborhood, the EM algorithm will always converge to the consistent solution, and Roche et al. [17] considered the convergence property of the three EM-like algorithms for Markov random field segmentation.

In general, the EM for Gaussian mixtures is popular as a clustering algorithm. Since the performance of the EM algorithm for Gaussian mixtures heavily depends on initializations, Ueda and Nakano [5] proposed the deterministic annealing EM (DA-EM) algorithm to improve it. The basic idea of the DA-EM is to begin at a high temperature $\beta$, and then decreases the temperature to zero according to some cooling strategy to avoid poor local optima. The DA-EM algorithm had been studied and applied in various areas, such as Shoham [18], Itaya et al. [19], Guo and Cui [20] and Okamura et al. [21]. On the other hand, Figueiredo and Jain [6] pointed out that, the heuristic behind the deterministic annealing is to force the entropy of the assignments to decrease slowly for avoiding poor local optima. They also mentioned that the EM algorithm itself has a self-annealing behavior without a cooling strategy and just set an uninformative initialization of the posterior probabilities to be $1/c$ with small random perturbations.

https://doi.org/10.1016/j.patcog.2017.12.014
0031-3203/© 2017 Elsevier Ltd. All rights reserved.
Such an observation suggests that, the equal posterior probability with small random perturbations is not an asymptotically stable fixed point of the EM algorithm for Gaussian mixtures, and this can actually interpret why the EM algorithm for Gaussian mixtures exhibits a self-annealing property. However, as we know, no theoretical analysis and proving were proposed in the literature.

In this paper, we investigate convergence behaviors of DA-EM, especially for the annealing parameter in the DA-EM algorithm, and give theoretical results for self-annealing behavior of the EM algorithm. We first construct a new Jacobian matrix of the DA-EM for Gaussian mixtures with respect to posterior probabilities. We then derive a theoretical rule for the valid temperature parameter initialization bound of the DA-EM algorithm to avoid the equal posterior probabilities to be an asymptotically stable fixed point of the DA-EM by using the Jacobian matrix analysis. That is, we give the annealing parameter selection for the DA-EM algorithm. In general, the self-annealing behavior of EM is highly related to the annealing parameter $\beta$ of DA-EM where, if the annealing parameter $\beta$ is equal to 1, then DA-EM becomes EM. Therefore, we can also prove that the EM algorithm always satisfies the necessary condition for the equal posterior probability not being an asymptotically stable fixed point of EM for Gaussian mixtures. That is, we prove that the EM algorithm for Gaussian mixtures can exhibit a self-annealing property. The remainder of the paper is organized as follows. In Section 2, we review the EM and DA-EM algorithms for Gaussian mixtures with problem descriptions. In Section 3, we first construct the Jacobian matrix of the DA-EM with respect to posterior probabilities. We then give some convergence theorems and also the theoretical rule for the valid initialization bound of the annealing parameter $\beta$. And so, we give a theoretical result for the self-annealing behavior of EM. In Section 4, we carry out several experiments to demonstrate our theoretical results. We also use an example to show how to apply the Jacobian matrix of the DA-EM for analyzing its convergence rate. Finally, we give the conclusions in Section 5.

2. The EM and DA-EM algorithms for Gaussian mixtures

In this section, we first give a brief description of the EM algorithm for Gaussian mixtures. A Gaussian mixture can be described as follows:

$$f(x|\Theta) = \sum_{i=1}^{c} \alpha_i f(x|\Theta_i)$$ (1)

where $\alpha_i > 0$, $\sum_{i=1}^{c} \alpha_i = 1$, $\Theta_i = (\mu_i, \Sigma_i)$, $x \in \mathbb{R}^c$ is a column vector, and $f(x|\Theta_i)$ is defined as an s-variate Gaussian distribution with $f(x|\Theta_i) = (2\pi)^{-c/2} |\Sigma_i|^{-1/2} \exp[-\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)]$. Therefore, a Gaussian mixture $f(x|\Theta)$ can be parameterized by the set $\Theta = \{\Theta_1, ..., \Theta_c, \alpha_1, ..., \alpha_c\}$, where $c$ represents the number of components in the Gaussian mixture $f(x|\Theta)$. Let the data set $X = \{x_1, ..., x_n\}$ be a random sample of size $n$ drawn from the distribution $f(x|\Theta)$. Then its log-likelihood function can be written as follows:

$$\log f(X|\Theta) = \sum_{k=1}^{n} \log \sum_{i=1}^{c} \alpha_i f(x_k|\Theta_i)$$ (2)

Obviously, the parameter $\Theta = \{\Theta_1, ..., \Theta_c, \alpha_1, ..., \alpha_c\}$ can be estimated as

$$\Theta_{ML} = \arg \max_{\Theta} \left\{ \log f(X|\Theta) \right\}$$ (3)

For a finite mixture, the EM algorithm considers $X = \{x_1, ..., x_n\}$ as observations. A label set $L = \{l_1, ..., l_n\}$ is considered as a missing part corresponding to the given data $X = \{x_1, ..., x_n\}$ in which $l_k \in \{1, ..., c\}$. If $l_k = i$, it means that the kth data point belongs to the $i$th class. That is, we have $g_k = \{l_1, ..., l_n\}$, where $l_k = 1$ if $l_k = i$; 0, otherwise. Therefore, the complete log-likelihood can be represented as follows:

$$\log f(X, L|\Theta) = \sum_{k=1}^{n} \log \prod_{i=1}^{c} \left[ \alpha_i f(x_k|\Theta_i) \right]^{l_{ki}}$$

$$= \sum_{k=1}^{n} \sum_{i=1}^{c} l_{ki} \log \alpha_i f(x_k|\Theta_i)$$ (4)

Assumed that $\Theta = \{\Theta_1, ..., \Theta_c, \alpha_1, ..., \alpha_c\}$ is given, the conditional expectation $z_{ik}$ can be given by the following Eq. (5).

$$z_{ik} = E[l_{ik}|X, \Theta] = Pr[l_{ik} = 1|x_k, \Theta] = \frac{\alpha_i f(x_k|\Theta_i)}{\sum_{i=1}^{c} \alpha_i f(x_k|\Theta_i)}$$ (5)

For the Gaussian mixture models, after $z_{ik}$ is obtained, the maximum likelihood (ML) estimates of $\log f(X|\Theta)$ for the parameter $\Theta = \{\Theta_1, ..., \Theta_c, \alpha_1, ..., \alpha_c\}$ are as follows:

$$\alpha_i = \frac{1}{n} \sum_{k=1}^{n} z_{ik} \quad (6)$$

$$\mu_i = \frac{\sum_{k=1}^{n} z_{ik} x_k}{\sum_{k=1}^{n} z_{ik}} \quad (7)$$

$$\Sigma_i = \frac{\sum_{k=1}^{n} z_{ik} (x_k - \mu_i)(x_k - \mu_i)^T}{\sum_{k=1}^{n} z_{ik}} \quad (8)$$

According to the theorem of Wu [12] for the EM sequence, the EM algorithm for Gaussian mixtures converges to the stationary points of $\log f(X|\Theta)$. When $\Theta$ is a stationary point of $\log f(X|\Theta)$, Ma et al. [15] gave the Hessian matrix of $\log f(X|\Theta)$ with respect to $\Theta$ and then gave the result about the asymptotic convergence rate of the EM algorithm for Gaussian mixtures.

Since the EM algorithm for Gaussian mixtures is sensitive to initials, Ueda and Nakano [4] proposed the deterministic annealing EM (DA-EM) algorithm to improve it. The DA-EM algorithm introduces a parameter $\beta$ with its reciprocal corresponding to the “temperature”. The only difference between the DA-EM and EM algorithms is that the DA-EM adds the annealing parameter $\beta$ in the posterior probability $z_{ik}$ as

$$z_{ik} = \frac{(\alpha_i f(x_k|\Theta_i))^{\beta}}{\sum_{i=1}^{c} (\alpha_i f(x_k|\Theta_i))^{\beta}}$$ (9)

It is obvious that the DA-EM will become the EM when $\beta = 1$. The DA-EM algorithm starts $\beta(0)$ at a small enough value (i.e. high temperature) and slowly increases $\beta$ up to 1. Thus, the DA-EM algorithm (Ueda and Nakano [4]) can be rewritten as follows:

DA-EM algorithm (Ueda and Nakano [4])

1. Initialize
   • Set $\beta = \beta(0)$ if $\beta(0) < 1$.
   • Set $\Theta(0)$ using k-means algorithm for better results.

2. Iterate until convergence
   • E-step: estimate posterior probabilities by Eq. (9).
   • M-step: estimate $\Theta(\text{new})$ by Eqs. (6), (7) and (8).

3. Increase $\beta$.
4. If $\beta < 1$, go back to step 2;

Else stop the procedure.

Note that, in this paper we increase the value of parameter $\beta$ with 1.01 times, i.e. $\beta^{(\text{new})} = \beta^{(\text{old})} \times 1.01$. In fact, how much increasing in $\beta$ can be determined by users. We will discuss the influence of the increasing factor in numerical examples and experiments of Section 4. In the DA-EM algorithm, the problem of maximizing the log-likelihood function is reformulated as the problem of minimizing a free energy function. The algorithm begins at high temperature corresponding to high entropy that the initial
\( \beta^{(0)} \) should be chosen to a small enough value such that the EM steps can achieve a single global optimum. However, if \( \beta^{(0)} \to 0 \), all the clusters are overlapping and the posterior probabilities will become a uniform distribution. In this sense, the initial \( \beta^{(0)} \) should not be chosen too small to avoid the iterative process achieving overlapping clusters. But the question is how a small value of \( \beta^{(0)} \) should be initialized, that is, what should be the lower bound for the valid initial \( \beta^{(0)} \). There is less work on this initialization selection for the annealing parameter \( \beta \) in the literature. In this paper, we will make theoretical analysis on it by using Jacobian matrix analysis. On the other hand, Figueiredo and Jain [5,22] had pointed out that the EM algorithm exhibits a self-annealing behavior [23]. In other words, the equal posterior probabilities with small random perturbations will not be an asymptotically stable fixed point of the EM algorithm for Gaussian mixtures. Since the update equations between EM and DA-EM are the same except different update equations for the posterior probability \( z_{ik} \), we will first construct a Jacobian matrix of the DA-EM as a basis of our theoretical analysis in next section.

In this paper, the parameter space \( \Psi = \{z = [z_{ik}]_{c \times n} \mid 1 \leq i \leq c, 1 \leq k \leq n, z_{ik} \geq 0, \sum_{i=1}^{c} z_{ik} = 1 \} \) is considered. We first rewrite the DA-EM algorithm for a fixed \( \beta \) with the mapping \( z^{(t+1)} = \theta^{(t)}(z^{(t)}) \) as follows:

\[
z_{ik}^{(t+1)} = \theta_{ik}^{(t)}(z^{(t)}) = \frac{(d^{(t+1)}_{ik})^{\beta}}{\sum_{j=1}^{c} (d^{(t+1)}_{jk})^{\beta}}
\]

and

\[
d_{ik}^{(t+1)} = \alpha_{ik}^{(t+1)} \left( \det \left( \sum_{i} \right) \right)^{-\frac{1}{2}} \times \exp \left( -\frac{1}{2} (x_{k} - \mu_{i}^{(t+1)})^T \left( \sum_{i} \right)^{-1} (x_{k} - \mu_{i}^{(t+1)}) \right)
\]

where the update equations for \( \alpha_{i}^{(t+1)}, \mu_{i}^{(t+1)}, \Sigma^{(t+1)} \) are the same as (6)-(8), respectively, and \( t \) represents the number of iterations. It is obvious that if \( z = [z_{ik}]_{c \times n} \) converges to \( z^{*} \in \Psi \), then \( z^{*} \) must satisfy

\[
z^{*} = \theta^{(\beta)}(z^{*})
\]

If we set the \( c \times n \) matrix point \( z = [z_{ik}]_{c \times n} \) with \( z = z^{*} = [c_{*}]_{c \times n} \), then the updated equations will give \( \mu_{*}^{*} = \sum_{i=1}^{c} x_{i} / n = \bar{x}, \Sigma_{*} = \frac{1}{2} \bar{x} \bar{x}^T \). Then the only clustering center of the DA-EM algorithm is the mass center \( \bar{x} \) of the data set, and so we still get \( z^{*} \) and \( \bar{x} \) in the next iteration. In other words, \( z^{*} = \theta^{(\beta)}(z^{*}) \) and we have that \( z^{*} \) is exactly a stable fixed point of the DA-EM algorithm. If the DA-EM algorithm converge to this point \( z = z^{*} = [c_{*}]_{c \times n} \), then it should output the results with all cluster centers \( \mu_{*}^{*} = \bar{x} \). Of course, this kind of situation should be avoided. In fact, for the DA-EM algorithm, if we initialize \( \beta \) with an improper value \( \beta^{(0)} \), then the algorithm will not escape from \( z^{*} \). In this sense, how to choose an appropriate initial value of \( \beta \) for the DA-EM algorithm becomes an important problem. We address this parameter initialization selection by using the proposed Jacobian matrix analysis in the next section. Furthermore, we also theoretically analyze the self-annealing properties of the EM algorithm.

3. Convergence theorems and parameter selection of the EM and DA-EM based on Jacobian matrix analysis

In general, the Jacobian matrix \( \frac{\partial \theta^{(t)}(z^{(t)})}{\partial z} \) can theoretically judge whether or not the DA-EM algorithm converges to a local maxi-
mum. If the spectral radius of the Jacobian matrix \( \frac{\partial \theta^{(t)}(z^{(t)})}{\partial z} \) at \( z = z^{*} \) is less than 1, then the DA-EM algorithm converges to the local maximum \( z = z^{*} \). This can follow from Olver’s Corollary [24] as follows:

Olver’s Corollary [Olver 24, p. 143] If the Jacobian matrix \( g'(z^{*}) = \frac{\partial \theta^{(t)}(z^{(t)})}{\partial z} \) is a convergent matrix, meaning that its spectral radius \( \lambda(g'(z^{*})) \) satisfies \( \lambda(g'(z^{*})) < 1 \), then \( z^{*} \) is an asymptotically stable fixed point.

We have rewritten the DA-EM algorithm with the mapping \( z^{(t+1)} = \theta^{(t)}(z^{(t)}) \). Next, we first construct the formula for the element \( \frac{\partial \theta_{ik}^{(t)}}{\partial z_{jr}} \) of the Jacobian matrix \( \frac{\partial \theta_{ik}^{(t)}}{\partial z} \) as the following Theorem 1. By the fact, the posterior probability \( z_{ik} \) has the constraint \( \sum_{j=1}^{c} z_{ik} = 1 \), i.e. \( z_{ik} = 1 - \sum_{j=1}^{c-1} z_{jk} \), and so we derive the Jacobian matrix \( \frac{\partial \theta_{ik}^{(t)}}{\partial z} \) by taking the derivatives of \( \theta_{ik}^{(t)} \) with respect to \( z_{jr} \) for \( j = 1, \ldots, c-1 \) and \( r = 1, \ldots, n \).

Theorem 1. For \( i = 1, \ldots, c, k = 1, \ldots, n \) and \( j = 1, \ldots, c-1 \), \( r = 1, \ldots, n \),

\[
\frac{\partial \theta_{ik}^{(t)}}{\partial z_{jr}} = -\beta \frac{z_{ik} z_{jk}}{2 n \sigma_{jr}} (H_{j} + \beta \frac{z_{ik} z_{jk}}{2 n \sigma_{jr}}) + \beta \frac{\delta_{ik} z_{jk}}{2 n \sigma_{jr}} (H_{jr} + \beta \frac{\delta_{ik} z_{jk}}{2 n \sigma_{jr}}) \tag{13}
\]

where \( (H_{j} + \beta \frac{z_{ik} z_{jk}}{2 n \sigma_{jr}}) \) is the correcting matrix of \( \theta_{ik}^{(t)} \) by taking the derivatives of \( \theta_{ik}^{(t)} \) with respect to \( z_{jr} \) for \( j = 1, \ldots, c-1 \) and \( r = 1, \ldots, n \).

Proof. Each element \( \frac{\partial \theta_{ik}^{(t)}}{\partial z_{jr}} \) of the Jacobian matrix \( \frac{\partial \theta_{ik}^{(t)}}{\partial z} \) of the DA-EM algorithm is obtained by taking the derivatives of \( \theta_{ik}^{(t)} \) with respect to \( z_{jr} \) for \( j = 1, \ldots, c-1 \) and \( r = 1, \ldots, n \) as follows:

\[
\frac{\partial \theta_{ik}^{(t)}}{\partial z_{jr}} = \frac{\left( \sum_{i=1}^{c} (d_{ik})^{\beta} - (d_{jk})^{\beta} \right) \left( \frac{\partial (\sum_{i=1}^{c} (d_{ik})^{\beta})}{\partial z_{jr}} \right)}{\left( \sum_{i=1}^{c} (d_{ik})^{\beta} \right)^{2}} - \frac{\left( \sum_{i=1}^{c} (d_{ik})^{\beta} \right) \left( \frac{\partial (\sum_{i=1}^{c} (d_{ik})^{\beta})}{\partial z_{jr}} \right)}{\left( \sum_{i=1}^{c} (d_{ik})^{\beta} \right)^{2}} \tag{14}
\]

Recall that \( z_{jr} = 1 - \sum_{j=1}^{r-1} z_{jk} \). Then we have

\[
\frac{\partial \theta_{ik}^{(t)}}{\partial z_{jr}} = \frac{\delta_{ik} (d_{jk})^{\beta-1} \frac{\partial (d_{jk})^{\beta}}{\partial z_{jr}} - (d_{jk})^{\beta} \left( \frac{\partial (\sum_{i=1}^{c} (d_{ik})^{\beta})}{\partial z_{jr}} - \frac{\partial (\sum_{i=1}^{c} (d_{ik})^{\beta})}{\partial z_{jr}} \right)}{\left( \sum_{i=1}^{c} (d_{ik})^{\beta} \right)^{2}} \tag{15}
\]

Since \( d_{jk} = \sqrt{\det (\sum_{j}^{1} z_{jk})^{1} \sum_{i=1}^{c} (x_{i} - \mu_{j}) \alpha_{j}} \), we have

\[
\frac{\partial d_{jk}}{\partial z_{jr}} = \left( -\frac{1}{2} \left( \frac{1}{2} \frac{\partial \det (\sum_{j}^{1} z_{jk})}{\partial z_{jr}} \right)^{1} \frac{\partial \det (\sum_{j}^{1} z_{jk})}{\partial z_{jr}} \right)^{1} \times \exp \left( -\frac{1}{2} (x_{k} - \mu_{j})^T \sum_{i=1}^{c} (x_{i} - \mu_{j}) \right) \alpha_{j}
\]
\[
\begin{align*}
+ \left( \sqrt{\det \sum_j} \right)^{-1} \frac{\partial \exp \left[ -\frac{1}{2} (x_k - \mu_j)^T \sum_j^{-1} (x_k - \mu_j) \right]}{\partial z_{jr}} \\
+ \frac{1}{n} \left( \sqrt{\det \sum_j} \right)^{-1} \exp \left[ -\frac{1}{2} (x_k - \mu_j)^T \sum_j^{-1} (x_k - \mu_j) \right]
\end{align*}
\]

Because
\[
\frac{\partial \exp \left[ -\frac{1}{2} (x_k - \mu_j)^T \sum_j^{-1} (x_k - \mu_j) \right]}{\partial z_{jr}} = \left( -\frac{1}{2} \right) \exp \left[ -\frac{1}{2} (x_k - \mu_j)^T \sum_j^{-1} (x_k - \mu_j) \right] \times \frac{\partial (x_k - \mu_j)^T \sum_j^{-1} (x_k - \mu_j)}{\partial z_{jr}}
\]
\[
= \exp \left[ -\frac{1}{2} (x_k - \mu_j)^T \sum_j^{-1} (x_k - \mu_j) \right] \left( x_k - \mu_j \right)^T \sum_j^{-1} \left( x_k - \mu_j \right) \times \left( x_k - \mu_j \right)^T \left( \frac{\partial \sum_j^{-1}}{\partial z_{jr}} \right) (x_k - \mu_j),
\]
we can obtain
\[
\begin{align*}
\frac{\partial d_{ik}}{\partial z_{jr}} &= \left( -\frac{1}{2} \right) d_{jk} \left( \det \sum_j \right)^{-1} \frac{\partial \det \sum_i}{\partial z_{jr}} + \frac{1}{na_j} \frac{d_{jk}}{2} \\
&+ d_{jk} \left( x_k - \mu_j \right)^T \sum_j^{-1} \left( x_k - \mu_j \right) \\
&- \frac{d_{jk}}{2} \left( x_k - \mu_j \right)^T \left( \frac{\partial \sum_j^{-1}}{\partial z_{jr}} \right) (x_k - \mu_j)
\end{align*}
\]

The element \( \frac{\partial \theta^p_j}{\partial z_{j^*}} \) in the Jacobian matrix \( \frac{\partial \theta^p}{\partial z} \) of the DA-EM algorithm can be rewritten as:
\[
\frac{\partial \theta^p_j}{\partial z_{j^*}} = -\delta_{ij} \beta (d_{ik})^\beta \frac{\partial \det \sum_i}{\partial z_{j^*}} + \delta_{ij} \beta (d_{ik})^\beta \sum_{i=1}^c (x_k - \mu_i)^T \\
\times \sum_{i} \left( x_k - \mu_i \right)^T + \frac{\delta_{ij} \beta (d_{ik})^\beta}{na_j} \sum_{i=1}^c \frac{d_{ik}}{(d_{ik})^\beta} (x_k - \mu_i)^T \\
- \delta_{ij} \beta (d_{ik})^\beta \sum_{i=1}^c \frac{d_{ik}}{(d_{ik})^\beta} (x_k - \mu_i)^T \\
+ \frac{(d_{ik})^\beta \beta (d_{ik})^\beta}{2 \left( \sum_{i=1}^c (d_{ik})^\beta \right)^2} \frac{\beta (d_{ik})^\beta}{\beta (d_{ik})^\beta} \frac{\partial \det \sum_j}{\partial z_{jr}} \\
+ \frac{(d_{ik})^\beta \beta (d_{ik})^\beta}{2 \left( \sum_{i=1}^c (d_{ik})^\beta \right)^2} \frac{\beta (d_{ik})^\beta}{\beta (d_{ik})^\beta} \sum_{i=1}^n \sum_{k=1}^n z_{ik} \\
- \frac{\beta (d_{ik})^\beta (d_{ik})^\beta}{na_j} \left( \sum_{i=1}^c (d_{ik})^\beta \right)^2 \sum_{k=1}^n z_{ik}
\]
Furthermore, we have
\[
\frac{\partial z_{j^*}}{\partial z_{jr}} = -\sum_i \frac{\partial \sum_i}{\partial z_{jr}} = -\sum_i \left( \frac{1}{\sum_{k=1}^c z_{ik}} (x_k - \mu_i) (x_k - \mu_i)^T \right) \frac{\partial \sum_i}{\partial z_{jr}} \\
+ \left( \frac{1}{\sum_{k=1}^c z_{ik}} \right) \sum_{i}
\]

After appropriately simplified, we get that
\[
\frac{\partial \theta^p_j}{\partial z_{j^*}} = -\delta_{ij} \beta z_{ik} \left( x_k - \mu_i \right)^T \sum_{j=1}^n (x_k - \mu_i) \\
- \frac{\delta_{ij} \beta z_{ik} \beta (d_{ik})^\beta}{na_i} \sum_{i=1}^c \frac{d_{ik}}{(d_{ik})^\beta} (x_k - \mu_i)^T \\
- \frac{\beta z_{ik} \beta (d_{ik})^\beta}{na_i} \frac{\beta (d_{ik})^\beta}{\beta (d_{ik})^\beta} \frac{\partial \det \sum_j}{\partial z_{jr}} \\
+ \frac{\beta z_{ik} \beta (d_{ik})^\beta}{na_i} \frac{\beta (d_{ik})^\beta}{\beta (d_{ik})^\beta} \sum_{i=1}^n \sum_{k=1}^n z_{ik} \\
- \frac{\beta z_{ik} \beta (d_{ik})^\beta}{na_i} \left( \sum_{i=1}^c (d_{ik})^\beta \right)^2 \sum_{k=1}^n z_{ik}
\]
\[ + \beta z_i z_k \left( (x_i - \mu_c)^T \sum_{c}^{-1} (x_i - \mu_c) \right) + \frac{\beta z_i z_k}{n \alpha_c} \]
\[ - \frac{\beta z_i z_k}{2} (x_i - \mu_c)^T \]
\[ \times \left( - \left( \frac{1}{c} (x_i - \mu_c) (x_i - \mu_c) \sum_{c}^{-1} \right) (x_i - \mu_c) \right) \]
\[ = - \frac{\delta_i \beta z_i}{2n \alpha_i} \left( (x_i - \mu_c)^T \right) \sum_{i}^{-1} (x_i - \mu_c) - 5 \]
\[ - 2 (x_i - \mu_c)^T \sum_{c}^{-1} (x_i - \mu_c) - 2 (x_i - \mu_c)^T \]
\[ \times \left( \frac{1}{c} (x_i - \mu_c) (x_i - \mu_c) \sum_{c}^{-1} \right) (x_i - \mu_c) \right) \]
\[ = \frac{\delta_i \beta z_i}{2n \alpha_i} \left( 2 + 2 (x_i - \mu_c)^T \sum_{c}^{-1} (x_i - \mu_c) - (x_i - \mu_c)^T \right) \]
\[ \times \left( \frac{1}{c} (x_i - \mu_c) \right) \sum_{c}^{-1} (x_i - \mu_c) - 5 \]
\[ + \left( (x_i - \mu_c)^T \sum_{c}^{-1} (x_i - \mu_c) \right)^2 \]
\[ - \frac{\beta z_i z_k}{2n \alpha_j} \left( 2 + 2 (x_i - \mu_c)^T \sum_{c}^{-1} (x_i - \mu_c) - (x_i - \mu_c)^T \right) \]
\[ \times \left( \frac{1}{c} (x_i - \mu_c) \right) \sum_{c}^{-1} (x_i - \mu_c) - 5 \]
\[ + \left( (x_i - \mu_c)^T \sum_{c}^{-1} (x_i - \mu_c) \right)^2 \]
\[ + \beta z_i z_k \left( (x_i - \mu_c)^T \sum_{c}^{-1} (x_i - \mu_c) \right) + \frac{\beta z_i z_k}{n \alpha_c} \]
\[ = \frac{\delta_i \beta z_i}{2n \alpha_i} \left( (x_i - \mu_c)^T \sum_{i}^{-1} (x_i - \mu_c) + 5 + (x_i - \mu_c)^T \right) \]
\[ \times \left( \frac{1}{c} (x_i - \mu_c) \right) \sum_{c}^{-1} (x_i - \mu_c) - 5 \]
\[ + \left( (x_i - \mu_c)^T \sum_{c}^{-1} (x_i - \mu_c) \right)^2 \]
\[ \times \left( (x_i - \mu_c)^T \sum_{c}^{-1} (x_i - \mu_c) \right) + \frac{\beta z_i z_k}{n \alpha_c} \]
\[ \]
Proof. We first set $z_{ik} = c^{-1} \forall i, k$. By Eqs. (6)-(9), we have that $\forall i, \mu_i = x = n^{-1} \sum_{k=1}^{n} x^{(k)}$ and $\Sigma_i = n^{-1} \sum_{k=1}^{n} (x^{(k)} - \bar{x})(x^{(k)} - \bar{x})^T$. Therefore, we can obtain $\forall i, H_i = H$. In other words, we have that $\beta H_i = (\theta H_i)$. By a simple computation, we know that

$$\begin{align*}
A = & \begin{bmatrix}
\sqrt{2} & \sqrt{2} & \ldots & \sqrt{2} \\
\sqrt{2} & \sqrt{2} & \ldots & \sqrt{2} \\
\vdots & \vdots & \ddots & \vdots \\
\sqrt{2} & \sqrt{2} & \ldots & \sqrt{2} \\
\text{vec}(\sigma^{-1/2}_x(x_i-x)) & \text{vec}(\sigma^{-1/2}_x(x_i-x)) & \ldots & \text{vec}(\sigma^{-1/2}_x(x_i-x))
\end{bmatrix}
\begin{bmatrix}
(x_i-x)^T \sigma^{-1/2}_x \\
(x_i-x)^T \sigma^{-1/2}_x \\
\vdots \\
(x_i-x)^T \sigma^{-1/2}_x \\
(x_i-x)^T \sigma^{-1/2}_x
\end{bmatrix}
\begin{bmatrix}
\sqrt{2} \\
\sqrt{2} \\
\vdots \\
\sqrt{2} \\
\sqrt{2}
\end{bmatrix}
\end{align*}
$$

For the matrices $A^T = (\; \vdots \; )$ and $B = (\; \vdots \; )$, simple computation can lead to the equation $\text{tr}(A^TB) = \sum_{j=1}^{n} a_{ij} b_{ij} = (\text{vec}(A))^T \text{vec}(B)$. It implies that

$$
\begin{align*}
\text{tr}(\sigma^{-1/2}_x (x_i-x)(x_i-x)^T \sigma^{-1/2}_x - I)
& = (\sigma^{-1/2}_x (x_i-x)(x_i-x)^T \sigma^{-1/2}_x - I)
\end{align*}
$$

Therefore, we obtain

$$(H)^{1/2} = -(x_i-x)^T \sigma^{-1}_x (x_i-x) + s - 1 - (x_i-x)^T \sigma^{-1}_x (x_i-x) + \left[ (x_i-x)^T \sigma^{-1}_x (x_i-x) + 1 \right]^2
+ \text{vec}(\sigma^{-1/2}_x (x_i-x)(x_i-x)^T \sigma^{-1/2}_x - I)
\times \text{vec}(\sigma^{-1/2}_x (x_i-x)(x_i-x)^T \sigma^{-1/2}_x - I)
$$

By computation, the above equation implies that

$$
\begin{align*}
\text{vec}(\sigma^{-1/2}_x (x_i-x)(x_i-x)^T \sigma^{-1/2}_x - I)
\times \text{vec}(\sigma^{-1/2}_x (x_i-x)(x_i-x)^T \sigma^{-1/2}_x - I)
& = -(x_i-x)^T \sigma^{-1}_x (x_i-x) - (x_i-x)^T \sigma^{-1}_x (x_i-x)
+ \left[ (x_i-x)^T \sigma^{-1}_x (x_i-x) + 1 \right]^2
\times \text{vec}(\sigma^{-1/2}_x (x_i-x)(x_i-x)^T \sigma^{-1/2}_x - I)
\times \text{vec}(\sigma^{-1/2}_x (x_i-x)(x_i-x)^T \sigma^{-1/2}_x - I)
\end{align*}
$$

Since $a^T b = \text{tr}(a^T b)$ where $a$ and $b$ are column vectors, we have

$$(x_i-x)^T \sigma^{-1}_x (x_i-x) = \text{tr}(\sigma^{-1/2}_x (x_i-x)(x_i-x)^T \sigma^{-1/2}_x)
$$

and

$$(x_i-x)^T \sigma^{-1}_x (x_i-x) = \text{tr}(\sigma^{-1/2}_x (x_i-x)(x_i-x)^T \sigma^{-1/2}_x)
$$

Consider that $\text{tr}(ABC) = \text{tr}(BCA) = \text{tr}(CAB)$, we obtain

$$(x_i-x)^T \sigma^{-1}_x (x_i-x) + s - 1 - (x_i-x)^T \sigma^{-1}_x (x_i-x) + \left[ (x_i-x)^T \sigma^{-1}_x (x_i-x) + 1 \right]^2
$$. 

Consequently, we have that

$$
\begin{align*}
-(x_i-x)^T \sigma^{-1}_x (x_i-x) + s - 1 - (x_i-x)^T \sigma^{-1}_x (x_i-x)
+ \left[ (x_i-x)^T \sigma^{-1}_x (x_i-x) + 1 \right]^2
\end{align*}
$$

Thus, the necessary condition for that, $x^* = [c^{-1}]_{1 \times n}$ is not an asymptotically stable fixed point of the DA-EM algorithm, should be $\beta \geq (\lambda_{\text{max}})^{-1}$. 

In Lemma 1, we derive the simple matrix calculation formula (14) for the Jacobian matrix of the DA-EM at the special matrix point $z^* = [c^{-1}]_{1 \times n}$. These can be simply computed by using the MATLAB system. We next create a theoretical method to calculate the lower bound of the annealing parameter $\beta$. Based on Olver’s Corollary and Lemma 1, we can prove the following Theorem 2.

Theorem 2. Let $\lambda_{\text{max}}^*$ be the spectral radius of the matrix $\frac{\beta}{2} \frac{\partial (\beta \lambda_{\text{max}}^*)}{\partial z}$ at the matrix point $z^* = [c^{-1}]_{1 \times n}$ with $\frac{\partial (\beta \lambda_{\text{max}}^*)}{\partial z} |_{z^*} = \frac{\beta}{2} (H)^{1/2}$. The necessary condition for that, $z^* = [c^{-1}]_{1 \times n}$ is not an asymptotically stable fixed point of the DA-EM algorithm, should be $\beta \geq (\lambda_{\text{max}}^*)^{-1}$. 

Proof. Since $\lambda_{\text{max}}^*$ is the spectral radius of the matrix $\frac{\beta}{2} \frac{\partial (\beta \lambda_{\text{max}}^*)}{\partial z}$ at $z^* = [c^{-1}]_{1 \times n}$, $\lambda_{\text{max}}^*$ is the spectral radius of the Jacobian matrix $\frac{\partial (\beta \lambda_{\text{max}}^*)}{\partial z}$ at $z^* = [c^{-1}]_{1 \times n}$. From Olver’s Corollary, we know that, if the spectral radius of the Jacobian matrix $\frac{\partial (\beta \lambda_{\text{max}}^*)}{\partial z}$ at $z^* = [c^{-1}]_{1 \times n}$ is less than 1, i.e. $\beta \lambda_{\text{max}}^* < 1$, then the matrix point $x^* = [c^{-1}]_{1 \times n}$ should be an asymptotically stable fixed point of the DA-EM algorithm. Thus, the necessary condition for that, $z^* = [c^{-1}]_{1 \times n}$ is not an asymptotically stable fixed point of the DA-EM algorithm, is $\beta \lambda_{\text{max}}^* > 1$. Since $\lambda_{\text{max}}^*$ is the spectral radius and then $\lambda_{\text{max}}^* > 0$, we can obtain that $\beta \geq (\lambda_{\text{max}}^*)^{-1}$. Thus, the proof is completed. □

In practice, $x^* = [c^{-1}]_{1 \times n}$ should be avoided to be used in the DA-EM algorithm. This is because it may output all cluster centers $\mu_i = \bar{x}$, $i$ with the mass center $\bar{x}$, which is not interesting clustering results. According to Theorem 2, to avoid $\mu_i = \bar{x}$, we output the DA-EM algorithm, we need to choose the initialization bound $\beta_{\text{bound}} = (\lambda_{\text{max}}^*)^{-1}$ of the valid annealing parameter $\beta$. 


with $\beta \geq \beta_{\text{bound}}$. This becomes an important theoretical rule for initialization selection of the annealing parameter $\beta$.

In the paper, we consider two main problems. One is the annealing parameter selection of the DA-EM algorithm. Another one is the self-annealing behavior of the EM algorithm. The first one had been solved by Theorem 2 which provides a theoretical rule for the parameter selection of $\beta$. We next consider the second problem that is the self-annealing behavior of the EM. Since the EM is a special case of the DA-EM with the annealing parameter $\beta = 1$, we can focus the annealing properties of the DA-EM on the special case of $\beta = 1$. We next create a theorem to represent the relationship between the spectral radius of the Jacobian matrix $\frac{\partial g^{(t)}(z)}{\partial z}$ at $z^{*} = [c^{-1}]_{s \times n}$ and the annealing parameter $\beta$.

**Theorem 3.** Let $\lambda_{JM}^{\beta}$ be the spectral radius of the Jacobian matrix $\frac{\partial g^{(t)}(z)}{\partial z}$ at $z^{*} = [c^{-1}]_{s \times n}$ for the DA-EM. Then $\lambda_{JM}^{\beta} \geq \beta$.

**Proof.** By Lemma 1, we have that $\frac{\partial g^{(t)}(z)}{\partial z}$ is a symmetric semi-definite matrix. Therefore, the spectral radius of $\frac{\partial g^{(t)}(z)}{\partial z}$ is equal to the largest eigenvalue of the matrix $\frac{1}{\beta} (A(T)^{T} A(T))$, which is also equal to the largest eigenvalue of the matrix $\frac{1}{\beta} (A(T)^{T} A(T))$. Moreover, we have that $\frac{1}{\beta} (A(T)^{T} A(T)) = \frac{1}{\beta} \begin{pmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{pmatrix}$, where $L_{11} = 2n, L_{12} = 0, L_{13} = 0, L_{21} = L_{21}, L_{12} = L_{22} = 2\mathcal{N}_{n},$ and $L_{23} = 2\sqrt{2} \sum_{i=1}^{n} \frac{1}{s_{x,y}}$, $L_{31} = \sum_{i=1}^{n} \frac{1}{s_{x,y}}$, $L_{32} = \sum_{i=1}^{n} \frac{1}{s_{x,y}}$, $L_{33} = \sum_{i=1}^{n} \frac{1}{s_{x,y}}$. We know that $\lambda_{\max}(R) = \max_{x \neq 0} \frac{x^{T} R x}{x^{T} x}$, where the matrix $R$ is a symmetric matrix. In particular, set $x = e(0)$, for which $e(0)$ represents a vector, in which all elements are zero except that the $i$th element is one. Obviously, we know that $\max_{x \neq 0} \frac{x^{T} R x}{x^{T} x} \geq (s_{x,y})_{ii}$. In other words, $\max_{x \neq 0} \frac{x^{T} R x}{x^{T} x} \geq (s_{x,y})_{ii}$. Therefore, we have that $\lambda_{JM}^{\beta} \geq \beta$. The proof is completed. \hfill $\square$

According to Lemma 1 and the proof of Theorem 3, the spectral radius $\lambda_{JM}^{\max}$ or $\lambda_{JM}^{\beta}$ of the Jacobian matrix $\frac{\partial g^{(t)}(z)}{\partial z}$ at the matrix point $z^{*} = [c^{-1}]_{s \times n}$ of equal posterior probabilities can be calculated by $A^{T} A$ or $A^{T} A$. We next analyze the computational complexity of $A^{T}$ and $A^{T} A$. Suppose that the data number is $n$ and the feature dimension is $s$. According to Lemma 1, we have that

$$A = \begin{bmatrix} \sqrt{2} & \ldots & \sqrt{2} \\ \sqrt{2} \sigma_{s-1}^{-1/2} (x_{1} - x) & \ldots & \sqrt{2} \sigma_{s-1}^{-1/2} (x_{n} - x) \\ \text{vec}(\sigma_{s-1}^{-1/2} (x_{1} - x)) & \ldots & \text{vec}(\sigma_{s-1}^{-1/2} (x_{n} - x)) \\ (x_{1} - x)^{T} \sigma_{s-1}^{-1/2} & \ldots & (x_{n} - x)^{T} \sigma_{s-1}^{-1/2} \end{bmatrix}$$

and so the matrix $A$ has the dimension $(1 + s + s^{2}) \times n$, i.e. $A_{(1+s+s^{2}) \times n}$. It means that the computational complexity for computing the matrix $A^{T} A$ is $O(n^{2} (1 + s + s^{2})) = O(n^{2} s^{2})$. On the other hand, since the Jacobian matrix $\frac{\partial g^{(t)}(z)}{\partial z}$ is $A^{T} A$, a symmetric semi-definite matrix, the spectral radius of $\frac{\partial g^{(t)}(z)}{\partial z}$ is equal to the spectral radius of the matrix $\frac{1}{\beta} (A(T)^{T} A(T))$. In the proof of Theorem 3, we have $\frac{1}{\beta} (A(T)^{T} A(T)) = \frac{1}{\beta} \begin{pmatrix} 1 & 0 & 0 \\ 0 & L_{i} & \frac{L_{i}^{2}}{\beta} \\ 0 & \frac{L_{i}^{2}}{\beta} & \frac{L_{i}^{3}}{\beta^{2}} \end{pmatrix}$. From the forms of $L_{12}$ and $L_{23}$, we find that the computational complexity of $L_{12}$ and $L_{23}$ is $O(s^{2})$ and $O(s^{3})$, respectively. Thus, the computational complexity for computing the matrix $A^{T} A$ should be $O(s^{2} + s^{4}) = O(s^{4})$.

4. **Numerical examples and experiments.**

In this section, we carry out numerical and real experiments to demonstrate these theoretical results of the DA-EM algorithm for Gaussian mixtures by using the Jacobian matrix analysis in Section 3. We also further demonstrate the usefulness of the annealing parameter in the DA-EM algorithm. For all examples, we assign to increase the value of parameter $\beta$ with 1.01 times, i.e. $\beta_{\text{new}} = \beta_{\text{old}} \times 1.01$. In all examples, $\lambda_{\max}$ denotes the spectral radius of the matrix $\frac{\partial g^{(t)}(z)}{\partial z}$. $\lambda_{JM}^{\beta}$ denotes the spectral radius of the Jacobian matrix $\frac{\partial g^{(t)}(z)}{\partial z}$ of the DA-EM algorithm at $z^{*} = [c^{-1}]_{s \times n}$. We demonstrate the following five results: 1) The DA-EM algorithm gives better performance than the EM algorithm; 2) The DA-EM algorithm really has different theoretical lower bounds $\beta_{\text{bound}}$ for the valid annealing parameter initial $\beta_{0}$ from different data sets, and it can be estimated by our proposed theoretical rule with $\beta_{\text{bound}} = (\lambda_{JM}^{max})^{-1}$ from Theorem 2. That is, our theoretical lower bound for the valid initialization of the annealing parameter $\beta$ is useful in practice when apply the DA-EM algorithm; 3) $z^{*} = [c^{-1}]_{s \times n}$ will be outputted by the DA-EM algorithm when $\beta_{0} < (\lambda_{JM}^{max})^{-1}$; 4) $\lambda_{JM}^{\beta} = \lambda_{JM}^{\max}$ should be always larger than 1 when $\beta = 1$. That is, $z^{*} = [c^{-1}]_{s \times n}$ will be not outputted by the EM algorithm for most data sets, meaning that the EM always has a self-annealing property; and 5) The result from Theorem 1 can be applied in analyzing the convergence rate of the DA-EM algorithm.
Fig. 1. Clustering results of the EM and DA-EM algorithms.

(d) \( \beta^{(0)} = 0.7 > \beta^{\text{bound}}, \text{err} = 0\% \)

(e) \( \beta^{(0)} = 0.85 > \beta^{\text{bound}}, \text{err} = 0\% \)

(f) \( \text{EM}(\beta^{(0)} = 1), \text{err} = 0\% \)

set from this mixture model is shown in Fig. 1(a) where one cluster is marked by "•" with the cluster center \( \mu_1 = (0,2)^T \), another cluster is marked by "*" with the cluster center \( \mu_2 = (0,0)^T \), and the third cluster is marked by "*" with the cluster center \( \mu_3 = (0,-2)^T \). We can see from Fig. 1(a) that the three clusters in the data set are separated. We calculate the theoretical lower bound with \( \beta^{\text{bound}} = 0.6932 \) for the data set in Fig. 1(a).

To compare the performance of EM and DA-EM, we give initials for both algorithms with fixed mean values (cluster centers) with \( \mu_1^{(0)} = (1,0)^T \), \( \mu_2^{(0)} = (0,0)^T \), \( \mu_3^{(0)} = (-1,0)^T \), and equal mixing proportions with \( \alpha_1^{(0)} = \alpha_2^{(0)} = \alpha_3^{(0)} \), and identity covariance matrices with \( \Sigma_1^{(0)} = \Sigma_2^{(0)} = \Sigma_3^{(0)} = I \). The initializations of EM and DA-EM are shown in Fig. 1(b). Then we apply DA-EM to the data set in Fig. 1(a). We initialize the annealing parameter \( \beta^{(0)} \) with different values, e.g. \( \beta^{(0)} = 0.69 < \beta^{\text{bound}} \) and \( \beta^{(0)} = 0.7 > \beta^{\text{bound}} \), where the results are listed in Fig. 1(c)–(e). We also calculate the error rates (err) for each result shown in Fig. 1. From the experimental result, we can see that if we initialize the annealing parameter with a value less than theoretical bound then the mass center is always outputted by the DA-EM algorithm. If we initialize \( \beta^{(0)} \) as a value bigger than \( \beta^{\text{bound}} \), a good clustering result can be obtained as shown in Fig. 1(d) and (e). We also implement the EM algorithm (i.e. with \( \beta^{(0)} = 1 \)) on the data set and get a good clustering result as shown in Fig. 1(f). It is supposed that the data set in Fig. 1(a) has separated three clusters so that both EM and DA-EM can better handle this kind of data sets.

Next, we use another mixture model of three multivariate normal components in which the mixing proportions, mean values and covariance matrices are known and fixed by setting \( \alpha_1 = 0.33, \alpha_2 = 0.33, \alpha_3 = 0.33, \mu_1 = (0,-1)^T, \mu_2 = (0,0)^T, \mu_3 = (0,1)^T, \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0.2 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.1 \end{pmatrix}, \Sigma_3 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.1 \end{pmatrix} \).

The generated data set from this mixture model is shown in Fig. 2(a) where the three clusters in the data set are overlapped. We calculate the theoretical lower bound with \( \beta^{\text{bound}} = 0.8388 \) for the data set in Fig. 2(a). We also initialize the algorithms with fixed mean values (cluster centers) of \( \mu_1^{(0)} = (1,0)^T \), \( \mu_2^{(0)} = (0,0)^T \), \( \mu_3^{(0)} = (-1,0)^T \), and equal mixing proportions with \( \alpha_1^{(0)} = \alpha_2^{(0)} = \alpha_3^{(0)} \), and identity covariance matrices with \( \Sigma_1^{(0)} = \Sigma_2^{(0)} = \Sigma_3^{(0)} = I \). We calculate the error rate (err) for each result shown in Fig. 2. The clustering results obtained by EM-DA are listed in Fig. 2(c)–(e). It is clearly that the mass center is always outputted by the DA-EM algorithm due to the badly initialized annealing parameter as shown in Fig. 2(c) with \( \text{err} = 66.7\% \). If we give a proper initial annealing parameter value, e.g. \( \beta^{(0)} = 0.85 > \beta^{\text{bound}} \), then a better clustering result is obtained by EM-DA with \( \text{err} = 36\% \). However, if we initialize the annealing parameter with a large value, e.g. \( \beta^{(0)} = 0.9 > \beta^{\text{bound}} \), then the DA-EM algorithm may obtain poor results with \( \text{err} = 47.67\% \), as shown in Fig. 2(e). This is because a larger initial annealing parameter value \( \beta^{(0)} \) corresponds to a smaller range for annealing. For most cluster-overlapped data sets, it is not good to initialize the annealing parameter with too larger value. At last, the clustering results obtained by EM (i.e. \( \beta^{(0)} = 1 \)) are shown in Fig. 2(f). We find that EM gets the worst clustering results as shown in Fig. 2(a) with \( \text{err} = 63\% \) which is always higher than error rates from DA-EM as \( \beta^{(0)} < 1 \).

We next consider a mixture model of three multivariate normal components in which the mixing proportions, mean values and covariance matrices are known and fixed by setting \( \alpha_1 = 0.2, \alpha_2 = 0.3, \alpha_3 = 0.5, \mu_1 = (1.1,2)^T, \mu_2 = (5.3,0.5)^T, \mu_3 = (2.6,5)^T, \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0.2 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.1 \end{pmatrix}, \Sigma_3 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.1 \end{pmatrix} \).

The generated data set from this mixture model is shown in Fig. 3(a) where one cluster is marked by "•" with the cluster center \( \mu_1 = (2.6,5)^T \), another cluster is marked by "*" with the cluster center \( \mu_2 = (5.3,0.5)^T \), and the third cluster is marked by "*" with the cluster center \( \mu_3 = (1,1,2)^T \). We calculate \( \alpha^{\max} \) and the theoretical lower bound \( \beta^{\text{bound}} \) of \( \beta \) by using the theoretical
We obtain rule $\beta^{\text{bound}} = (\lambda_{\text{max}})^{-1}$. We obtain $\beta^{\text{bound}} = 0.4108$ for the data set of Fig. 3(a). After we get the $\beta^{\text{bound}}$, we initialize the DA-EM algorithm with several different initials $\beta^{(0)}$. These DA-EM clustering results are shown in Fig. 3(b)–(f). When we give $\beta^{(0)} = 0.1 < \beta^{\text{bound}}$, the clustering result of DA-EM has only one cluster center with the mass center $\bar{x}$ as shown in Fig. 3(b). The clustering result with the mass center $\bar{x}$ for DA-EM under $\beta^{(0)} = 0.4 < \beta^{\text{bound}}$ is also obtained as shown in Fig. 3(c). However, if we give $\beta^{(0)} = 0.42$ that is a little bit larger than $\beta^{\text{bound}}$, then the DA-EM algorithm still has worse clustering results, but with two cluster centers (“•” and “▲”) as shown in Fig. 3(d). If we give $\beta^{(0)} = 0.43$, then the DA-EM algorithm has better clustering results with three cluster centers as shown in Fig. 3(e). Similar good clustering results are also obtained even for the larger $\beta^{(0)} = 0.7$ as shown in Fig. 3(f). In general, we find that, if initial $\beta^{(0)}$ is smaller than theoretical lower bound $\beta^{\text{bound}} = 0.4108$, then the mass center $\bar{x}$ is always outputted
(a) $\mu_1 = (0,0,1)^T$, $\mu_2 = (1,0,0)^T$, $\mu_3 = (0,1,0)^T$, $\beta_{\text{bound}} = 0.4175$

(b) $\mu_1 = (0,0,2)^T$, $\mu_2 = (2,0,0)^T$, $\mu_3 = (0,2,0)^T$, $\beta_{\text{bound}} = 0.277$

(c) $\mu_1 = (0,0,3)^T$, $\mu_2 = (3,0,0)^T$, $\mu_3 = (0,3,0)^T$, $\beta_{\text{bound}} = 0.234$

(d) $\mu_1 = (0,0,8)^T$, $\mu_2 = (8,0,0)^T$, $\mu_3 = (0,8,0)^T$, $\beta_{\text{bound}} = 0.218$

Fig. 4. $\beta_{\text{bound}}$ for different data sets.

by the DA-EM algorithm, but if the initial $\beta^{(0)}$ is larger than theoretical lower bound $\beta_{\text{bound}}$, even not too large, then the DA-EM algorithm can find clustering results other than the mass center $\bar{x}$. However, if the initial $\beta^{(0)}$ is only a little bit larger than theoretical bound, then a worse clustering result may be obtained as shown in Fig. 3(d). This means that an estimate of the theoretical bound for the annealing parameter $\beta^{(0)}$ is actually important in practice.

**Example 2.** In this example, we discuss the influence of different data sets on the theoretical bound of the annealing parameter. We consider mixture models of three multivariate normal components in which the mixing proportions and covariance matrices are fixed as follows: $\alpha_1 = 0.1$, $\alpha_2 = 0.7$, $\alpha_3 = 0.2$, $\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$. We give different data sets generated from different mean values where we increase distances between the three clusters of the data set as shown in Fig. 4(a)–(d). We calculate theoretical lower bound $\beta_{\text{bound}}$ shown in Fig. 4(a)–(d). We clearly see that the theoretical low bound $\beta_{\text{bound}}$ becomes smaller as the distance between clusters become larger.

We next consider several higher dimensional data sets generated from mixture models of multivariate normal distributions. Different mixing proportions, mean values and covariance matrices of data sets are listed in Table 1. It is noted that, for the data set Data_4_2, it represents the data set is generated from a four-dimensional, two-component Gaussian mixture model. Different theoretical bounds $\beta_{\text{bound}}$ for the annealing parameter are shown in Table 2. From results, we can see that theoretical bounds for the annealing parameter may demonstrate different behavior of different data sets.

**Example 3.** In this example, we consider several real data sets from the UCI Repository of Machine Learning Databases [25]. We test the eight typical data sets for our experiments and list their results of the DA-EM algorithm in Table 3. From Table 3, it clearly indicates that most theoretical lower bounds $\beta_{\text{bound}}$ for these real data sets are quite small. Note that $\lambda_{\text{max}} = \lambda_{\text{JM}}^*$ as $\beta = 1$. That is, $\lambda_{\text{max}}\geq 1$ by Theorem 3 in Section 3. From Table 3, these $\lambda_{\text{max}}$ are actually larger than 1, and so for all these data sets, the matrix point $\mathbf{z} = [c^{-1}]_{k\times k}$ should not be a stable fixed point of the EM algorithm for Gaussian mixtures. Furthermore, if the initial $\beta^{(0)}$ is lower than the theoretical lower bounds $\beta_{\text{bound}}$, then the mass center $\bar{x}$ is always outputted by the DA-EM algorithm, and so the posterior probability becomes uniform with each point $x_i$ equally contributed to each component of the mixture. In this case, the DA-EM algorithm will be unable to escape from the point $\mathbf{z} = [c^{-1}]_{k\times k}$ with outputting the mass center $\bar{x}$. For example, we implement the DA-EM algorithm for Haberman’s Survival data set with the initial $\beta^{(0)} = 0.13 < \beta_{\text{bound}} = 0.134$. We finally get the parameter estimation results as follows:

$$\alpha_1 = \alpha_2 = 0.5, \quad \mu_1 = \mu_2 = (52, 62.85, 4.03)^T,$$

$$\sum_1 = \sum_2 = \begin{bmatrix} 116.3 & 3.13 & -4.89 \\ 3.13 & 10.5 & -0.088 \\ -4.89 & -0.088 & 51.5 \end{bmatrix}.$$
### Table 1

Different generated data sets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\alpha_1$</th>
<th>$\alpha_2$</th>
<th>$\alpha_3$</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\mu_3$</th>
<th>$\Sigma_1$</th>
<th>$\Sigma_2$</th>
<th>$\Sigma_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data_4_2</td>
<td>0.3</td>
<td>0.7</td>
<td>-</td>
<td>(1, 2, 0, 1)$^T$</td>
<td>(1, 0, 0, 1)$^T$</td>
<td>-</td>
<td>$\begin{bmatrix} 0.1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 0.1 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0.1 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 0.1 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1.0 &amp; 0 &amp; 0 \ 0 &amp; 0.1 &amp; 0 \ 0 &amp; 0 &amp; 0.5 \ 0 &amp; 0 &amp; 0 \end{bmatrix}$</td>
<td>-</td>
</tr>
<tr>
<td>Data_5_3</td>
<td>0.3</td>
<td>0.2</td>
<td>0.5</td>
<td>(1, 2, 0, 1)$^T$</td>
<td>(1, 0.5, 0, 1)$^T$</td>
<td>(1, 0.1, 1, 1)$^T$</td>
<td>$\begin{bmatrix} 0.1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 0.1 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0.1 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 0.2 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1.0 &amp; 0 &amp; 0 \ 0.5 &amp; 0 &amp; 0 \ 0 &amp; 0.1 &amp; 0 \ 0 &amp; 0 &amp; 0.3 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 2.0 &amp; 0 &amp; 0 \ 0 &amp; 0.1 &amp; 0 \ 0 &amp; 0 &amp; 0.2 \ 0 &amp; 0 &amp; 0 \end{bmatrix}$</td>
</tr>
<tr>
<td>Data_6_2</td>
<td>0.5</td>
<td>0.5</td>
<td>-</td>
<td>(1, 0, 0, 1, 0)$^T$</td>
<td>(0, 2, 1, 0, 2)$^T$</td>
<td>-</td>
<td>$\begin{bmatrix} 0.1 &amp; 0 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 0.1 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0.1 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 0.1 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 0 &amp; 0 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1.0 &amp; 0 &amp; 0 \ 0 &amp; 1 &amp; 0 \ 0 &amp; 0 &amp; 1 \ 0 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0 \end{bmatrix}$</td>
<td>-</td>
</tr>
<tr>
<td>Data_6_3</td>
<td>0.1</td>
<td>0.2</td>
<td>0.7</td>
<td>(1, 3, 0, 0, 1)$^T$</td>
<td>(0.1, 2, 3, 0, 0)$^T$</td>
<td>(1, 1, 1, 1, 1)$^T$</td>
<td>$\begin{bmatrix} 0.1 &amp; 0 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 2 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 1 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 1 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 1.0 &amp; 0 &amp; 0 \ 1 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 1 \ 0 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.1 &amp; 0 &amp; 0 &amp; 0 &amp; 0 \ 0 &amp; 0.2 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0.1 &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 0.1 \ 0 &amp; 0 &amp; 0 &amp; 0 &amp; 0 \end{bmatrix}$</td>
</tr>
</tbody>
</table>
That is, the DA-EM algorithm outputs the mass center $\bar{x}$ and gives not interesting clustering results.

**Example 4.** In this example, we consider another three real data sets to further analyze the theoretical lower bounds $\beta_{\text{bound}}$. The first data set is the well-known Iris data [25,26]. The Iris data set is composed of three clusters of flowers. There are four measurements which are sepal length, sepal width, petal length, and petal width from each of 3 species of Iris Setosa, Iris Versicolor and Iris Virginica. The Setosa plants are well separated from the others, but there are some overlaps between two the other species. The scatter plot for the four-dimensional Iris data set is shown in Fig. 5 where Fig. 5(a) uses the triangle, star and solid circle symbols to represent the Setosa, Versicolor and Virginica, respectively, in the three dimensions, and Fig. 5(b) demonstrates two-dimensional plot from the Wikipedia. The second real data set is from Olivetti Research Laboratory (ORL) database of faces that contains 400 different images from 40 individuals, where each individual has 10 different images [27]. The images were taken at different times, different condition of lighting, different variations of face expression (open/closed eyes, smiling/not smiling), and different facial details (glasses/no glasses). All images were taken against a dark homogeneous background with a tolerance for some tilting and rotation of the face. We only consider 100 face images from 10 individuals of the ORL database with 1024 attributes (feature components), as shown in Fig. 6. The third data set is the Waveform data set [25]. The Waveform data set contains 5000 instances from 3 classes of waves. Each of 3 classes accounts for 33 percent of the whole data set. Each wave is described by 21 attributes with continuous values between 0 and 6, and no missing attribute values in this data set. This data set is used in classification task due to the label. We use the three data sets to demonstrate the usefulness of the theoretical lower bound $\beta_{\text{bound}}$. We mention that, before we find the theoretical lower bound $\beta_{\text{bound}}$ for ORL face images, we first implement the FR-FCM [28] for ORL face images with feature reduction. These theoretical lower bounds $\beta_{\text{bound}}$ are shown in Table 4. From Table 4, we find that the theoretical lower bound $\beta_{\text{bound}}$ for ORL face images is small with 0.0292. That is, it is not easy to get only one cluster, except that you have assigned a high starting initial $\beta^{(0)}$ larger than 0.4799 or 0.6416. To observe this behavior, we use the Iris data set for demonstration, as shown in Fig. 7. From Fig. 7, we find that, if the initial $\beta^{(0)}$ is assigned as 0.46 (~0.4799), then we get only one cluster with the ec is 100. However, if the initial $\beta^{(0)}$ is assigned as larger than 0.4799, such as 0.5, 0.6, 0.7, then we can get good clustering results with ec = 17.

![Iris data set](image1)

![Iris data plot from Wikipedia](image2)

Fig. 5. Scatter plot for the Iris data set.

**Table 2**

$\beta_{\text{bound}}$ for different data sets.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>$\beta_{\text{bound}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data_4_2</td>
<td>0.392</td>
</tr>
<tr>
<td>Data_5_3</td>
<td>0.51</td>
</tr>
<tr>
<td>Data_6_2</td>
<td>0.203</td>
</tr>
<tr>
<td>Data_6_3</td>
<td>0.243</td>
</tr>
</tbody>
</table>

**Table 3**

Experimental results for real data sets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No. of Samples</th>
<th>No. of Features</th>
<th>No. of Classes</th>
<th>$\lambda_{\text{max}}$</th>
<th>$\beta_{\text{bound}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haberman’s Survival</td>
<td>306</td>
<td>3</td>
<td>2</td>
<td>7.452</td>
<td>0.134</td>
</tr>
<tr>
<td>Breast Cancer Wisconsin (Original)</td>
<td>699</td>
<td>9</td>
<td>8</td>
<td>9.711</td>
<td>0.303</td>
</tr>
<tr>
<td>Cloud</td>
<td>1024</td>
<td>10</td>
<td>3</td>
<td>50.074</td>
<td>0.01997</td>
</tr>
<tr>
<td>Wine</td>
<td>1599</td>
<td>11</td>
<td>3</td>
<td>25.11</td>
<td>0.039</td>
</tr>
<tr>
<td>Wine-quality-red</td>
<td>4898</td>
<td>11</td>
<td>4</td>
<td>312.291</td>
<td>0.00032</td>
</tr>
<tr>
<td>Wine-quality-white</td>
<td>178</td>
<td>13</td>
<td>3</td>
<td>10.479</td>
<td>0.0955</td>
</tr>
<tr>
<td>SPECTF Heart</td>
<td>187</td>
<td>22</td>
<td>2</td>
<td>8.503</td>
<td>0.118</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>2</td>
<td>51.448</td>
<td>0.019</td>
</tr>
</tbody>
</table>
Example 5. In this example, we investigate the influence of the increasing factor of parameter $\beta$. We use the most-used Iris data set that had been demonstrated in Example 4. These clustering results with error count (ec) are shown in Table 5. We find that, if we assign to increase the value of parameter $\beta$ with a scale less than 1.025, i.e. $\beta^{(\text{new})} \leftarrow \beta^{(\text{old})} \times 1.025$, all ec are 17 that is good for DA-EM (or EM). However, if we assign to increase the value of parameter $\beta$ with a scale larger than 1.03, i.e. $\beta^{(\text{new})} = \beta^{(\text{old})} \times 1.03$, most of ec are not good, even more than 28. Totally, we recommend increasing the value of parameter $\beta$ with 1.01 times, i.e. $\beta^{(\text{new})} = \beta^{(\text{old})} \times 1.01$.

Example 6. In this example, we further investigate the influence of different initial values of $\beta$ on the convergence rate of the DA-EM algorithm. This also demonstrates that the results from Theorem 1 can be applied in analyzing the convergence rate of the DA-EM algorithm. We first set $\beta^{(0)} = 0.1$ in the DA-EM algorithm and then gradually increase the annealing parameter initial value $\beta^{(0)}$ as shown in Table 6. For each value $\beta^{(0)}$, we compute the spectral radius $\lambda^{\text{conv}}$ of the Jacobian matrix $\frac{\partial \theta^{(t)}}{\partial z^{(t)}}$ of the DA-EM algorithm evaluated at the convergence point by using Eq. (13) in Theorem 1. The experimental results with $\lambda^{\text{conv}}$ for the different initial value $\beta^{(0)}$ are listed in Table 6. The DA-EM algorithm had been interpreted as a mapping $z \rightarrow \theta^{(t)}(z)$ such that $z^{(t+1)} = \theta^{(t)}(z^{(t)})$ with an iteration sequence $\{z^{(t)}\}$. If the iterations terminate at a matrix point $z^*$, then the convergence rate can be well approximated by the ratio $r = \lim_{t \to \infty} \frac{\|z^{(t+1)} - z^*\|}{\|z^{(t)} - z^*\|}$. From the ratio $r$, we can see that the larger value of $r$ will have smaller convergence rate. Suppose that the mapping is differentiable in a neighborhood of $z^*$, we can obtain a Taylor expansion as $z^{(t+1)} - z^* \approx (z^{(t)} - z^*) \frac{\partial \theta^{(t)}}{\partial z^{(t)}} |_{z=z^*}$. Thus, the DA-EM algorithm should be a linear iteration with the rate according to the Jacobian matrix $\frac{\partial \theta^{(t)}}{\partial z^{(t)}} |_{z=z^*}$ at the convergence point $z = z^*$. Furthermore, from the Taylor expansion and induced matrix norm of $\frac{\partial \theta^{(t)}}{\partial z^{(t)}} |_{z=z^*}$, we can see that the ratio $r$ corresponding to the convergence rate should be related to the spectral radius of the Jacobian matrix $\frac{\partial \theta^{(t)}}{\partial z^{(t)}} |_{z=z^*}$. That is, the spectral radius $\lambda^{\text{conv}}$ of the Jacobian matrix $\frac{\partial \theta^{(t)}}{\partial z^{(t)}} |_{z=z^*}$ can be referred to as the convergence rate. We know that a larger value of $r$, which is corresponding a larger value of the spectral radius $\lambda^{\text{conv}}$ of the Jacobian matrix $\frac{\partial \theta^{(t)}}{\partial z^{(t)}} |_{z=z^*}$, implies slow convergence. From Table 6, we find that, different data sets have different influence behavior from temperature initial values $\beta^{(0)}$ on the convergence rate of the DA-EM algorithm.

5. Conclusions

The EM algorithm is a popular learning-based optimization method as it can efficiently deal with the maximum likelihood parameter estimation. On the other hand, EM for Gaussian mixtures is also popular as a clustering algorithm, but it cannot avoid a local optimal problem and depends on initializations, and so Ueda and Nakano [4] proposed the DA-EM algorithm to improve it. Even though the DA-EM algorithm may solve poor local optimum problem, it is not yet guaranteed to obtain the global optimum because the clustering results depend on the initial value of the annealing parameter $\beta$. In this paper, we constructed the Jacobian matrix of the DA-EM for Gaussian mixtures and then showed that these.
convergence behaviors of the DA-EM algorithm for Gaussian mixtures can be analyzed by the Jacobian matrix analysis. In theoretical analysis, we also get an important rule for the initialization selection of the annealing parameter $\beta$ such that we can guarantee the coincident clustering results with the mass center $\hat{x}$ not to be outputted by the DA-EM algorithm. Furthermore, we theoretically prove that, the EM algorithm for Gaussian mixtures actually exhibits the self-annealing behavior. In our experimental results, we demonstrate that our theoretical results are applicable in practice. Moreover, our theoretical analysis can be also used to observe the behavior of convergence rates of the DA-EM algorithm for Gaussian mixtures.

Acknowledgments

The authors would like to thank the anonymous referees for their helpful comments in improving the presentation of the paper. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Miin-Shen Yang’s Grant MOST 105-2118-M-033-004-MY2.

References


Fig. 7. Clustering results with different $\beta^{(0)}$ for Iris data set.

(a)$\beta^{(0)} = 0.46, ec = 100$

(b)$\beta^{(0)} = 0.5, ec = 17$

(c)$\beta^{(0)} = 0.6, ec = 17$

(d)$\beta^{(0)} = 0.7, ec = 17$
Jian Yu received the B.S. and M.S. degrees in mathematics and Ph.D. degree in applied mathematics from Peking University, Beijing, China, in 1991, 1994, and 2000, respectively. He is currently a Professor and Head of Institute of Computer Science at Beijing Jiaotong University and the director of the Beijing Key Lab of Traffic Data Analysis and Mining. His research interests include fuzzy clustering, machine learning, image processing, and pattern recognition.

Chaomurilige is currently pursuing a PhD degree in Computer Science at School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. She received the BE degree in Biomedical Engineering, the MS degree in Pattern Recognition and Artificial Intelligence from the Beijing Jiaotong University, in 2010 and 2012. Her main areas of research include fuzzy clustering, pattern recognition, and machine learning.

Miin-Shen Yang received the BS degree in mathematics from the Chung Yuan Christian University, Chung-Li, Taiwan, in 1977, the M.S degree in applied mathematics from the National Chiao-Tung University, Hsinchu, Taiwan, in 1980, and the PhD degree in statistics from the University of South Carolina, Columbia, USA, in 1989. In 1989, he joined the faculty of the Department of Mathematics in the Chung Yuan Christian University (CYCU) as an Associate Professor, where, since 1994, he has been a Professor. From 1997 to 1998, he was a Visiting Professor with the Department of Industrial Engineering, University of Washington, Seattle. During 2001–2005, he was the Chairman of the Department of Applied Mathematics in CYCU, and from 2001 to 2016, he was the Director of Chaplain’s Office in CYCU. Since 2012, he has been a Distinguished Professor of the Department of Applied Mathematics in CYCU. His research interests include fuzzy clustering, applications of statistics, neural fuzzy systems, pattern recognition, and machine learning.

Dr. Yang was an Associate Editor of the IEEE Transactions on Fuzzy Systems (2005–2011), and is an Associate Editor of the Applied Computational Intelligence & Soft Computing and Editor-in-Chief of Advances in Computational Research. He was awarded with 2008 Outstanding Associate Editor of IEEE Transactions on Fuzzy Systems, IEEE; 2009 and 2016 Outstanding Research Award of Chung Yuan Christian University; 2012–2018 Distinguished Professorship of Chung Yuan Christian University.