IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal
SPECIAL SECTION ON ROADMAP TO 5G: RISING TO THE CHALLENGE

Received June 10, 2019, accepted July 3, 2019, date of publication July 8, 2019, date of current version July 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927438

An Improved Minor Component Analysis Algorithm Based on Convergence Analysis of 5G Multi-Dimensional Signals

PENG CHEN[1], (Member, IEEE), LEI WU[2], (Member, IEEE), DEZHONG PENG[3], (Member, IEEE), XIAOYUE GONG[2], AND VICTOR HUGO C. DE ALBUQUERQUE[4], (Senior Member, IEEE)

[1]School of Computer and Software Engineering, Xihua University, Chengdu 610039, China
[2]School of Mathematical Science, University of Electronic Science and Technology of China Chengdu 611731, China
[3]College of Computer Science, Sichuan University, Chengdu 610065, China
[4]Graduate Program in Applied Informatics, University of Fortaleza, Fortaleza, Brazil

Corresponding author: Lei Wu (wulei@uestc.edu.cn)

This work was supported in part by the Science and Technology Program of Sichuan Province under Grant 2019YFH0075, Grant 2018GZDZ0030, Grant 2016CC0016, Grant 2015GZ0119, and Grant 2012GZX085.

ABSTRACT With the availability of 5G, multi-dimensional signals' characteristic analysis becomes more challenging. Minor component analysis (MCA) is a useful data analysis approach to estimate the important features of multi-dimensional signals. MCA neural network approaches can extract minor components from real-time data streams not only adaptively but also online. The rapid convergence of MCA neural network approaches is valuable in practical applications. We propose a deterministic discrete time-based method to analyze the convergence performance of MCA neural network algorithms and reveal the effects of initial weight vectors' norm in MCA algorithms on convergence speed. Then, based on the convergence analysis, a new MCA algorithm is proposed to achieve faster convergence than existing MCA algorithms, which are proven by both the theoretical analysis and simulation results.

INDEX TERMS 5G, minor component analysis (MCA), neural networks, Eigen value, deterministic discrete time (DDT) method.

I. INTRODUCTION

5G technology brings drastic improvements as well as challenges almost everywhere. With much higher network speed and wider bandwidth, one of the main challenge exists in multi-dimensional signals' characteristic analysis since signals are far more complicated. Minor component analysis (MCA) is a powerful data analysis method and widely adopted in many practical applications, as shown in Fig. 1 for 5G. It is crucial for 5G multi-dimensional signals that minor component should be paid more attention even than principal components. Neural networks can be exploited to achieve online estimation of the minor components of a dataset [1]–[3]. Compared with matrix approaches (e.g., eigenvalue decomposition), the neural network approach possesses lower computational complexity.

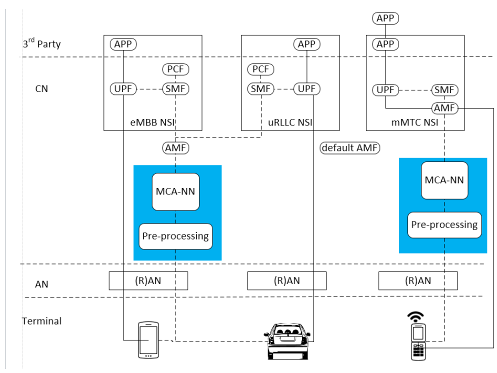


FIGURE 1. 5G Architecture with signal characteristic analysis (MCA).

Many minor component analysis (MCA) algorithms based on neural networks have been developed in [6], [8], [11], [13], [21], [25], [26], [48]. For example, [8] used a simple linear neuron and the anti-Hebbian rule to perform MCA.

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Imran.

Xu *et al.* [6] proposed a normalized MCA neural network learning algorithm called OJAn and applied it to curve and surface fitting. Douglas *et al.* [11] presented an MCA learning algorithm with a self-stabilizing property, which can guarantee that the weight vector approaches the minor component with the unit norm. Several MCA learning algorithms were recently proposed by Ouyang *et al.* [13], Cirrincione *et al.* [21] and Feng *et al.* [25], [26]. Evidently, interest in the dynamic analysis of MCA learning algorithms is increasing [21], [27], [29], [30], [47].

As a powerful tool for data analysis, neural networks has been successfully used in many practical applications [4], [5], [7], [9], [10], [12], [14]–[20], [22]–[24], [28], [33], [36], [37], [40], [43]–[45], in which neural network learning algorithms play an essential role. MCA neural network algorithms are usually represented as Stochastic Discrete Time (SDT) systems. However, studying the dynamics of these SDT systems is difficult. To analyze indirectly the dynamics of stochastic learning algorithms, [31], [32], [34], [35], [38], [39], [46] used a so-called Deterministic Discrete Time (DDT) method. The method transforms the original versions of SDT algorithms into corresponding DDT systems, and the dynamical properties of the original SDT systems are indirectly obtained by analyzing these DDT systems.

The fast convergence of MCA neural network algorithms is valuable to practical applications. Hence, analyzing the performance of MCA algorithms and improving their convergence speeds are two important and interesting topics. In adaptive MCA learning algorithms, the initial weight vector and the learning rate are two important parameters that affect algorithm performance. Generally, a high learning rate accelerates the convergence of algorithms, and a low learning rate improves the numerical stability of adaptive algorithms. However, the relations between convergence speed and initial weight vector have not been fully studied in existing research. Therefore, we propose a convergence analysis method based on DDT and disclose relations between convergence performance of MCA algorithms and initial weight vectors of these algorithms. This theoretical finding can establish useful guidelines for selecting initial weight vectors to achieve fast convergence of MCA algorithms. Then, based on such convergence analysis, we propose an improved MCA algorithm with fast convergence.

This paper is organized as follows. Section 2 presents an analysis of the convergence speeds of MCA learning algorithms. Section 3 provides a comparison of the convergence speeds of different MCA algorithms. Section 4 presents the improved MCA algorithm based on convergence analysis and Section 5 comes the conclusion.

II. ANALYSIS OF CONVERGENCE SPEEDS

Since the pioneering work of [8] on MCA neural network algorithms, many adaptive MCA neural network learning algorithms have been developed [11], [26]. Almost all of these MCA algorithms share the following unified form

$$z(t) = \Theta^T(t)y(t) \tag{1a}$$

$$\Theta(\mathrm{t+1}) = \Theta(\mathrm{t}) - \mu[\mathrm{f}(\Theta(\mathrm{t}))\cdot \mathrm{z(t)y(t)} - \mathrm{g}(\Theta(\mathrm{t}))\cdot\Theta(\mathrm{t})] \tag{2a}$$

where $\mu > 0$ denotes the learning rate. In (1a), $y(t) \in R^m$, $\Theta(t) \in R^m$, and $z(t)$ stands for the input vector, weight vector, and output of the neurons, respectively. In (1b), both $f(\Theta(t))$ and $g(\Theta(t))$ are two scalar functions of the weight vector $\Theta(t)$.

By incorporating conditional expectation operator $E\{\Theta(t)/\Theta(0), y(i), i < t\}$ on (1b) and using the conditional expected value in next iteration [34], we obtain a DDT system as follows:

$$\Theta(t+1) = \Theta(t) - \mu[f(\Theta(t))\cdot R\cdot\Theta(t) - g(\Theta(t))\cdot\Theta(t)], \tag{2}$$

where $R = E[y(t)y^T(t)]$ is the correlated matrix of inputs.

Correlation matrix R is usually non-negative and definite. Without loss of generality, we let $\sigma_1, \ldots, \sigma_m$ be all the eigenvalues of R ordered as $\sigma_1 > \ldots > \sigma_m \geq 0$. We let $a_j(j = 1, 2, \ldots, m)$ stand for a unit eigenvector of correlation matrix R associated with eigenvalue σ_j. Given that R is a symmetric matrix, $\{a_j | j = 1, \ldots, m\}$ is an orthonormal basis of R^m. Thus, for all $t \geq 0$, $\Theta(t)$ can be described as:

$$\Theta(t) = \sum_{j=1}^{m} \phi_j(t)a_j \tag{3}$$

where $\phi_j(t)$ $(j = 1, 2, \ldots, m)$ are constants.

Evidently,

$$R\Theta(t) = \sum_{j=1}^{m} \sigma_j\phi_j(t)a_j. \tag{4}$$

By substituting (3) and (4) in (2), we obtain

$$\sum_{j=1}^{m}\phi(t+1)a_j = \sum_{j=1}^{m}\phi_j(t)a_j - \mu\left[f(\Theta(t))\sum_{j=1}^{m}\sigma_j\phi_j(t)a_j - g(\Theta(t))\sum_{j=1}^{m}\phi_j(t)a_j\right]$$

where $\{a_1, a_2, \ldots, a_m\}$ are mutually orthogonal.

So we have:

$$\phi_\mathrm{j}(t+1) = [1 - \mu f(\Theta(t))\cdot\sigma_j + \mu g(\Theta(t))\cdot\phi_j(t), \quad (j = 1, 2, \ldots, m)] \tag{5}$$

for all $t \geq 0$.

To ensure the convergence of algorithms, learning rate μ is regarded as a small constant in MCA learning algorithms. For analysis convenience, we assume that in (2), learning step μ is sufficiently small to satisfy the conditions below:

$$\mu\sigma_1 \leq 1 \text{ and } \mu\cdot[\sigma_1 f(\Theta(t)) - g(\Theta(t))] < 1, \tag{6}$$

for all $t \geq 0$. Then,

$$1 - \mu\sigma_j f(\Theta(t)) + \mu g(\Theta(t)) > 0, \quad (j = 1, 2, \ldots, m), \tag{7}$$

for all $t \geq 0$.

We establish

$$q_j(t)=\frac{1-\mu f(\Theta(t))\cdot\sigma_j+\mu g(\Theta(t))}{1-\mu f(\Theta(t))\cdot\sigma_m+\mu g(\Theta(t))}, \quad (j=1,2,\ldots,m-1). \tag{8}$$

From (7), we have

$$0 < q_1(t) < q_2(t)\ldots < q_{m-1}(t) < 1, \tag{9}$$

for all $t \geq 0$. Then, from (5) and (9), we have

$$\begin{aligned}\frac{\phi_j^2(t+1)}{\phi_m^2(t+1)} &= \left[\frac{1-\mu f(\Theta(t))\cdot\sigma_j+\mu g(\Theta(t))}{1-\mu f(\Theta(t))\cdot\sigma_m+\mu g(\Theta(t))}\right]^2\cdot\frac{\phi_j^2(t)}{\phi_m^2(t)}\\ &= q_j^2(t)\cdot\frac{\phi_j^2(t)}{\phi_m^2(t)} \geq q_1^2(t)\cdot\frac{\phi_j^2(t)}{\phi_m^2(t)} \geq \prod_{i=0}^{t} q_1^2(i)\cdot\frac{\phi_j^2(0)}{\phi_m^2(0)},\\ &\quad (j=1,2,\ldots,m-1),\end{aligned} \tag{10}$$

for all $t \geq 0$.

To evaluate the convergence speeds of MCA neural network algorithms, we define direction cosine $\Theta(t)$ as

$$\theta(t)=\frac{\left|\Theta^T(t)a_m\right|}{\|\Theta(t)\|},$$

where a_m is the minor component, i.e., the unit eigenvector corresponding to the minimum eigenvalue of the correlation matrix of the dataset. If and only if $\theta(t)$ approaches 1 as $t \to \infty$, weight vector $\Theta(t)$ converges to the direction of minor component a_m. Subsequently, we examine which factors affect the convergence speed on direction cosine $\theta(t)$. From (3) and (10), we have

$$\begin{aligned}\theta(t) &= \frac{\left|\Theta^T(t)a_m\right|}{\|\Theta(t)\|}\\ &= \frac{|\phi_m(t)|}{\sqrt{\phi_1^2(t)+\ldots+\phi_m^2(t)}}\\ &= \left[\frac{\phi_1^2(t)}{\phi_m^2(t)}+\ldots+\frac{\phi_{m-1}^2(t)}{\phi_m^2(t)}+1\right]^{-1/2}\\ &\leq \left[\prod_{i=0}^{t-1} q_1^2(i)\cdot\sum_{j=1}^{m-1}\frac{\phi_j^2(0)}{\phi_m^2(0)}+1\right]^{-1/2}\end{aligned} \tag{11}$$

for all $t \geq 0$.

The speed at which direction cosine θ(t) converges to 1 depends on $\prod_{i=0}^{t-1} q_1^2(i)$. From (9) and (11), we find that a small $q_1(i)(i=0,1,\ldots,t-1)$ can accelerate the convergence of $\theta(t)$. In different MCA learning algorithms, $q_1(i)$ has different forms. We study the convergence speeds of different MCA learning algorithms by analyzing the value of $q_1(t)$ and illustrate the relations between convergence speed and initial weight vector.

A. CONVERGENCE SPEED OF THE OJAN MCA ALGORITHM

Xu *et al.* [6] proposed a normalized Oja's MCA learning algorithm called OJAn, which has the following form.

$$\Theta(t+1)=\Theta(t)-\mu\left[R\Theta(t)-\frac{\Theta^T(t)R\Theta(t)}{\Theta^T(t)\Theta(t)}\right]. \tag{12}$$

In OJAn algorithm,

$$\begin{aligned}q_1(t) &= \frac{1-\mu\sigma_1+\mu\cdot\frac{\Theta^T(t)R\Theta(t)}{\Theta^T(t)\Theta(t)}}{1-\mu\sigma_m+\mu\cdot\frac{\Theta^T(t)R\Theta(t)}{\Theta^T(t)\Theta(t)}}\\ &= 1-\frac{\sigma_1-\sigma_m}{\frac{1}{\mu}+\frac{\Theta^T(t)R\Theta(t)}{\Theta^T(t)\Theta(t)}-\sigma_m}.\end{aligned}$$

The value of $q_1(t)$ depends on $\frac{\Theta^T(t)R\Theta(t)}{\Theta^T(t)\Theta(t)}$, which is independent of the norm of $\Theta(t)$ and depends only on the direction of $\Theta(t)$. Considering that the direction of initial weight vector $\Theta(0)$ is random, the selection of $\Theta(0)$ does not significantly affect the convergence speed of the OJAn MCA neural network algorithm.

Next, we use the OJAn MCA algorithm (12) to estimate the minor component of a symmetric matrix as follows:

$$R_1=\begin{bmatrix}0.4182 & 0.2625 & 0.1948 & 0.0106\\ 0.2625 & 0.4731 & 0.1483 & 0.0661\\ 0.1948 & 0.1483 & 0.3856 & 0.3885\\ 0.0106 & 0.0661 & 0.3885 & 0.5331\end{bmatrix}. \tag{13}$$

In the simulation, learning step$\mu=0.01$ and four different initial weight vectors $\Theta_j(0)(j=1,2,3,4)$ are selected. These initial weight vectors $\Theta_j(0)(j=1,2,3,4)$ are in the same direction as the vector

$$\Theta^*=\begin{bmatrix}0.8056 & 0.3709 & 0.1486 & 0.4373\end{bmatrix}^T \tag{14}$$

but have different norms, i.e.,

$$\|\Theta_1(0)\|=0.001,\ \|\Theta_2(0)\|=0.1,\ \|\Theta_3(0)\|=1,\ \|\Theta_4(0)\|=100$$

The simulation result shows that whichever initial weight vector is selected, we obtain the same convergence result as follows:

i. After 100 iterations, direction cosine $\theta(t)$ equals 0.7151;
ii. After 500 iterations, direction cosine $\theta(t)$ equals 0.9791; and
iii. After 1600 iterations, direction cosine $\theta(t)$ equals 0.9997.

B. CONVERGENCE SPEED OF FENG'S MCA ALGORITHM

Feng *et al.* [25] developed an MCA neural network learning algorithm to solve the problem of least squares. Feng's MCA algorithm has the following form:

$$\Theta(t+1)=\Theta(t)-\mu\left[\Theta^T(t)\Theta(t)R\Theta(t)-\Theta(t)\right]. \tag{15}$$

On the one hand, Feng's MCA algorithm indicates that

$$
\begin{aligned}
q_1(t) &= \frac{1-\mu\sigma_1\Theta^T(t)\Theta(t)+\mu}{1-\mu\sigma_m\Theta^T(t)\Theta(t)+\mu} \\
&= 1-\frac{\mu\Theta^T(t)\Theta(t)(\sigma_1-\sigma_m)}{1-\mu\sigma_m\Theta^T(t)\Theta(t)+\mu} \\
&= 1-\frac{\sigma_1-\sigma_m}{\frac{1}{\mu\Theta^T(t)\Theta(t)}+\frac{1}{\Theta^T(t)\Theta(t)}-\sigma_m}.
\end{aligned}
$$

Evidently, $\Theta(t)$ with a small norm can make $q_1(t)$ large and cause the direction cosine $\theta(t)$ in (11) to converge slowly to 1. On the other hand, from (5), (7) and (15) the algorithm holds that

$$
\phi_j(t+1)=\left[1+\mu-\mu\sigma_j\left\|\Theta(t)\right\|^2\right]\cdot\phi_j(t), \quad (j=1,2,\ldots,m), \tag{16}
$$

and

$$
1-\mu\sigma_j\left\|\Theta(t)\right\|^2+\mu>0, \quad (j=1,2,\ldots,m), \tag{17}
$$

for all $t \geq 0$. From (16) and (17), we have

$$
\begin{aligned}
\left\|\Theta(t+1)\right\|^2 &= \sum_{j=1}^{m}\phi_j^2 \\
&= \sum_{j=1}^{m}\left[1+\mu-\mu\sigma_j\left\|\Theta(t)\right\|^2\right]^2\cdot\phi_j^2(t) \\
&\leq \left[1+\mu-\mu\sigma_m\left\|\Theta(t)\right\|^2\right]^2\cdot\left\|\Theta(t)\right\|^2 \\
&\leq (1+\mu)^2\cdot\left\|\Theta(t)\right\|^2,
\end{aligned} \tag{18}
$$

for all $t \geq 0$.

Given that learning step μ is a small positive constant to guarantee convergence, according to (18), if $\Theta(0)$ has a small norm, the increments in $||\Theta(t)||$ will be very limited after a small number of iterations. Therefore, selecting an initial weight vector with a large norm will increase the convergence speed of Feng's MCA neural network algorithm.

Next, Feng's MCA neural networks algorithm (15) is used to estimate the minor component of R_1 in (13). Given learning step $\mu = 0.01$ and four different initial weight vectors $\Theta_j(0)(j=1,2,3,4)$ are selected; all $\Theta_j(0)(j=1,2,3,4)$ are in the same direction as Θ^* and $||\Theta_1(0)||=0.01$, $||\Theta_2(0)||=0.1$, $||\Theta_3(0)||=1$, $||\Theta_4(0)||=10$. Fig. 2(a) shows the iterations of direction cosine $\theta(t)$ with different initial weight vectors. We then consider a general case in which autocorrelation matrix R are randomly generated, and their dimensions are 50×50. Fig. 2(b) illustrates the simulation results averaged over 100 independent runs. The results show that the initial weight vector with a large norm results in rapid convergence.

C. CONVERGENCE SPEED OF DOUGLAS' MCA ALGORITHM

Douglas *et al.* [11] presented self-stabilizing MCA learning algorithm as follows:

$$
\begin{aligned}
\Theta(t+1)=\Theta(t)-\mu\Big[&\Theta^T(t)\Theta(t)\Theta^T(t)\Theta(t)R\Theta(t) \\
&-\Theta^T(t)R\Theta(t)\Theta(t)\Big].
\end{aligned} \tag{19}
$$

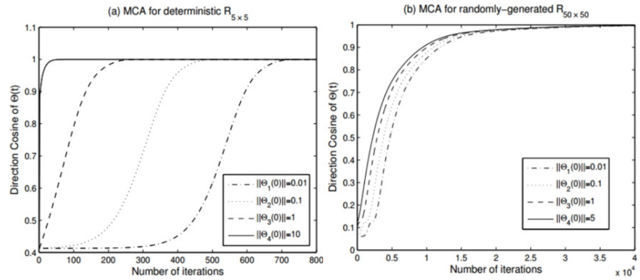


FIGURE 2. Convergence of direction cosine $\Theta(t)$ in Feng's MCA algorithm. (a) Convergence results with 5×5 deterministic matrix R; (b) Convergence results with 50×50 randomly-generated matrix R.

In Douglas' MCA algorithm (19), $q_1(t)$ is in the form:

$$
\begin{aligned}
q_1(t) &= \frac{1-\mu\sigma_1\left\|\Theta(t)\right\|^4+\mu\Theta^T(t)R\Theta(t)}{1-\mu\sigma_m\left\|\Theta(t)\right\|^4+\mu\Theta^T(t)R\Theta(t)} \\
&= 1-\frac{\mu\sigma_1\left\|\Theta(t)\right\|^4-\mu\sigma_m\left\|\Theta(t)\right\|^4}{1-\mu\sigma_m\left\|\Theta(t)\right\|^4+\mu\Theta^T(t)R\Theta(t)} \\
&= 1-\frac{\sigma_1-\sigma_m}{\frac{1}{\mu\left\|\Theta(t)\right\|^4}+\frac{\Theta^T(t)R\Theta(t)}{\Theta^T(t)\Theta(t)}\cdot\frac{1}{\left\|\Theta(t)\right\|^2}-\sigma_m}
\end{aligned}
$$

Considering the properties of the Rayleigh quotient, we find that

$$
\sigma_m \leq \frac{\Theta^T(t)R\Theta(t)}{\Theta^T(t)\Theta(t)} \leq \sigma_1 \tag{20}
$$

for all $t \geq 0$.

Obviously,

$$
\begin{aligned}
1-\frac{\sigma_1-\sigma_\mathrm{m}}{\frac{1}{\mu\left\|\Theta(t)\right\|^4}+\frac{\sigma_\mathrm{m}}{\left\|\Theta(t)\right\|^2}-\sigma_\mathrm{m}} &\leq q_1(t) \\
&\leq 1-\frac{\sigma_1-\sigma_\mathrm{m}}{\frac{1}{\mu\left\|\Theta(t)\right\|^4}+\frac{\sigma_1}{\left\|\Theta(t)\right\|^2}-\sigma_\mathrm{m}},
\end{aligned} \tag{21}
$$

for all $t \geq 0$.

From (11) and (21), we find that a large $||\Theta(t)||$ results in a small $q_1(t)$ and makes direction cosine $\theta(t)$ converge to 1 quickly.

Meanwhile, from (5), (7) and (19) of Douglas algorithm, we have

$$
\phi_j(t+1)=\left[1=\mu\sigma_j\left\|\Theta(t)\right\|^4+\mu\Theta^T(t)R\Theta(t)\right]\cdot\phi_j(t), \quad (j=1,2,\ldots,m), \tag{22}
$$

And

$$
1-\mu\sigma_j\left\|\Theta(t)\right\|^4+\mu\Theta^T(t)R\Theta(t)>0, \quad (j=1,2,\ldots,m), \tag{23}
$$

for all $t \geq 0$.

Therefore, (6), (20), (22) and (23) indicate that

$$
\begin{aligned}
\left\|\Theta(t+1)\right\|^2 &= \sum_{j=1}^{m}\phi_j^2(t+1) \\
&= \sum_{j=1}^{m}\phi_j^2(t)\cdot\left[1-\mu\sigma_j\left\|\Theta(t)\right\|^4+\mu\Theta^T(t)R\Theta(t)\right]^2
\end{aligned}
$$

$$\leq \sum_{j=1}^{m} \phi_j^2(t) \cdot \left[1 + \mu\Theta^T(t)R\Theta(t)\right]^2$$
$$= \left[1 + \mu\Theta^{\mathrm{T}}(t)R\Theta(t)\right]^2 \cdot \|\Theta(t)\|^2$$
$$\leq \left[1 + \mu\sigma_1 \|\Theta(t)\|^2\right]^2 \cdot \|\Theta(t)\|^2$$
$$\leq \left[1 + \|\Theta(\mathrm{t})\|^2\right]^2 \cdot \|\Theta(t)\|^2,$$

i. e.,

$$\|\Theta(t+1)\| \leq \left[1 + \|\Theta(t)\|^2\right] \cdot \|\Theta(t)\|, \tag{24}$$

for all $t \geq 0$.

Clearly, (24) means that if $||\Theta(0)||$ is small, the norm of the weight vector $\Theta(t)$ will increase slowly. Selecting an initial weight vector with a large norm will thus accelerate the convergence of Douglas MCA algorithm.

Next, we use Douglas MCA algorithm (19) to extract the minor component of R_1 with learning step$\mu = 0.001$ and four different initial weight vectors $\Theta_j(0)(j = 1, 2, 3, 4)$, where $\Theta_1(0) = 0.01\Theta^*$, $\Theta_2(0) = 0.5\Theta^*$, $\Theta_3(0) = \Theta^*$, $\Theta_4(0) = 5\Theta^*$. The simulation result illustrated in Fig. 3(a) shows how initial weight vectors with different norms affect the convergence speeds of Douglas MCA algorithm. Fig. 3(b) provides the simulation results averaged over 100 independent runs with the 50×50 randomly generated autocorrelation matrices. The results show that an initial weight vector with a large norm results in rapid convergence.

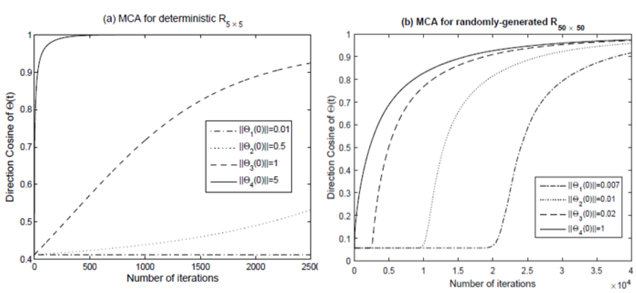


FIGURE 3. Convergence of direction cosine Θ(*t*) in Douglas' MCA algorithm. (a) Convergence results with 5 × 5 deterministic matrix R; (b) Convergence results with 50 × 50 randomly-generated matrix R.

D. CONVERGENCE SPEED OF THE AMEX MCA ALGORITHM

Ouyang *et al.* [13] proposed an adaptive MCA algorithm called AMEX with the following form.

$$\Theta(t+1) = \Theta(t) - \mu\left[R\Theta(t) - \frac{\Theta(t)}{\Theta^T(t)\Theta(t)}\right]. \tag{25}$$

From (5), we have

$$\phi_j(t+1) = \phi_j(t) \cdot \left[1 - \mu\sigma_j + \frac{\mu}{\Theta^T(t)\Theta(t)}\right], \quad (\mathrm{j} = 1, 2, \ldots, \mathrm{m}), \tag{26}$$

for all $t \geq 0$.

In the AMEX MCA algorithm, $q_1(t)$ is regarded as:

$$q_1(t) = \frac{1 - \mu\sigma_1 + \frac{\mu}{\Theta^T(t)\Theta(t)}}{1 - \mu\sigma_m + \frac{\mu}{\Theta^T(t)\Theta(t)}} = 1 - \frac{\sigma_1 - \sigma_m}{\frac{1}{\mu} - \sigma_m + \frac{1}{\Theta^T(t)\Theta(t)}}.$$

From (11), we find that a small $||\Theta(t)||$ can produce a large $q_1(t)$ and result in slow convergence of direction cosine $\theta(t)$. Meanwhile, (6) and (26) indicate that

$$\|\Theta(t+1)\|^2 = \sum_{\mathrm{j}=1}^{\mathrm{m}} \phi_j^2(t) \cdot \left[1 - \mu\sigma_j + \frac{\mu}{\Theta^T(t)\Theta(t)}\right]^2$$
$$\geq \sum_{j=1}^{m} \phi_j^2(t) \cdot \left[\frac{\mu}{\Theta^T(t)\Theta(t)}\right]^2$$
$$\geq \frac{\mu^2}{\|\Theta(\mathrm{t})\|^2} \tag{27}$$

for all $t \geq 0$.

Evidently, (27) means that if $||\Theta(0)||$ is small, $||\Theta(t)||$ will increase rapidly. Therefore, $||\Theta(0)||$ does not affect the convergence speed of the AMEX algorithm significantly.

Next, we use the AMEX algorithm (25) to extract the minor component of R_1.In this simulation, learning step $\mu = 0.1$ and four different initial weight vectors $w_j(0)(j = 1, 2, 3, 4)$ are selected, where $w_1(0) = 0.001\Theta^*$, $w_2(0) = 0.1\Theta^*$, $w_3(0) = \Theta^*$, and $w_4(0) = 100\Theta^*$. Θ^* is defined in (14). Fig. 4(a) illustrates the convergence of direction cosine $\theta(t)$ of the weight vector $\Theta(t)$ in the AMEX algorithm starting from the different initial weight vectors. Fig. 4(b) shows the simulation results averaged over 100 independent runs with the 50 × 50 randomly generated autocorrelation matrices. The simulation results in Fig. 4 indicates that the convergence speed of the AMEX algorithm is basically independent of $||\Theta(0)||$.

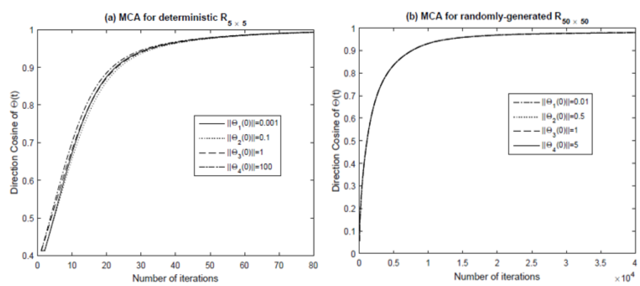


FIGURE 4. Convergence of direction cosine Θ(*t*) in AMEX algorithm. (a) Convergence results with 5 × 5 deterministic matrix R; (b) Convergence results with 50 × 50 randomly-generated matrix R.

The DDT based convergence analysis indicates that MCA learning algorithms based on neural networks can be divided into two classes according to the relation between convergence speed and initial weight vector's norm $||\Theta(0)||$. In the first class of MCA algorithms (e.g., OJAn and AMEX algorithms), $||\Theta(0)||$ does not significantly affect the convergence speed. In the second class of MCA algorithms (e.g., Douglas' algorithm and Feng's algorithm), the convergence speed depends on $||\Theta(0)||$ and a large $||\Theta(0)||$ can accelerate the convergence of learning algorithms. However, MCA learning algorithms belonging to the second class are usually locally convergent, and a large $||\Theta(0)||$ may result in the divergence problem. Finding an upper bound of $||\Theta(0)||$ that can guarantee convergence is therefore interesting. Several related analyses are provided in [35], [39].

Based on the analysis results, we present the following guidelines for selecting initial weight vectors to improve convergence performance of algorithms.

(1) For OJAn and AMEX algorithms, an initial weight vector with a small norm does not negatively affect convergence speed and helps guarantee the convergence of algorithms.

(2) For Douglas algorithm and Feng's algorithm, the selection of an initial weight vector that possesses a large norm and satisfies convergence conditions will effectively improve algorithm performance.

III. COMPARISON OF CONVERGENCE SPEEDS

We compare the convergence speeds of different MCA learning algorithms. From (2) and (5), we easily find that when weight vector $\Theta(t)$ approaches minor component a_m, i.e., the learning is close to convergence, then

$$1 - \mu\sigma_m f(\Theta(t)) + \mu g(\Theta(t)) \approx 1. \tag{28}$$

With (8) and (28) we get:

$$q_1(t) \approx 1 - \mu f(\Theta(t)) \cdot (\sigma_1 - \sigma_m), \tag{29}$$

A. OJAn, AMEX, and OJAm MCA ALGORITHMS

Reference [26] developed a modified MCA learning algorithm called OJAm, which has the following form.

$$\Theta(t+1) = \Theta(t) - \mu\left[R\Theta(t) - \frac{\Theta^T(t)R\Theta(t)}{\Theta^T(t)\Theta(t)\Theta^T(t)\Theta(t)} \cdot \Theta(t)\right]. \tag{30}$$

In the OJAn (12), AMEX (25) and OJAm (30) algorithms,

$$f(\Theta(t)) = 1, \quad \text{for all } t \geq 0.$$

(29) indicates that in OJAn, AMEX, and OJAm algorithms,

$$q_1(t) \approx 1 - \mu \cdot (\sigma_1 - \sigma_m), \tag{31}$$

for all $t \geq t_0$, where t_0 is a sufficiently large constant.

Thus, (11) and (31) show that OJAn, AMEX, and OJAm algorithms possess approximately the same convergence speed.

A simulation is conducted to illustrate the convergence speeds of OJAn, AMEX, and OJAm algorithms. In this simulation, the three MCA learning algorithms are used to extract the minor component of correlated matrix R1. The same learning step$\mu = 0.5$ and the same initial weight vector

$$\Theta(0) = [1.7518 \quad 0.7390 \quad 0.3884 \quad 0.3626]^T$$

are selected. Fig. 5 shows the iterations of the direction cosine $\theta(t)$ in OJAn, AMEX, and OJAm algorithms. These MCA learning algorithms have approximately equal convergence speeds.

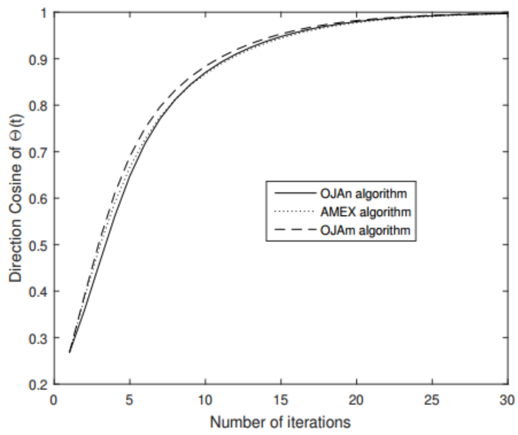


FIGURE 5. Comparison of the convergence speeds of OJAn, AMEX, and OJAm Algorithms.

B. DOUGLAS' MCA ALGORITHM

In Douglas' MCA algorithm (19),

$$f(\Theta(t)) = \Theta^T(t)\Theta(t)\Theta^T(t)\Theta(t)$$

for all $t \geq 0$. From (29), we have

$$q_1(t) \approx 1 - \mu \|\Theta(t)\|^4 \cdot (\sigma_1 - \sigma_m), \tag{32}$$

for all $t \geq 0$, where t_0 is a sufficiently large constant.

From (11), (31) and (32), we can draw the following conclusions.

When $||\Theta(t)|| > 1$, Douglas' MCA algorithm has a smaller $q_1(t)$ and higher convergence speed compared with OJAn, OJAm, and AMEX MCA algorithms. When $||\Theta(t)|| < 1$, Douglas' MCA algorithm has a larger $q_1(t)$ and lower convergence speed compared with OJAn, OJAm, and AMEX MCA algorithms.

Fig. 6 and Fig. 7 show comparisons of the convergence speeds of Douglas algorithm, OJAn, OJAm, and AMEX when $||\Theta(0)||$ is equal to 0.5 and 2, respectively. The experimental results shown in the two figures are identical to the analysis result.

C. FENG'S MCA ALGORITHM

References [25] and [39] prove that in Feng's MCA algorithm,

$$\lim_{t\to\infty} \|\Theta(t)\|^2 = \frac{1}{\sigma_m},$$

where σ_m stands for the minimal eigenvalue of the correlated matrix of inputs. Thus, from and (29), we have

$$q_1(t) \approx 1 - \mu \|\Theta(t)\|^2 \cdot (\sigma_1 - \sigma_m) \approx 1 - \frac{\mu}{\sigma_m} \cdot (\sigma_1 - \sigma_m), \tag{33}$$

for all $t \geq t_0$, where t_0 is a sufficiently large constant.

In practical applications, the minimum eigenvalue σ_m is usually a very small positive due to the existence of noise.

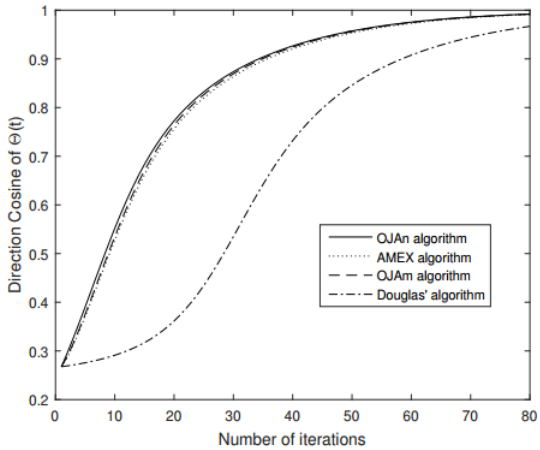


FIGURE 6. Comparison of the convergence speeds of Douglas' algorithm, OJAn, AMEX and OJAm when $\|\Theta(0)\| = 0.5$.

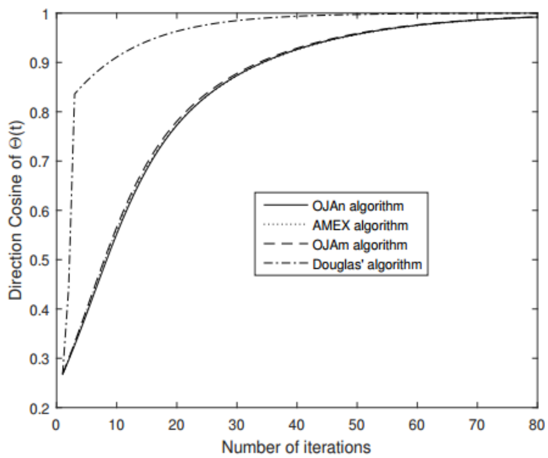


FIGURE 7. Comparison of the convergence speeds of Douglas' algorithm, OJAn, AMEX and OJAm when $\|\Theta(0)\| = 2$.

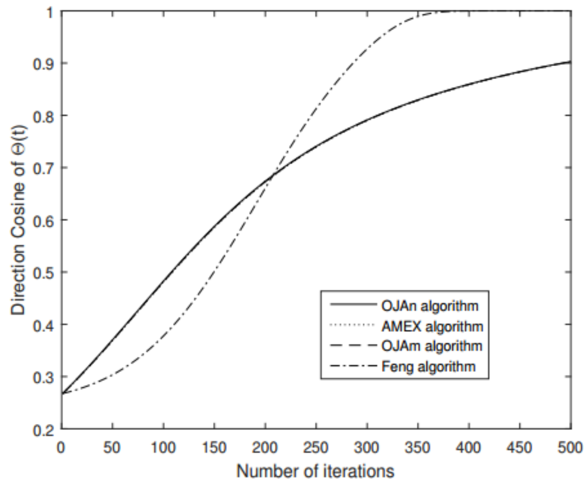


FIGURE 8. Comparison of the convergence speeds of Feng's algorithm, OJAn, AMEX and OJAm when $\|\Theta(0)\| = 0.5$.

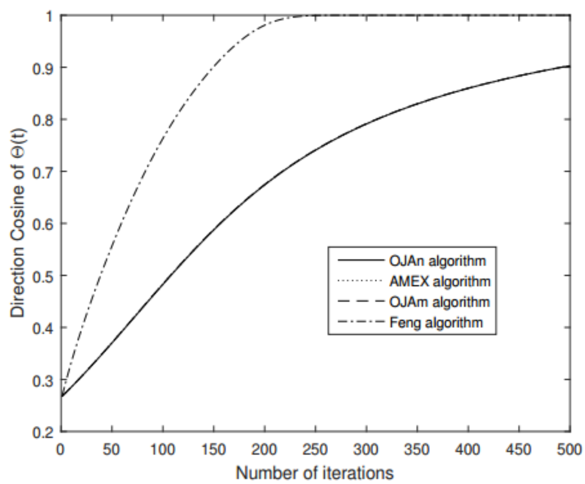


FIGURE 9. Comparison of the convergence speeds of Feng's algorithm, OJAn, AMEX and OJAm when $\|\Theta(0)\| = 2$.

Thus, (11), (31) and (33) indicate that when the learning process is close to convergence, Feng's MCA neural network algorithm has a smaller $q_1(t)$ and higher convergence speed compared with OJAn, AMEX, and OJAm algorithms. In the initial phase of learning, the convergence speed of Feng's MCA neural network algorithm depends on the norm of the initial weight vector $\Theta(0)$. Fig. 8 and Fig. 9 show comparisons of the convergence speeds of Feng's algorithm, OJAn, OJAm, and AMEX when $||\Theta(0)|| = 0.5$ and $||\Theta(0)|| = 2$, respectively. The simulation result is identical to the analysis result.

IV. IMPROVED MCA ALGORITHM WITH FAST CONVERGENCE

In previous section, convergence performance of MCA algorithm, especially the convergence speed, was studied and factors affecting convergence performance were revealed.

Based on above analysis, we proposes a new MCA algorithm with fast convergence. Then we use determining discrete time method to prove theoretically its convergence and also provide simulation results.

The proposed MCA algorithm is as follows:

$$\theta(\mathrm{t}+1) = \theta(t) - \mu[z(t)y(t) - (0.5 - \|\theta(t) \quad \|) \cdot \theta(t)] \quad (34)$$

where $\theta(t)$ is the network connection weight vector at interval t, μ is the learning factor, $z(t)$ is the output value, and $y(t)$ is the signal vector.

We will prove that this learning algorithm iteratively updates the weight vector $\theta(t)$ which will finally converge to its minor component a_m with the constraint: the minimum eigenvalue σ_m of the autocorrelation matrix

$$R = E[y(t)y^T(t)] \quad \text{of } \mathrm{y}(t)(t = 1, 2, \ldots)$$

is less than 0.5.

In practical engineering applications with multi-dimensional data, minimum eigenvalues of the autocorrelation matrix of the signal vector are usually close to zero. Therefore, above constraint we have proposed

$$\sigma_m < 0.5 \tag{35}$$

is easy to be satisfied.

Next, we need to prove the convergence of the algorithm (34), that is, to prove that $\theta(t)$ will converge to a_m. We still use deterministic discrete method that takes the conditional expectation E $\{\theta(t+1)/\theta(0), y(i), i < t\}$ on both sides of (34), which can get discrete time (DDT) system as follows:

$$\theta(t+1) = \theta(t) - \mu[R\theta(t) - (0.5 - \|\theta(t)\|) \cdot \theta(t)]. \tag{36}$$

By substituting (3) into (36), we can get:

$$\phi_j(t+1) = [1 - \mu\sigma_j + \mu(0.5 - \|\theta(t)\|)] \cdot \phi_j(t), \quad (j = 1, 2, \ldots, m). \tag{37}$$

In order to complete the convergence analysis of the discrete time system (36), we first introduce necessary lemmas and theorems as follows.

Lemma 1: If $\|\theta(t)\| \leq 1 - 2\sigma_m$ and

$$\mu \leq \min\left\{2, \frac{1}{\sigma_1 + |2\sigma_m - 0.5|}\right\}$$

Then

$$1 - \mu\sigma_1 + \mu(0.5 - \|\theta(t)\|) \geq 0$$

and

$$1 - \mu\sigma_j + \mu(0.5 - \|\theta(t)\|) > 0, \quad (2 \leq j \leq m).$$

Proof: Since

$$\mu \leq \min\left\{2, \frac{1}{\sigma_1 + |2\sigma_m - 0.5|}\right\}$$

So, we have

$$\mu\sigma_1 + \mu\,|2\sigma_m - 0.5| \leq 1 \tag{38}$$

Next, we continue our proving by dividing into two situations.

Situation 1: $\|\theta(t)\| \leq 0.5 - \sigma_m$

According to (38), it can be found that for all $\|\theta(t)\| \leq 0.5 - \sigma_m$, there is

$$\begin{aligned} 1 - \mu\sigma_1 + \mu(0.5 - \|\theta(t)\|) &\geq 1 - \mu\sigma_1 + \mu\sigma_m \\ &\geq 1 - \mu\sigma_1 \\ &\geq 0 \end{aligned}$$

Situation 2: $0.5 - \sigma_m < \|\theta(t)\| \leq 1 - 2\sigma_m$

Because $\|\theta(t)\| \leq 1 - 2\sigma_m$, obviously,

$$0.5 - \|\theta(t)\| \geq 2\sigma_m - 0.5 \tag{39}$$

According to (38) and (39), following conclusion can be drawn:

$$\begin{aligned} 1 - \mu\sigma_1 + \mu(0.5 - \|\theta(t)\|) &\geq 1 - \mu\sigma_1 + \mu(2\sigma_m - 0.5) \\ &\geq 1 - \mu\sigma_1 + \mu\,|2\sigma_m - 0.5| \\ &\geq 0 \end{aligned}$$

Combining above two cases, we can see that for all $\|\theta(t)\| \leq 1 - 2\sigma_m$, there is

$$1 - \mu\sigma_1 + \mu(0.5 - \|\theta(t)\|) \geq 0$$

Simultaneously,

$$\begin{aligned} 1 - \mu\sigma_j + \mu(0.5 - \|\theta(t)\|) > 1 - \mu\sigma_1 \\ + \mu(0.5 - \|\theta(t)\|) > 0, \quad (2 \leq j \leq m) \end{aligned}$$

Therefore, the proof is completed.

Next, we will introduce an important theorem that will lead to boundedness of the learning algorithm and ensure that the algorithm does not diverge.

Theorem 1: If $\|\theta(0)\| \leq 1 - 2\sigma_m$ and

$$\mu \leq \min\left\{2, \frac{1}{\sigma_1 + |2\sigma_m - 0.5|}\right\}$$

Then for all t ≥ 0,

$$\|\theta(t)\| \leq 1 - 2\sigma_m$$

Proof: Let $t_0 \geq 0$, assuming:

$$\|\theta(t_0)\| \leq 1 - 2\sigma_m \tag{40}$$

According to Lemma 1,

$$1 - \mu\sigma_j + \mu(0.5 - \|\theta(t_0)\|) \geq 0, \quad (j = 1, 2, \ldots, m) \tag{41}$$

Situation 1: $\|\theta(t_0)\| \leq 0.5 - \sigma_m$

Obviously,

$$0.5 \geq 0.5 - \|\theta(t_0)\| \geq \sigma_m \tag{42}$$

Since

$$\mu \leq \min\left\{2, \frac{1}{\sigma_1 + |2\sigma_m - 0.5|}\right\}$$

from (37), (41), (42), we can get

$$\begin{aligned} \|\theta\,(t_0+1)\|^2 &= \sum_{j=1}^{m} \phi_j^2\,(t_0+1) \\ &= \sum_{j=1}^{m} \left[1 - \mu\sigma_j + \mu\,(0.5 - \|\theta(t_0)\|)\right]^2 \cdot \phi_j^2\,(t_0) \\ &\leq \sum_{j=1}^{m} [1 + \mu\,(0.5 - \|\theta(t_0)\|)]^2 \cdot \phi_j^2\,(t_0) \\ &= [1 + \mu\,(0.5 - \|\theta(t_0)\|)]^2 \cdot \|\theta(t_0)\|^2 \\ &\leq (1 + 0.5\mu)^2 \cdot \|\theta(t_0)\|^2 \\ &\leq (1 + 0.5\mu)^2 \cdot (0.5 - \sigma_m)^2 \\ &\leq 4\,(0.5 - \sigma_m)^2 \end{aligned}$$

That is, $\|\theta\,(t_0+1)\| \leq 1 - 2\sigma_m$.

Situation 2: $0.5 - \sigma_m < \|\theta(t_0)\| \leq 1 - 2\sigma_m$

Since $0.5 - \sigma_m < \|\theta(t_0)\|$, so

$$1 - \mu\sigma_m + \mu\,(0.5 - \|\theta(t_0)\|) \leq 1 - \mu\sigma_m + \mu\sigma_m = 1 \tag{43}$$

According to (37), (41), (42) and (43), we can get

$$\begin{aligned}\|\theta(t_0+1)\|^2 &= \sum_{j=1}^{m}\phi_j^2(t_0+1)\\ &= \sum_{j=1}^{m}\left[1-\mu\sigma_j+\mu(0.5-\|\theta(t_0)\|)\right]^2\cdot\phi_j^2(t_0)\\ &\le \sum_{j=1}^{m}[1-\mu\sigma_m+\mu(0.5-\|\theta(t_0)\|)]^2\cdot\phi_j^2(t_0)\\ &\le \|\theta(t_0)\|^2\\ &\le (1-2\sigma_m)^2\end{aligned}$$

That is, $\|\theta(t_0+1)\| \le 1-2\sigma_m$

Then it can be known that if $\|\theta(0)\| \le 1-2\sigma_m$, then for all t ≥ 0, have $\|\theta(t)\| \le 1-2\sigma_m$.

Theorem 1 shows the boundedness of the weight vector $\theta(t)$ in the system (36). By selecting the initial weight vector $\theta(0)$ with a norm which is less than $1-2\sigma_m$, it is guaranteed that the weight vector $\theta(t)$ is confined to a finite set throughout the learning process, which fundamentally prevents the divergence of the algorithm.

Next, we will carry out further convergence analysis on (36), and prove that the weight vector $\theta(t)$ can converge to the eigenvector corresponding to the minimum eigenvalue σ_m of the autocorrelation matrix.

Theorem 2; If $\|\theta(0)\| \le 1-2\sigma_m, \theta^T(0)a_{\mathrm{m}} \ne 0$, and

$$\mu \le \min\left\{2, \frac{1}{\sigma_1+|2\sigma_m-0.5|}\right\}$$

Then,

$$\lim_{t\to\infty}\frac{\theta(t)}{\|\theta(t)\|} = \pm a_{\mathrm{m}}.$$

Proof: Since $\|\theta(0)\| \le 1-2\sigma_m$, according to Theorem 1, it can be seen that for all t ≥ 0, $\|\theta(t)\| \le 1-2\sigma_m$. By Lemma 1, we know,

$$1-\mu\sigma_j+\mu(0.5-\|\theta(t)\|) \ge 0, \quad (j=1,2,\ldots,m-1) \tag{44}$$

and

$$1-\mu\sigma_m+\mu(0.5-\|\theta(t)\|) > 0 \tag{45}$$

According to (35) and (45), it can be known that for all t ≥ 0, there are

$$\begin{aligned}&\frac{1-\mu\sigma_j+\mu(0.5-\|\theta(t)\|)}{1-\mu\sigma_m+\mu(0.5-\|\theta(t)\|)}\\ &\quad= 1-\frac{\mu\sigma_j-\mu\sigma_m}{1-\mu\sigma_m+\mu(0.5-\|\theta(t)\|)}\\ &\quad\le 1-\frac{\mu\sigma_{m-1}-\mu\sigma_m}{1-\mu\sigma_m+0.5\mu}, \quad (j=1,2,\ldots,m-1).\end{aligned}$$

Define:

$$\beta = 1-\frac{\mu(\sigma_{m-1}-\sigma_m)}{1-\mu(\sigma_m-0.5)}$$

From (44) and (45), it is known

$$1 > \beta \ge 0 \tag{46}$$

Since $\theta^T(0)a_{\mathrm{m}} \ne 0, \phi_{\mathrm{m}}(0) \ne 0$. So according to (37) and (45), obviously for all t ≥ 0, there are

$$\begin{aligned}\frac{|\phi_j(t+1)|}{|\phi_m(t+1)|} &= \frac{1-\mu\sigma_j+\mu(0.5-\|\theta(t)\|)}{1-\mu\sigma_m+\mu(0.5-\|\theta(t)\|)}\cdot\left|\frac{\phi_j(t)}{\phi_m(t)}\right|\\ &\le \beta\cdot\left|\frac{\phi_j(t)}{\phi_m(t)}\right|\\ &\le \beta^{t+1}\cdot\left|\frac{\phi_j(0)}{\phi_m(0)}\right|, \quad (j=1,2,\ldots,m-1).\end{aligned}$$

According to (46), obviously

$$\lim_{t\to\infty}\left|\frac{\phi_j(t)}{\phi_m(t)}\right| = 0, \quad (j=1,2,\ldots,m-1).$$

According to Theorem 1, it can be known that $|\phi_m(t)|$ must be bounded. Therefore

$$\lim_{t\to\infty}\phi_j(t) = 0, \quad (j=1,2,\ldots,m-1)$$

From the above formula and (37), wou can get

$$\lim_{t\to\infty}\frac{\theta(t)}{\|\theta(t)\|} = \lim_{t\to\infty}\frac{\phi_m(t)}{|\phi_m(t)|}\cdot a_m = \pm a_m.$$

Theorem 3: If

$$\mu \le \min\left\{2, \frac{1}{\sigma_1+|2\sigma_m-0.5|}, \frac{1}{0.5-\sigma_m}\right\}$$

and $\|\theta(0)\| \le 1-2\sigma_m, \theta^T(0)a_{\mathrm{m}} \ne 0$,then

$$\lim_{t\to\infty}\theta(t) = \pm(0.5-\sigma_m)\cdot a_m.$$

Proof: According to Theorem 2, we know that when t $\to\infty$, the weight vector $\theta(t)$ will converge to the direction of the minor component a_m. Suppose that after the t_0 iteration, $\theta(t)$ has approached the a_m direction:

$$\theta(t) \approx \phi_m(t)\cdot a_m, (t \ge t_0) \tag{47}$$

Then, the system (36) can be simplified to the following form:

$$\theta(t+1) = \theta(t)-\mu\sigma_m\theta(t)+\mu(0.5-\|\theta(t)\|)\theta(t) \tag{48}$$

That is:

$$\theta(t+1) = [1-\mu\sigma_m+\mu(0.5-\|\theta(t)\|)]\cdot\theta(t) \tag{49}$$

Take norm on both sides and according to Lemma 1, we can get:

$$\|\theta(t+1)\| = [1-\mu\sigma_m+\mu(0.5-\|\theta(t)\|)]\cdot\|\theta(t)\| \tag{50}$$

Obviously, the equilibrium point of the system (50) includes 0 and $0.5-\sigma_m$. Since $\theta^T(0)a_m \ne 0, \theta^T(0) \ne 0$.

From Lemma 1 and (37), it is known that for all $t \ge t_0$, $\phi_m(0) \ne 0$ and $\|\theta(t)\| \ne 0$.

Based on the above analysis, according to (50), it can be found that:

$$\frac{\|\theta(t+1)\|}{\|\theta(t)\|} = 1+\mu[0.5-\|\theta(t)\|-\sigma_m]. \tag{51}$$

It is easy to see from the above equation that if $\|\theta(t)\| < 0.5 - \sigma_m$, then $\|\theta(t+1)\| > \|\theta(t)\|$.

Therefore, 0 must not be the stable equilibrium point of the system (50), and $0.5-\sigma_m$ is the only stable equilibrium point. Next, we will consider three cases to complete the proof

Situation 1: $\|\theta(t_0)\| \leq 0.5 - \sigma_m$

Because

$$\mu \leq \frac{1}{0.5 - \sigma_m},$$

Then,

$$1 - \mu\,(0.5 - \sigma_m) \geq 0 \tag{52}$$

Define a function $K\,(\|\theta\,(t)\|)$ as follows:

$$K(\|\theta(t)\|) = [1 - \mu\sigma_m + \mu\,(0.5 - \|\theta(t)\|)] \cdot \|\theta(t)\|$$

Then, according to (52), it can be seen that for all $\|\theta(t)\| \leq 0.5 - \sigma_m$, there are

$$\begin{aligned} K'\,(\|\theta\,(t)\|) &= 1 - \mu\sigma_m + 0.5\mu - 2\mu\,\|\theta(t)\| \\ &\geq 1 - \mu\,\|\theta(t)\| \\ &\geq 1 - \mu(0.5 - \sigma_m) \\ &\geq 0 \end{aligned}$$

The above result means that the function $K(.)$ is monotonically increasing over the interval $[0, 0.5 - \sigma_m]$.

From (50), if $\|\theta(t)\| \leq 0.5 - \sigma_m$, then

$$\|\theta(t+1)\| = K(\|\theta(t)\|) \leq K(0.5 - \sigma_m) = 0.5 - \sigma_m.$$

Therefore, if $\|\theta(t_0)\| \leq 0.5 - \sigma_m$, then for all $t \geq t_0$, have

$$\|\theta(t)\| \leq 0.5 - \sigma_m \tag{53}$$

According to (51) and (53), it is known that $\|\theta(t)\|$ is monotonically increasing for all $t \geq t_0$. From (53), the following results can be obtained:

$$\lim_{t\to\infty} \|\theta(t)\| = 0.5 - \sigma_m$$

Situation 2: For all $t \geq t_0$, $\|\theta(t)\| > 0.5 - \sigma_m$

According to (51), it can be found that for all $t \geq t_0$, there are

$$\|\theta(t+1)\| < \|\theta(t)\|\,.$$

At the same time, since $\|\theta(t)\| > 0.5 - \sigma_m$ for all $t \geq t_0$, when $k \to \infty$, $\|\theta(t)\|$ will converge to $0.5 - \sigma_m$.

Situation 3: $\|\theta(t_0)\| > 0.5 - \sigma_m$ and there exists a constant Z($Z > t_0$) that $\|\theta(Z)\| \leq 0.5 - \sigma_m$

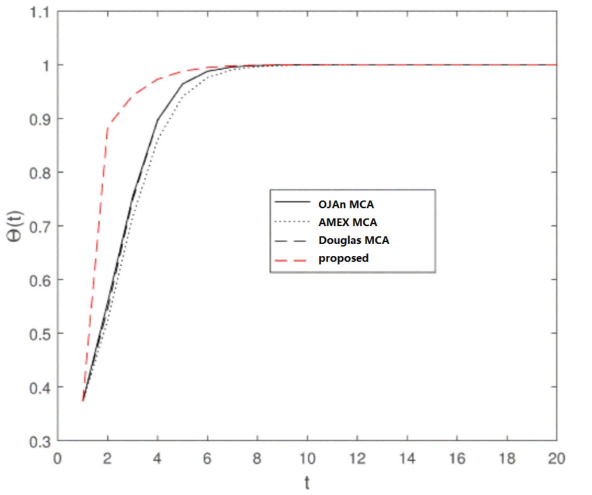


FIGURE 10. Comparison of convergence speeds of MCA neural network learning algorithms.

Since $\|\theta(Z)\| \leq 0.5 - \sigma_m$, similar as Situation 1, we can prove that $\|\theta(t)\|$ will converge to $0.5 - \sigma_m$.

Finally, following conclusions can be drawn:

$$\lim_{t\to\infty} \|\theta(t)\| = 0.5 - \sigma_m$$

Combined with Theorem 2, we know that:

$$\lim_{t\to\infty} \theta(t) = \pm(0.5 - \sigma_m)a_m$$

The proof is completed.

In order to verify the effectiveness of our proposed MCA algorithm (36), we compare the performance of the algorithm with OJAn MCA algorithm, AMEX MCA algorithm and Douglas MCA algorithm. We use the direction cosine between the neuron weight vector $\theta(t)$ and the subcomponent a_m as the performance index $\Theta\,(t)$ as follows:

$$\Theta\,(t) = \frac{\left|\theta^T\,(t)\,a_m\right|}{\|\theta\,(t)\|} \tag{54}$$

It is not difficult to find that the velocity at which $\Theta\,(t)$ approaches 1 can describe the velocity at which the neuron weight vector $\theta(t)$ converges to minor component a_m.

In the simulation experiment, we need to extract minor component of a randomly generated symmetric matrix (55), as shown at the bottom of this page.

The simulation performed a total of 100 Monte Carlo runs. The initial weight vector of each neuron was randomly taken

$$R = \begin{bmatrix} 0.3146 & 0.2496 & 0.2198 & 0.2533 & 0.2399 & 0.1714 & 0.1798 \\ 0.2496 & 0.2919 & 0.2094 & 0.2477 & 0.2680 & 0.2075 & 0.1559 \\ 0.2198 & 0.2094 & 0.3031 & 0.2609 & 0.2718 & 0.1792 & 0.1874 \\ 0.2533 & 0.2477 & 0.2609 & 0.3980 & 0.2827 & 0.1730 & 0.1601 \\ 0.2399 & 0.2680 & 0.2718 & 0.2827 & 0.3792 & 0.2225 & 0.1449 \\ 0.1714 & 0.2075 & 0.1792 & 0.1730 & 0.2225 & 0.2108 & 0.1329 \\ 0.1798 & 0.1559 & 0.1874 & 0.1601 & 0.1449 & 0.1329 & 0.2032 \end{bmatrix} \tag{55}$$

as: $\theta(0) = \text{rand}\ (7,1)$. The average convergence speed $\Theta\ (t)$, in 100 random runs, is shown in Figure 7. The learning factor μ of the algorithm is taken as $\mu = 0.05$. Fig.10 shows the variation of the performance index $\Theta\ (t)$ of the four different MCA algorithms. It can be seen the proposed learning algorithm has faster convergence than OJAn MCA algorithm, AMEX MCA algorithm and Douglas MCA algorithms.

V. CONCLUSIONS

Characteristic analysis of 5G multi-dimensional signals is challenging and MCA is a powerful approach to leverage. Convergence performance of MCA algorithms based on neural networks is analyzed with a deterministic discrete time method in this paper and reveals how initial weight vectors affect the convergence speeds of different algorithms. The selection of initial weight vectors does not exert significant effects on the performance of several MCA algorithms, but selecting initial weight vectors with large norms can accelerate convergence in other MCA algorithms. Comparison of the convergence speeds of several typical MCA algorithms are conducted under different initial conditions. The results provide useful guidelines to improve the performance of MCA algorithms by selecting initial weight vectors. Based on this finding, a new MCA algorithm is proposed and achieve fast convergence and will contribute to 5G multi-dimensional signals' characteristic analysis. Both theoretical proofs and simulation results are provided.

REFERENCES

[1] J. W. R. Griffiths, "Adaptive array processing. A tutorial," *IEE Proc. H-Microw., Opt. Antennas*, vol. 130, no. 1, pp. 3–10, Feb. 1983.

[2] G. Mathew and V. U. Reddy, "Development and analysis of a neural network approach to Pisarenko's harmonic retrieval method," *IEEE Trans. Signal Process.*, vol. 42, no. 3, pp. 663–667, Mar. 1994.

[3] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[4] N. Zeng, Z. Wang, B. Zineddin, Y. Li, M. Du, L. Xiao, X. Liu, and T. Young, "Image-based quantitative analysis of gold immunochromatographic strip via cellular neural network approach," *IEEE Trans. Med. Imag.*, vol. 33, no. 5, pp. 1129–1136, May 2014.

[5] X. Luo, Y. Xia, and Q. Zhu, "Applying the learning rate adaptation to the matrix factorization based collaborative filtering," *Knowl.-Based Syst.*, vol. 37, pp. 154–164, Jan. 2013.

[6] L. Xu, E. Oja, and C. Y. Suen, "Modified Hebbian learning for curve and surface fitting," *Neural Netw.*, vol. 5, no. 3, pp. 441–457, 1992.

[7] N. Zeng, Z. Wang, Y. Li, M. Du, and X. Liu, "Identification of nonlinear lateral flow immunoassay state-space models via particle filter approach," *IEEE Trans. Nanotechnol.*, vol. 11, no. 2, pp. 321–327, Mar. 2012.

[8] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Netw.*, vol. 5, no. 6, pp. 927–935, 1992.

[9] X. Luo, Z. Ming, Z. You, S. Li, Y. Xia, and H. Leung, "Improving network topology-based protein interactome mapping via collaborative filtering," *Knowl.-Based Syst.*, vol. 90, pp. 23–32, Dec. 2015.

[10] N. Zeng, Y. S. Hung, Y. Li, and M. Du, "A novel switching local evolutionary PSO for quantitative analysis of lateral flow immunoassay," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1708–1715, Mar. 2014.

[11] S. C. Douglas, S.-Y. Kung, and S. Amari, "A self-stabilized minor subspace rule," *IEEE Signal Process. Lett.*, vol. 5, no. 12, pp. 328–330, Dec. 1998.

[12] X. Luo, M. Zhou, H. Leung, Y. Xia, Q. Zhu, Z. You, and S. Li, "An incremental-and-static-combined scheme for matrix-factorization-based collaborative filtering," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 333–343, Jan. 2016.

[13] S. Ouyang, Z. Bao, G.-S. Liao, and P. C. Ching, "Adaptive minor component extraction with modular structure," *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 2127–2137, Sep. 2001.

[14] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1273–1284, May 2014.

[15] L. Jin, S. Li, H. M. La, and X. Luo, "Manipulability optimization of redundant manipulators using dynamic neural networks," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4710–4720, Jun. 2017. doi: 10.1109/TIE.2017.2674624.

[16] J. Wu, J. Long, and M. Liu, "Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm," *Neurocomputing*, vol. 148, no. 2, pp. 136–142, Jan. 2015.

[17] X. Luo, M. Zhou, S. Li, Y. Xia, Z.-H. You, Q. Zhu, and H. Leung, "Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing QoS data," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1216–1228, Apr. 2018. doi: 10.1109/TCYB.2017.2685521.

[18] S. Li, Z.-H. You, H. Guo, X. Luo, and Z.-Q. Zhao, "Inverse-free extreme learning machine with optimal information updating," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1229–1241, May 2016.

[19] N. Zeng, Z. Wang, Y. Li, M. Du, and X. Liu, "A hybrid EKF and switching PSO algorithm for joint state and parameter estimation of lateral flow immunoassay models," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 9, no. 2, pp. 321–329, Mar./Apr. 2012.

[20] W. Li, Y. Xia, M. Zhou, X. Sun, and Q. Zhu, "Fluctuation-aware and predictive workflow scheduling in cost-effective infrastructure-as-a-service clouds," *IEEE Access*, vol. 6, pp. 61488–61502, 2018.

[21] G. Cirrincione, M. Cirrincione, J. Herault, and S. V. Huffel, "The MCA EXIN neuron for the minor component analysis," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 160–187, Jan. 2002.

[22] X. Luo, M. Zhou, Z. Wang, Y. Xia, and Q. Zhu, "An effective scheme for QoS estimation via alternating direction method-based matrix factorization," *IEEE Trans. Serv. Comput.*, to be published. doi: 10.1109/TSC.2016.2597829.

[23] N. Zeng, Z. Wang, Y. Li, M. Du, J. Cao, and X. Liu, "Time series modeling of nano-gold immunochromatographic assay via expectation maximization algorithm," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 12, pp. 3418–3424, Dec. 2013.

[24] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Q. Zhu, "A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 579–592, Mar. 2016.

[25] D.-Z. Feng, Z. Bao, and L.-C. Jiao, "Total least mean squares algorithm," *IEEE Trans. Signal Process.*, vol. 46, no. 8, pp. 2122–2130, Aug. 1998.

[26] D.-Z. Feng, W.-X. Zheng, and Y. Jia, "Neural network learning algorithms for tracking minor subspace in high-dimensional data stream," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 513–521, May 2005.

[27] S. Attallah and K. Abed-Meraim, "Fast algorithms for subspace tracking," *IEEE Signal Process. Lett.*, vol. 8, no. 7, pp. 203–206, Jul. 2001.

[28] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A. C. Ammari, and A. Alabdulwahab, "Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 524–537, Mar. 2016.

[29] D. Peng, Y. Zhang, Y. Xiang, and H. Zhang, "A globally convergent MC algorithm with an adaptive learning rate," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 359–365, Feb. 2012.

[30] R. Möller, "A self-stabilizing learning rule for minor component analysis," *Int. J. Neural Syst.*, vol. 14, no. 1, pp. 1–8, 2004.

[31] P. J. Zufiria, "On the discrete-time dynamics of the basic Hebbian neural network node," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1342–1352, Nov. 2002.

[32] Q. Zhang, "On the discrete-time dynamics of a PCA learning algorithm," *Neurocomputing*, vol. 55, nos. 3–4, pp. 761–769, Oct. 2003.

[33] W. Li, K. Liao, Q. He, and Y. Xia, "Performance-aware cost-effective resource provisioning for future grid IoT-cloud system," *J. Energy Eng.*, to be published. doi: 10.1061/(ASCE)EY.1943-7897.0000611.

[34] Z. Yi, M. Ye, J. C. Lv, and K. K. Tan, "Convergence analysis of a deterministic discrete time system of Oja's PCA learning algorithm," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1318–1328, Nov. 2005.

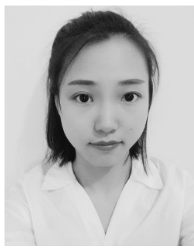
[35] D. Peng and Z. Yi, "Dynamics of generalized PCA and MCA learning algorithms," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1777–1784, Nov. 2007.

[36] N. Zeng, Z. Wang, Y. Li, M. Du, and X. Liu, "Inference of nonlinear state-space models for sandwich-type lateral flow immunoassay using extended Kalman filtering," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 7, pp. 1959–1966, Jul. 2011.
[37] X. Luo, M. Zhou, M. Shang, S. Li, and Y. Xia, "A novel approach to extracting non-negative latent factors from non-negative big sparse matrices," *IEEE Access*, vol. 4, pp. 2649–2655, 2016.
[38] J. C. Lv, Z. Yi, and K. K. Tan, "Global convergence of GHA learning algorithm with nonzero-approaching adaptive learning rates," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1557–1571, Nov. 2007.
[39] D. Peng, Z. Yi, and Y. Xiang, "A unified learning algorithm to extract principal and minor components," *Digit. Signal Process.*, vol. 19, no. 4, pp. 640–649, 2009.
[40] X. Luo, M. Shang, and S. Li, "Efficient extraction of non-negative latent factors from high-dimensional and sparse matrices in industrial applications," in *Proc. IEEE ICDM*, Dec. 2016, pp. 311–31918.
[41] H. Sakai and K. Shimizu, "A new adaptive algorithm for minor component analysis," *Signal Process.*, vol. 71, no. 3, pp. 301–308, Dec. 1998.
[42] T. Chen and S.-I. Amari, "Unified stabilization approach to principal and minor components extraction algorithms," *Neural Netw.*, vol. 14, pp. 1377–1387, Dec. 2001.
[43] X. Luo, Y. Xia, and Q. Zhu, "Incremental collaborative filtering recommender based on regularized matrix factorization," *Knowl. Based Syst.*, vol. 27, pp. 271–280, Mar. 2012.
[44] Y. Wang, H. Liu, W. Zheng, Y. Xia, Y. Li, P. Chen, K. Guo, and H. Xie, "Multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 39974–39982, 2019.
[45] Q. Peng, M. Zhou, Q. He, Y. Xia, C. Wu, and S. Deng, "Multi-objective optimization for location prediction of mobile devices in sensor-based applications," *IEEE Access*, vol. 6, pp. 77123–77132, 2018.
[46] Y. Xia, M. Zhou, X. Luo, Q. Zhu, J. Li, and Y. Huang, "Stochastic modeling and quality evaluation of infrastructure-as-a-service clouds," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 162–170, Jan. 2015.
[47] Y. Xia, M. Zhou, X. Luo, S. Pang, and Q. Zhu, "Stochastic modeling and performance analysis of migration-enabled and error-prone clouds," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 495–504, Apr. 2015.
[48] Y. Xia, M. Zhou, X. Luo, S. Pang, and Q. Zhu, "A stochastic approach to analysis of energy-aware DVS-enabled cloud datacenters," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 73–83, Jan. 2015.

PENG CHEN received the B.S. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2001, the M.S. degree in computer science from Peking University, Beijing, China, in 2004, and the Ph.D. degree in computer science and technology from the Sichuan University of China, Chengdu, in 2017. He is currently a Professor with the School of Computing and Software Engineering, Xihua University, Chengdu. His research interests include machine learning, time series analysis, and service computing.

LEI WU received the B.S., M.S., and Ph.D. degrees in computer science and technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2001, 2004, and 2010, respectively. From 2004 and 2007, he was a Lecturer with the Department of Applied Mathematics, University of Electronic Science and Technology of China. He is currently an Associate Professor with the School of Mathematical Sciences, University of Electronic Science and Technology of China. His current research interests include image processing, data mining, and pattern recognition.

DEZHONG PENG received the B.Sc. degree in applied mathematics and the M.Sc. and Ph.D. degrees in computer software and theory from the University of Electronic Science and Technology of China, Chengdu, China, in 1998, 2001, and 2006, respectively. He is currently a Professor with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu. His research interests include blind signal processing and neural networks.



XIAOYUE GONG received the B.S. degree in applied mathematics from Fuyang Normal University, in 2017. She is currently pursuing the M.S. degree in mathematics with the University of Electronic Science and Technology of China. Her research interests include natural language processing and machine learning.

VICTOR HUGO C. DE ALBUQUERQUE (M'17–SM'19) received the Ph.D. degree in mechanical engineering from the Federal University of Paraíba, the M.Sc. degree in teleinformatics engineering from the Federal University of Ceará, and the degree in mechatronics engineering from the Federal Center of Technological Education of Ceará. He is currently an Assistant VI Professor of the Graduate Program in applied informatics of UNIFOR and the Leader of the Industrial Informatics, Electronics and Health Research Group (CNPq). He is a specialist, mainly, in the IoT, machine/deep learning, pattern recognition, and robotics.