Decentralized Principal Component Analysis by Integrating Lagrange Programming Neural Networks With Alternating Direction Method of Multipliers

ZHONGHUA YE\textsuperscript{1,2,3} AND HONG ZHU\textsuperscript{1}

\textsuperscript{1}School of Automation and Information Engineering, Xi’an University of Technology, Xi’an 710048, China
\textsuperscript{2}School of Statistics, Xi’an University of Finance and Economics, Xi’an 710100, China
\textsuperscript{3}School of Electronics and Information, Northwestern Polytechnical University, Xi’an 710129, China

Corresponding author: Hong Zhu (zhuhong@xaut.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant NSFC 61471295, and in part by the Central University Funds under Grant G2016KY0308, Grant G2016KY0002, and Grant 17GH030144.

ABSTRACT

Conventional centralized methods use entire data to estimate the projection matrix of dimensionality reduction problem, which are not suitable for the network environment where the sensitive or private data are stored or there is no fusion center. In this paper, we develop a decentralized principal component analysis (DPCA) method to deal with the distributed data without sharing or collecting them together. The main contributions of this paper are as follows: i) The proposed DPCA method only needs the projection vector information communications among neighboring nodes other than the communications of the distributed data; ii) The decentralized projection vector determination problem is replaced by a set of subproblems with consensus constraints and the excellent processing capability of alternating direction method of multipliers (ADMM) is used to obtain the consistent projection vectors; iii) Especially, the integrating Lagrange programming neural networks (LPNN) is introduced to solve the projection vectors determination problem with the complex unitary and orthogonal constraints, and iv) the converge analysis of the proposed optimization problem is provided to ensure that the obtained projection vectors of the distributed method converge to those of the centralized one. Some simulations and experiments are given to show that the proposed algorithm is an alternative decentralized principal component analysis approach, and is suitable for the network environment.

INDEX TERMS

Decentralized principal component analysis (DPCA), distributed data, alternating-direction method of multipliers (ADMM), Lagrange programming neural networks (LPNN).

I. INTRODUCTION

Networks [1]–[3] are desired in diverse applications, such as security and surveillance, disaster response, and environmental modeling [4]. Conventional centralized scenario of network reserves data in a fusion center. However, the storage and computation load of the fusion center increase with the increasing data, possibly exceeding the available system resources. When the data contains the sensitive or private information, it is impossible to collet or share from other nodes. Furthermore, the entire network may be collapse with the breakdown of the fusion center, which indicates the centralized scenario is lack of robustness. Therefore, a decentralized scenario, where each network node can extract useful information by implementing some local computation, communication, and storage operations, is often considered in many applications [5]–[8].

Principal component analysis (PCA) [9], extracting low-dimensional subspaces from high-dimensional data, is one of the widely used method for dimensionality reduction in data science. Over the past decades, PCA has been used in many applications, e.g., dimensionality reduction for large, subspace learning for machine learning applications, and robust low-rank matrix recovery from missing samples or grossly corrupted data [10]–[13]. Moreover, many dimensionality reduction methods using the tensor
data are proposed to protect the inherent structure and the correlation in the original data, i.e., multilinear discriminant analysis (MDA) [14], concurrent subspace analysis (CSA) [15], discriminant locally linear embedding (DLLE) [16], and marginal Fisher analysis (MFA) [17]. However, these advances have been mostly limited to process all data at the same time in the centralized mode and have little idea on the large-scale, multi-modal, and multi-relational data sets encountered in a variety of applications, including chemometrics [18], hyperspectral imaging [19], high-resolution videos [20], biometrics [21], [22], and social network analysis [23], [24].

To solve the problem of dimensionality reduction in the distributed environment, a directed distributed PCA (DDPCA) [25] exploits a directed acyclic graphical model perspective of the network of measurements. But the DDPCA algorithm assumes a known sparsity structure in the Cholesky factor of the concentration matrix and uses an approximate iterative manner to relax the original problem, which is not suitable for the general situations and the accuracy cannot be guaranteed. Additionally, a distributed PCA algorithm [26] relies on in-network processing and does not require evaluation of the whole covariance matrix at a central location, or across all sensors. However, this distributed PCA ignores the unitary and orthogonal properties in the projection vectors and the dimensionality reduced data also needs to be shared with the neighborhood nodes. Recently, a decentralized algorithm [27], based on the integration of a dimensionality reduction step into an asynchronous Gossip consensus estimation protocol, uses all nodes of a network to compute PCA over the complete data. Although this decentralized algorithm reduces local computations and communication cost, all nodes of the decentralized algorithm still need to repeatedly send their current sum and weight to randomly selected peers, which leads to the sensitive information leaked.

On the other hand, the optimization methods, i.e., the alternating direction method of multipliers (ADMM) [28], [29] and the Lagrange programming neural network (LPNN) [30], [31] have received much attention. The former decomposes a constrained convex optimization problem into multiple smaller subproblems and provides superior convergence properties, which has been widely used in image reconstruction [32], distributed learning [6], sensor network localization [33], distributed dictionary learning [34], decentralized dimensionality reduction [35]. The latter is an analog neural computational technique for solving nonlinear constrained optimization problems according to the Lagrange multiplier theory [36]. There are two types of neurons in the network to compute the optimum solution, namely, variable and Lagrangian neurons, which are responsible for finding a minimum point of the cost function as well as the solution at an equilibrium point, and leading the dynamic trajectory into feasible region.

In this paper, we develop a decentralized principal component analysis (DPCA) method for the distributed data across nodes of the networks. It is worthwhile to highlight several aspects of the proposed algorithm here: i) The presented DPCA method can deal with the distributed data without sharing or collecting the entire data together for the network environment where the sensitive or private data are stored or there is no fusion center; ii) We convert the decentralized projection vector determination problem to a set of subproblems with consensus constraints and then only use local computations and information communications among neighboring nodes. It is noting that the information communications among neighboring nodes are the projection vectors, which differs from other distributed methods sending the corresponding information of the data; iii) To obtain the consistent projection vectors, we introduce new auxiliary variables and determine the optimization variables alternately and iteratively by ADMM; iv) Additionally, LPNN is used to solve the nonlinear programming problem with numerous complex unitary and orthogonal constraints; v) We also prove that the optimization problem is a global and convex one, which ensures that the obtained projection vectors of the distributed method converge to those of the centralized one.

The rest of this paper is organized as follows. Problem is formulated in Section II. The decentralized principal component analysis approach is developed in Section III. Experimental results are presented in Section IV. Conclusions are drawn in Section V.

Notation: Vectors and matrices are denoted by lowercase and uppercase letters, respectively. \( \| \cdot \|_p \) denotes the Frobenius norm of a vector or a matrix, \( (\cdot)^T \) denotes the transpose of a matrix or vector, \( O_{m \times n} \) and \( I_n \) denote the \( m \times n \) zero matrix and \( n \times n \) identity matrix, respectively. \( \operatorname{trace} \{ \cdot \} \) denotes the trace of a matrix. Other mathematical symbols are defined after their first appearance.

II. PROBLEM FORMULATION
Assume that there are \( K \) submatrix data \( \{Y_1, Y_2, \ldots, Y_K\} \) distributed across these \( K \) nodes, each with \( n_x \times n_y(k) \) dimensions, where \( n_x \) and \( n_y(k) \) denote the dimension and the number of the samples in matrix data \( Y_k \), respectively. It is worth noting that the value of \( n_y(k) \) can be different on account of the fact that the number of samples is not required to be the same in each node and each node can calculate independently with its locally storing data and exchange local information with its neighbors [1]–[3]. When the sub-matrices could be collected by a fusion center, the centralized schemes [9]–[13] can be applied to implement principal component analysis on them.

However, in some particular cases where the sensitive or private data are stored or there is no fusion center as mentioned in Section I, it is difficult to gather these distributed sub-matrices and implement PCA on the distributed submatrices [5]–[8]. Therefore, this paper proposes a decentralized counterpart to deal with the distributed sub-matrix data by the local computations and the communications among neighboring nodes. In this way, all nodes can attain approximate or even the same performance as that of the centralized counterpart (if it existed).
To meet with the distributed processing requirement of the networks, this subsection develops a decentralized principal component analysis algorithm to use the submatrix data \{Y_1, Y_2, \ldots, Y_K\} distributed across the \(K\) nodes only by means of the local computations and the communications among neighboring nodes rather than collecting the private information \{Y_1, Y_2, \ldots, Y_K\} together.

**A. PRINCIPAL COMPONENT ANALYSIS DETERMINATION**

The principal component analysis (PCA) is a typical method for dimensionality reduction. Considering the collected data \(Y = [Y_1, Y_2, \ldots, Y_K] \in \mathbb{R}^{n_r \times n_c}\) can be represented as

\[
\min_{U, V} \| Y - U U^T Y \|_F^2
\]

\[
s.t. \quad U^T U = I_L,
\]

where \(n_c\) represents the total number of the samples, \(U = [u_1, u_2, \ldots, u_L] \in \mathbb{R}^{n_r \times L}\) is the orthogonal projection matrix with unitary vectors, and \(L < n_r\) denotes the number of the projection vector.

Then (1) can be rewritten as the following optimization problem:

\[
\min_{[u_i]} - \sum_{i=1}^{L} u_i^T Y Y^T u_i
\]

\[
s.t. \quad u_i^T u_i = 1, \quad u_i^T u_j = 0, \quad i \neq j, \quad i, j = 1, \ldots, L.
\]

Therefore, we obtain the variables \([u_i]\) by an iterative manner to \(i = 1, \ldots, L\).

For the first projection vector \(u_1\) of \(U\), it can be determined by solving the following optimization problem:

\[
u_1 = \arg \min_u - u^T Y Y^T u
\]

\[
s.t. \quad u^T u = 1,
\]

where the corresponding proof can be found in Appendix A.

Additionally, we can determine the \(n\)-th projection vector by

\[
u_n = \arg \min_u - u^T Y Y^T u
\]

\[
s.t. \quad u^T u = 1, \quad u^T u_j = 0, \quad j = 1, \ldots, n-1, \quad n > 1,
\]

where the proof is shown in Appendix B.

**B. DECENTRALIZED PRINCIPAL COMPONENT ANALYSIS DETERMINATION**

It seems that we can determine the components of \(U\) sequentially according to the afore-mentioned strategy. However, the data, i.e., \(Y = [Y_1 Y_2 \cdots Y_K]\), is distributed across the nodes of the networks, which cannot be gathered together due to the fact the sensitive or private data are stored or there is no fusion. Therefore, the projection vector \(u\) must be determined by the local data in each node rather than the entire collected data.

Since (3) is just the special case of (4), we consider the distributed solution to (4) in the rest of this paper. We rewrite (4) in a separated matrix form as

\[
\min_u - u^T \left( \sum_{k=1}^{K} Y_k Y_k^T \right) u
\]

\[
= \min_u - \sum_{k=1}^{K} u^T Y_k Y_k^T u
\]

\[
s.t. \quad u^T u = 1,
\]

\[
u_j^T u = 0, \quad j \in \{1, \ldots, n-1\}.
\]

To operate all the distributed \([Y_k]\) \(k=1\) directly, we replace \(u\) in (5) with \(K\) copies, denoted as local variables \([u^1, u^2, \ldots, u^K]\) distributed across the \(K\) nodes, yielding the equivalent form as follows (here we omit the constraints in (5) for representation convenience):

\[
\min_{[u^k]} - \sum_{k=1}^{K} u^k Y_k Y_k^T u^k
\]

\[
s.t. \quad u^k = u^{k'}, \quad k' \in N_e(k),
\]

where \(N_e(k)\) stands for the neighbors of the \(k\)th node for \(k = 1, \ldots, K\), and the equivalent constraints denote the so-called consensus constraints [6].

**III. ALGORITHM DEVELOPMENT**

In this section, we explore a solution to the problem of consensus local variables \([u^1, u^2, \ldots, u^K]\) for \(k = 1, \ldots, K\).

**A. PROPOSED SOLUTION**

To update \(u^k\) and \(u^{k'}\) separately, we define new vector \(w^k\) with the related equivalent constraints of (6):

\[
\min_{[u^k, w^k]} - \sum_{k=1}^{K} (u^k)^T Y_k Y_k^T u^k
\]

\[
s.t. \quad u^k = w^{k'}, \quad u^{k'} = w^k, \quad k' \in N_e(k), \quad k = 1, \ldots, K.
\]

Note that (7) is a optimization function with equality constraints. Therefore, the augmented Lagrangian of (7) can be rewritten in an unconstraint form according to ADMM [28], [29]:

\[
\min_{[u^k, w^k, \lambda_{k,k'}: \lambda_{k,k'}]} - \sum_{k=1}^{K} (u^k)^T Y_k Y_k^T u^k
\]

\[
+ \sum_{k=1}^{K} \lambda_{k,k'} (u^k - w^k)
\]

\[
+ \frac{\rho}{2} \sum_{k=1}^{K} ||w^k - w^k||_F^2
\]

\[
+ \sum_{k=1}^{K} \sum_{k' \in N_e(k)} \lambda_{k,k'} (u^k - w^k)
\]

\[
+ \frac{\rho}{2} \sum_{k=1}^{K} \sum_{k' \in N_e(k)} ||w^k - w^k||_F^2.
\]
where $\lambda_{k,k'}$ and $\lambda_{k,k}$ are Lagrange multipliers corresponding to the constraints $u^k = w^k$ and $u^k = w^k$ for $k = 1, \ldots, K$ and $k' \in Ne(k)$, and $\rho > 0$ is the augmented Lagrangian parameter. Then $\{u^k, w^k, \lambda_{k,k'}, \lambda_{k,k}\}$ can be determined by an iterative method.

With given $\{w^k(t), \lambda_{k,k'}(t), \lambda_{k,k}(t)\}$, we solve $\{u^k(t + 1)\}$ using the following optimization problem:

$$
\{u^k(t + 1)\} = \arg \min_{u^k} - \sum_{k=1}^{K} (u^k)^T Y_k Y_k^T u^k + \frac{\rho}{2} \sum_{k=1}^{K} \sum_{k' \in Ne(k)} \lambda_{k,k'}^T (u^k - w^k) + \frac{\rho}{2} \sum_{k=1}^{K} \sum_{k' \in Ne(k)} \lambda_{k,k'}^T (u^k - w^k) + \frac{\rho}{2} \sum_{k=1}^{K} \sum_{k' \in Ne(k)} ||u^k - w^k||_F^2, \tag{9}
$$

where $t$ is the iteration number.

Then $u^k(t + 1)$ can be solved in a separate and parallel form [28], [29], for $k = 1, \ldots, K$. Therefore, adding unitary and orthogonality constraints, we divide (9) into $K$ subproblems:

$$
u^k(t + 1) = \arg \min_{u^k} - (u^k)^T Y_k Y_k^T u^k + \lambda_{k,k'}^T (u^k) u^k + \frac{\rho}{2} \sum_{k=1}^{K} \sum_{k' \in Ne(k)} ||u^k - w^k||_F^2
$$

s.t. $(u^k)^T u^k = 1,$

$$(u^k)^T u^j = 0, \text{ for } j = 1, \ldots, n - 1, \tag{10}$$

where Lagrange multipliers $\lambda_{k}(t) = \lambda_{k,k} + \sum_{k' \in Ne(k)} \lambda_{k,k'}(t), k = 1, \ldots, K$.

With given $\{u^k(t + 1), \lambda_{k,k'}(t), \lambda_{k,k}(t)\}$, $\{w^k(t + 1)\}$ can be solved by

$$
\{w^k(t + 1)\} = \arg \min_{w^k} - \sum_{k=1}^{K} (u^k(t + 1))^T Y_k Y_k^T u^k(t + 1)
$$

+ $\frac{\rho}{2} \sum_{k=1}^{K} \sum_{k' \in Ne(k)} ||u^k(t + 1) - w^k||_F^2$ + $\frac{\rho}{2} \sum_{k=1}^{K} \sum_{k' \in Ne(k)} ||u^k(t + 1) - w^k||_F^2$.

Similarly, (11) can be solved in a separate and parallel form:

$$
\{w^k(t + 1)\} = \arg \min_{w^k} - \lambda_{k,k'}^T (w^k(t + 1) - w^k) + \frac{\rho}{2} \sum_{k=1}^{K} \sum_{k' \in Ne(k)} ||w^k(t + 1) - w^k||_F^2. \tag{11}
$$

where Lagrange multipliers $\lambda_{k}(t) = \lambda_{k,k} + \sum_{k' \in Ne(k)} \lambda_{k,k'}(t), k = 1, \ldots, K$. Additionally, the symmetry of the undirected graph is used in (12), i.e., if $k' \in Ne(k)$, then $k \in Ne(k')$.

Therefore, $\{u^k, w^k\}$ can be determined only by processing the local data $\{Y_k\}$ and exchanging $\{w^k(t), u^k(t)\}$, which is a competitive manner compared to the distributed PCA methods [25]-[27].

With given $\{u^k(t + 1), w^k(t + 1), \lambda_{k,k'}(t), \lambda_{k,k}(t)\}$ and considering the relationships between $\{\lambda_{k,k}, \lambda_{k,k'}\}$ and $\{\lambda_{k}, \lambda_{k'}\}$, we update $\lambda_{k}(t + 1)$ and $\lambda_{k}(t + 1)$ as follows:

$$
\lambda_{k}(t + 1) = \lambda_{k}(t) + \rho(\text{Card}(k) + 1)u^k(t + 1) - \rho w^k(t + 1)
$$

and

$$
\lambda_{k'}(t + 1) = \lambda_{k'}(t) + \rho(\text{Card}(k) + 1)u^k(t + 1) - \rho w^k(t + 1), \tag{14}
$$

where $\text{Card}(k)$ denotes the neighbor number of the $k$th node, $k = 1, \ldots, K, k' \in Ne(k)$.

However, (10) and (12) are complex programming problems. Especially, (10) can be represented as follows ((12) has the similar form, and thus it is omitted here):

$$
\min_{u^k} (u^k)^T A_k u^k + (b^k)^T u^k
$$

s.t. $(u^k)^T u^j = 1,$

$$(u^k)^T u^j = 0, \text{ for } j = 1, \ldots, n - 1, \tag{15}$$

where

$$
A_k = -Y_k Y_k^T + \frac{\rho}{2} (\text{Card}(k) + 1) I_{n_k}, \tag{16}
$$

and

$$
b^k = \lambda_{k}(t) - \rho \left( w^k(t) + \sum_{k' \in Ne(k)} w^k(t) \right). \tag{17}
$$

For the problem in (15), we can construct the following Lagrangian function according to the LPNN approaches [30], [31]:

$$
\mathcal{L}(u^k, \xi_u, \xi_w) = \min_{u^k} \left( (u^k)^T A_k u^k + (b^k)^T u^k + \xi_u ((u^k)^T u^k - 1) + \sum_{j=1}^{n-1} \xi_w (u^j)^T u^j, \right) \tag{18}
$$
where \( u^k \) is state variables, while \( \xi_u \) and \( \{\xi_o(j)\}_{j=1}^{n-1} \) are Lagrange variables. That means, there are \( n_r \) state variable neurons to hold state variables \( u^k, n \) state Lagrangian neurons to hold the Lagrangian variables.

With (18), the dynamic of the network is given by

\[
\begin{aligned}
\frac{du^k}{dt} &= -\frac{\partial L(u^k, \xi_u, \{\xi_o(j)\}_{j=1}^{n-1})}{\partial u^k} \\
&= -2A_k u^k - b^k \\
&= -2\xi_u u^k - \sum_{j=1}^{n-1} \xi_o(j) u^k_j,
\end{aligned}
\]

\( \frac{d\xi_u}{dt} = \frac{\partial L(u^k, \xi_u, \{\xi_o(j)\}_{j=1}^{n-1})}{\partial \xi_u} \\
= (u^k)^T u^k - 1,
\]

\( \frac{d\xi_o(j)}{dt} = \frac{\partial L(u^k, \xi_u, \{\xi_o(j)\}_{j=1}^{n-1})}{\partial \xi_o(j)} \\
= (u^k_j)^T u^k, \quad \text{for } j = 1, \ldots, n - 1,
\]

where \( t \) is the time variable.

Then the decentralized principal component analysis (DPCA) method can be described as follows:

---

**Decentralized Principal Component Analysis (DPCA) Method**

**Initialization**: Setting the same initial normalized \( u \) and \( \rho \) for all nodes;

for \( n = 1, \ldots, L \)
for \( t = 1, \ldots, T_{\text{max}} \)
    for the \( k \)th node, solving the local eigenvector \( u^k(t + 1) \) of (10) using the LPNN steps shown in (18)-(21), for \( k = 1, \ldots, K \):
    Broadcast \( u^k(t + 1) \) to the related neighbors;
    Broadcast \( w^k(t + 1) \) to the related neighbors;
    Update \( \lambda_k(t + 1) \) using (13);
    Update \( \lambda_{\rho}(t + 1) \) using (14);
end for \( t \) or the prespecified criterion is satisfied.
end for \( n \)

---

**B. DISCUSSION OF THE PROPOSED ALGORITHM**

Note that LPNN is an analog neural computational technique for solving nonlinear constrained optimization problem and some basic properties are required when we use LPNN method to solve (15). Then, by analysing (15)-(21), we can obtain the following lemma:

**Lemma**: If the gradient vector of the constraints at an equilibrium point is linearly independent, then the requirement for the local stability of LPNN is met [30], [31].

**Proof**: For our problem, the gradient vector of the constraint \((u^k)^T u^k = 1\) at an equilibrium point \((u^k)^*\) is \(2(u^k)^*\) and other constraint \((u_j^k)^T u^k = 0\) at the point is \(u_j^k\), \(j = 1, \ldots, n - 1\). Therefore, \(2(u^k)^*\) and \(u_j^k, j = 1, \ldots, n - 1\) are orthogonal to each other, i.e., they are linearly independent. The proof is complete.

Additionally, the linearized neural dynamics around \((u^k, \xi_u, \{\xi_o(j)\}_{j=1}^{n-1})\) is given by:

\[
\begin{bmatrix}
\frac{du^k}{dt} \\
\frac{d\xi_u}{dt} \\
\frac{d\xi_o(j)}{dt}
\end{bmatrix} = -G \begin{bmatrix}
\xi_{-1} \\
\xi_{-1}
\end{bmatrix},
\]

where

\[
G = \begin{bmatrix}
2A_k + 2\xi_u n_r & B \\
-B^T & 0_{n \times n_r}
\end{bmatrix},
\]

and \( \xi = [\xi_u \xi_o(1) \cdots \xi_o(n-1)]^T \). Then we prove that \( G \) is positive definite in Appendix C.

Moreover, we evaluate the required computational complexity of the proposed DPCA method. It can be found that the computational complexity of updating \( u^k \) occupies a major component in each iteration of the proposed method and it requires \( O(2n^2) \) flops and broadcasts \( \text{Card}(k) \) vectors to its neighbors for each node.

**C. CONVERGENCE ANALYSIS**

In this subsection, we prove that the optimization problem shown in (15) is a global and convex one in a hidden form, and it satisfies Lemma 4.1 in [8], which ensures that the obtained atoms by the distributed method converge to the centralized one.

First we prove (15) is a global maximizer and is convex in a hidden form. In other words, the original optimization problems are actually both global and convex ones in a hidden form.

That is to say first we should prove that the problem (15) is equivalent to that with such an inequality constraint:

\[
\min_{u^k} (u^k)^T A_k u^k + (b^k)^T u^k
\]

s.t. \((u^k)^T u^k \leq 1, (u^k)^T u_j^k = 0, \quad \text{for } j = 1, \ldots, n - 1. \quad (25)

Under the inequality constraint \((u^k)^T u^k \leq 1\), we assume that the minimal value of \((u^k)^T A_k u^k + (b^k)^T u^k\) with respect to \( u^k \) will appear at \((u^k)^T u^k = \beta\), where \(0 < \beta < 1\). Thus, (25) is transformed to the following form:

\[
\min_{u^k} (u^k)^T A_k u^k + (b^k)^T u^k
\]

s.t. \((u^k)^T u^k = \beta, (u^k)^T u_j^k = 0, \quad \text{for } j = 1, \ldots, n - 1. \quad (26)

We define Lagrangian function like (18):

\[
\begin{align*}
\tilde{L}(u^k, \xi_u, \{\xi_o(j)\}_{j=1}^{n-1})
&= (u^k)^T A_k u^k + (b^k)^T u^k + \xi_u((u^k)^T u^k - \beta)
\end{align*}
\]
+ \sum_{j=1}^{n-1} \xi_u(j)(u^k_j)^T u^k. \tag{27}

From which we define the related dual function as follows:
\[ \tilde{F}(\xi_u) = \sum_{i=1}^{n} \frac{(\tilde{b}^T v_i)^2}{(\tilde{q}_i + \xi_u)^2} = 4\beta. \tag{28} \]

It is easy to notice that \( \tilde{F}(\xi_u) \) is also a monotonically decreasing function of \( \xi_u \), and \( \lim_{\xi_u \to \infty} \tilde{F}(\xi_u) = -4\beta \).\lim_{\xi_u \to -\infty} \tilde{F}(\xi_u) = +\infty. \) Therefore, if \( \beta_1 > \beta_2 \), and \( \beta_1, \beta_2 \in (0, 1] \), the root of
\[ \tilde{F}(\xi_u) = \sum_{i=1}^{n} \frac{(\tilde{b}^T v_i)^2}{(\tilde{q}_i + \xi_u)^2} - 4\beta_1 = 0 \tag{29} \]
is certainly not larger than that of
\[ \tilde{F}(\xi_u) = \sum_{i=1}^{n} \frac{(\tilde{b}^T v_i)^2}{(\tilde{q}_i + \xi_u)^2} - 4\beta_2 = 0. \tag{30} \]

In other words, the larger the scalar \( \beta \in (0, 1] \) is, the smaller the root \( \xi_u \) of \( \tilde{F}(\xi_u) = 0 \) is. Thus the larger the scalar \( \beta \in (0, 1] \) is, the smaller the primal objective function is. Note that 1 is the largest value available for \( \beta \). Therefore, the objective function (15) is equivalent to
\[ \min_{u^k} (u^k)^T A_k u^k + (b^k)^T u^k \]
s.t. \((u^k)^T u^k \leq 1,
(u^k)^T u^k_j = 0, \quad j = 1, \cdots, n-1, \tag{31} \]
and achieves its minimum value at boundary \((u^k)^T u^k = 1\) and \((u^k)^T u^k_j = 0 \). Note that the set defined by
\[ \{u \in C | (u^k)^T u^k \leq 1 \text{ and } (u^k)^T u^k_j = 0\} \tag{32} \]
is a convex set. We can prove that C is a convex set, the proof process is as follows.

Assuming that \( u_1 \in C, u_2 \in C, \) and \( \theta \in [0, 1] \), then
\( (\theta u_1 + (1 - \theta) u_2)^T (\theta u_1 + (1 - \theta) u_2) \]
\[ = \theta^2 (u_1)^T (u_2) + (1 - \theta)^2 (u_2)^T (u_2) \leq 1, \tag{33} \]
and
\[ (\theta u_1 + (1 - \theta) u_2)^T u_j = 0, \tag{34} \]
\( \theta u_1 + (1 - \theta) u_2 \in C, \) thus \( C \) is a convex set. The objective function, \((u^k)^T u^k\), and \((u^k)^T u^k_j\) in (15) are all convex functions. Besides, according to (61), we can see that the matrix \( A_k + \xi_u I \) is positive definite. Therefore, according to Theorem III as well as the \( \varepsilon \)-subdifferentials of convex functions and \( \varepsilon \)-normal directions in [8], the sufficient and necessary conditions in above can ensure that (15) is a global maximizer and convex in a hidden form. In other words, the original optimization problems is a global and convex one in a hidden form.

Second, we should prove the similar lemma (as Lemma 4.1 in [8]) holds for the hidden convex functions above, which ensures that the obtained atoms by the distributed method converge to that of the centralized one. Similar to the part between Eqs. (4.86) and (4.87) of Lemma 4.1 in [8], we define
\[ u^k_u = \arg \min_{u^k \in S} (J_1(u^k) + J_2(u^k)) \]
\[ = \arg \min_{u^k \in S} (u^k)^T A_k u^k + (b^k)^T u^k, \]
\[ \{u^k \in S | (u^k)^T u^k = 1 \text{ and } (u^k)^T u^k_j = 0\}, \tag{35} \]
where \( J_1(u) = (u^k)^T A_k u^k \) and \( J_2(u) = (b^k)^T u^k \) are continuously differentiable. Then (35) equals to
\[ \min_{u^k} (u^k)^T A_k u^k + (b^k)^T u^k \]
s.t. \((u^k)^T u^k = 1, \tag{36} \]
\[ (u^k)^T u^k_j = 0, \quad \text{for } j = 1, \cdots, n-1. \]

Therefore, according to Lagrange multiplier method, the solution of (36) is
\[ u^k_u = -\frac{1}{2} (A_k + \xi_u^* I)^{-1} (b^k + \sum_{j=1}^{n} \xi_u^*(j) u^k_j) \]
\[ = -\frac{1}{2} (A_k + \xi_u^* I)^{-1} \tilde{b}, \tag{37} \]
where \( \xi_u^* \) satisfies
\[ ((A_k + \xi_u^* I)^{-1} \tilde{b})^T (A_k + \xi_u^* I)^{-1} \tilde{b} \]
\[ = \sum_{i=1}^{n} (\tilde{b}^T v_i)^2 = 4. \tag{38} \]

According to Eq. (4.85) in [8],
\[ u^k_u = \arg \min_{u^k \in S} (J_1(u^k) + [\nabla J_2(u^k)]^T u^k) \]
\[ = \arg \min_{u^k \in S} (u^k)^T A_k u^k + (b^k)^T u^k = u^k_u, \tag{39} \]
which shows that Lemma 4.1 in that book [8] holds for (15). Therefore, it also holds for the hidden convex objective functions shown in (5).

Based on the derivation above, we let \( G_1(a) \) denotes the hidden convex optimization problem shown in (15), and \( G_2(a) = 0 \), as those of [8]. Then the constraints \((u^k)^T u^k = 1 \) and \((u^k)^T u^k_j = 0 \) are always bounded and every limit point of \( \{u^k\} \) is an optimal solution to the original problem (15). Due to that \( G_1(a) \) shown in (15) satisfies the Lemma 4.1 and by virtue of Proposition 4.2 [8], we follow the derivations shown in Eqs. (4.87)-(4.99) [8] and thus the atoms \( \{u^k\} \) obtained by the proposed decentralized method converge to the optimal \( \{u^k\} \) shown in original problem.

IV. SIMULATION AND EXPERIMENTAL RESULTS

In this section, the simulation data, open Handwritten Digit database [37], and face database [38] are used to evaluate the performance of the proposed DPCA method for the distributed stored data across the networks.
TABLE 1. Digit distribution for training and testing.

|       | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | Total |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-------|
| Training | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 250 | 2500  |
| Testing  | 339 | 264 | 198 | 166 | 200 | 160 | 170 | 147 | 166 | 177 | 2007  |

A. DPCA ON SIMULATION DATA

In the first experiment, we consider a network with \( K = 20 \) nodes. Each node is able to process the simulation data to extract relevant information via collaborating with neighboring nodes.

In this experiment, we use 500000 simulation samples to the 20 nodes at random. Additionally, the randomly-generated normalized column vector is used as the initializes projection vectors \( \{u_k\}_{k=1}^{20} \), the step size \( \rho = 0.01 \), and the maximum iteration number \( T_{\text{max}} = 2000 \).

To evaluate the consistency and the convergence performance of the proposed DPCA algorithm, we define the maximum absolute coefficient difference error between the projection matrices obtained by the centralized (gathering all data together) and distributed manners as

\[
\xi(k, t, n) = \text{MAX}(|u^k_n(t) - u_n|),
\]

\( k = 1, \ldots, K, \quad t = 1, \ldots, T_{\text{max}}, \quad n = 1, \ldots, L, \quad (40) \)

where MAX(.) denotes the operator of choosing the maximum value of . and \( \xi(k, t, n) \) corresponds to the \( t \)th iteration and \( n \)th projection vector at the \( k \)th node.

Then we compute \( \xi(k, t, n) \) using the obtained projection matrices of the proposed DPCA algorithm and only plot the corresponding results of the special case \( n = 1 \) in Fig. 1. It can be found that the maximum absolute coefficient difference error of each node is decreasing with the increasing iteration number and all of the maximum absolute coefficient difference errors between the distributed and centralized ones are less than \( 10^{-9.8442} \) after 900 iterations. Therefore, the proposed algorithm has the satisfactory convergence performance and can obtain consistent projection vector (i.e., \( n = 1 \)) with negligible difference for all the nodes. Additionally, other results, i.e., \( n = 2, \ldots, L \), are similar to that of the case \( n = 1 \). Therefore, the proposed DPCA is capable of obtaining the consistent projection matrices compared with the centralized cases.

B. DPCA ON HANDWRITTEN DIGIT DATABASE

In the second experiment, we consider a network with \( K = 5 \) intelligent post offices (IPOs). Each IPO is able to obtain grayscale digit images of the local postal mails, and also can process the image data to extract relevant information via collaborating with neighboring IPOs.

In this experiment, the two sets available from the USPS database [37] are used as the training and testing samples, i.e., 2500 of the set with 7291 16 \( \times \) 16 digit images shown in Fig. 2 are used for training and the set with 2007 ones is set for testing. The related distributions are given in Table 1.

Similarly, we compute \( \xi(k, t, n) \) to evaluate the consistency and the convergence performance of the proposed DPCA algorithm and plot it in Fig. 3. It shows that all of the maximum absolute coefficient difference errors between the distributed and centralized ones are less than \( 10^{-14.7257} \) after 100 iterations. Therefore, the convergence performance and consistent projection vectors can be guaranteed in this proposed DPCA method.

Due to the efficient consistency property of the proposed DPCA method in each node, we only use the projection matrices of the 1st node to assess the DPCA algorithm for the recognition task. When the feature number vary from 1 to 16, the related recognition rates of the proposed DPCA method are given in Fig. 4. For comparison purpose, other results from centralized PCA [9], MDA [14], CSA [15], DLLE [16], MFA [17], and the decentralized DDR [35] algorithms
FIGURE 4. Recognition rates versus feature number.

FIGURE 5. The sample images cropped from the ORL database.

are also plotted in Fig. 4. It can be seen from Fig. 4 that: i) the recognition rates of all the seven methods rise with the increase of the feature number (from 1 to 16); ii) the best handwritten digit recognition rates achieved by PCA, MDA, CSA, DLLE, MFA, DDR, and DPCA are 0.9162, 0.9063, 0.9243, 0.9143, 0.9243, 0.9243, and 0.9168, respectively; iii) the DPCA method with the distributed computation manner has the approximate recognition property as the centralized dimensionality reduction methods; iv) Compared to the decentralized DDR method, the DPCA method obtained the similar recognition rates but has a lower complexity.

C. DPCA ON OLIVETTI & ORACLE RESEARCH LABORATORY (ORL) FACE DATABASE

The ORL database [38] is used in the third experiment to simulate the data of the vision network, where the faces of a subject were captured by different-view cameras in the network, and each observation represents a facial image captured at different times and with different variations including expression (open or closed eyes, smiling or non-smiling) and facial details (glasses or no glasses). Moreover, 400 images of 40 individuals were in grayscale and normalized to the resolution of 112 × 92 pixels in the ORL database [38] and a tolerance for some tilting and rotation of the face up to 20° was taken in each image. Fig. 5 shows a snapshot of a person captured by the intelligent cameras with the corresponding viewing angles.

To simulate the network, we use 240 images and the rest of the database as the training and testing samples, respectively. The training images are arranged to a network with K = 5 nodes. Similarly, the same step size and the iteration number are used as the first experiment. Furthermore, we also compute \( \xi(k, t, n) \) and plot it in Fig. 6. From Fig. 6, we can find that they are all not larger than \( 10^{-8.0664} \) after 100 iterations, which imply that the proposed algorithm has satisfactory convergence and consistent performance.

Additionally, to obtain the classification of the testing samples, we also use the nearest neighbor criterion and show the recognition rates of the seven algorithms in Fig. 7. From Fig. 7, we can see that the best recognition rates of PCA, MDA, CSA, DLLE, MFA, DDR, and DPCA are 0.9750, 0.9625, 0.9687, 0.9750, 0.9812, 0.9750, and 0.9750, which indicate that the recognition performance of the proposed DPCA method is comparable to those of PCA, MDA, CSA, DLLE, MFA, and DDR. Although MFA method obtains the highest recognition rate when the dimensions increase to 1110, the DDR and DPCA methods can handle the sensitive or private data in a distributed manner to obtain the satisfactory result.

V. CONCLUSION

In this paper, we have developed an efficient DPCA method to process the distributed data across the networks. To circumvent the decentralized projection vector determination problem in the network environment only by local computations and information communications among neighboring nodes, we update the optimization variables alternately and iteratively, and determine the projection vectors with unitary modulus and orthogonal constraints in terms of the
remarkable capability of Lagrange programming neural networks in solving general nonlinear programming. Numerical examples have shown that the proposed DPCA method outperforms the state-of-the-art techniques for distributed data across the networks.

**APPENDIX A**

Let the eigenvalue decomposition (EVD) of $YY^T$ be

$$ YY^T = U \Lambda U^T, $$

where $U = [u_1, u_2, \cdots, u_{n_r}]$ and $\Lambda = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_{n_r})$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{n_r} \geq 0$. According to the Rayleigh-Ritz Theorem, we have

$$ u^T YY^T u = u^T U \Lambda U^T u = (U^T u)^T \Lambda (U^T u) = \sum_{i=1}^{n_r} \sigma_i |(U^T u)_i|^2, $$

(42)

for any $u \in \mathbb{R}^{n \times 1}$, where $(U^T u)_i$ denotes the $i$th element of $U^T u$.

Since $U$ is unitary and $U^TU = I_{n_r}$, we have

$$ \sum_{i=1}^{n_r} |(U^T u)_i|^2 = u^T \sum_{i=1}^{n_r} u_i u_i^T u = u^T u. $$

(43)

Inserting (43) into (42) yields

$$ \sigma_{n_r} u^T u \leq \sum_{i=1}^{n_r} \sigma_i |(U^T u)_i|^2 = u^T YY^T u \leq \sigma_1 u^T u. $$

(44)

Therefore, if $u$ is an eigenvector of $YY^T$ corresponding to the eigenvalue $\sigma_1$, then $u^T YY^T u$ obtains the maximum value $\sigma_1$.

Furthermore, if $u \neq 0$, we have

$$ \frac{u^T YY^T u}{u^T u} = \left( \frac{u}{\sqrt{u^T u}} \right)^T YY^T \left( \frac{u}{\sqrt{u^T u}} \right), $$

(45)

and

$$ \left( \frac{u}{\sqrt{u^T u}} \right)^T \left( \frac{u}{\sqrt{u^T u}} \right) = 1, $$

(46)

which shows that (3) holds. The proof is complete.

**APPENDIX B**

We begin with the $n = 2$ case where the optimized $u$ should satisfy with $u^T u_1 = 0$, and thus we have

$$ u^T YY^T u = \sum_{i=1}^{n_r} \sigma_i u_i u_i^T u = \sum_{i=2}^{n_r} \sigma_i u_i u_i^T u, $$

(47)

where the component $\sigma_1 u_1 u_1^T$ is dropped due to $u^T u_1 = 0$.

Define $R_1 = \sum_{i=2}^{n_r} \sigma_i u_i u_i^T = YY^T - \sigma_1 u_1 u_1^T$, then the problem in (4) reduces to

$$ \min_{u} - u^T R_1 u $$

s.t. $u^T u = 1$. (48)

Note that (48) is very similar to (3), thus the solution to (48) is the eigenvector to the largest eigenvalue of $R_1$.

Similar proofs hold for $n = 3, \cdots, L$. The proof is complete.

**APPENDIX C**

Since $G$ is strictly positive definite provided that $2A_k + 2\xi_u I_{n_r}$ is positive definite. Let $\alpha$ be an eigenvector of $G$, and let $[z^T s^T]$ be a solution with the zero vector, be the corresponding eigenvector where $z$ and $s$ are vectors of dimension of $n_r$ and $n$, respectively. We have

$$ [z^T s^T] G [z^T s^T] = \alpha (|z|^2 + |s|^2), $$

(49)

Inserting the definition of $G$ into (49) yields

$$ [z^T s^T] \left( 2A_k + 2\xi_u I_{n_r} \right) [z^T s^T] = \alpha (|z|^2 + |s|^2). $$

(50)

where $z^T s - s^T B s = 0$ is applied.

Therefore, combining (49) and (50), we obtain

$$ z^T (2A_k + 2\xi_u I_{n_r}) z = \alpha (|z|^2 + |s|^2). $$

(51)

According to the KKT condition, we have

$$ \frac{\partial \mathcal{L}(u^k, \xi_u, \{\xi_o(j)\}^n_{j=1})}{\partial u^k} = 2A_k u^k + b + 2\xi_u u^k + \sum_{j=1}^{n-1} \xi_o(j) u_j^k $$

(52)

$$ = (2A_k + 2\xi_u) u^k + (b + \sum_{j=1}^{n-1} \xi_o(j) u_j^k) $$

(53)

which shows that (3) holds. The proof is complete.

In other words, it should satisfy with the following $n_r + n$ equations:

$$ \begin{bmatrix} 2A_k + 2\xi_u I_{n_r} & B \\ B^T & 0_{n \times n_r} \end{bmatrix} \begin{bmatrix} u^k \\ \xi_o(1) \\ \vdots \\ \xi_o(n-1) \end{bmatrix} = \begin{bmatrix} -b^k \\ 0 \\ \vdots \\ 0 \end{bmatrix} $$

(55)
If there is a unique optimization solution to the problem, then the variable \( (u^k)^T \mathbf{0} \xi_u(1) \cdots \xi_u(n-1) \) is unique, which implies that the equations have a unique solution. Therefore, the matrix \( \begin{bmatrix} 2A_k + 2\xi_u I_n & B \\ B^T & \mathbf{0}_{n \times n} \end{bmatrix} \) is inverse.

Furthermore, we prove that the matrix \( 2A_k + 2\xi_u I_n \) is inverse.

In fact, \( (u^k)^T \mathbf{0} \xi_u(1) \cdots \xi_u(n-1) \) is the function of \( \xi_u \).

Based on (52), we have

\[
u^k = -\frac{1}{2}(A_k + \xi_u I_n)^{-1}\tilde{b}.
\] (56)

where

\[
\tilde{b} = b^k + \sum_{j=1}^{n-1} \xi_u(j)u^k.
\] (57)

Submitting (56) into (53), we have

\[
((A_k + \xi_u I_n)^{-1}\tilde{b})^T(A_k + \xi_u I_n)^{-1}\tilde{b} = 4.
\] (58)

With the definition that the EVD of the symmetric matrix \( A_k \) with \( \zeta_1 \geq \zeta_2 \geq \cdots \geq \zeta_n \) is given by

\[
A_k = \sum_{i=1}^{n} \zeta_i v_i v_i^T,
\] (59)

(58) becomes

\[
F(\xi_u) = \sum_{i=1}^{n} (\tilde{b}^Tv_i)^2 = 4,
\] (60)

which implies that \( F(\xi_u) \) attains +\( \infty \) and 0 at \( \xi_u = -\zeta_n \) and +\( \infty \), respectively. Additionally, \( F(\xi_u) \) is a monotonically decreasing function in \( \xi_u \in [-\zeta_n, +\infty) \) and also there is a unique solution \( \xi_u \) to \( F(\xi_u) = 4 \). Thus, \( \xi_u > -\zeta_n \). Then

\[
A_k + \xi_u^* I_n = \sum_{i=1}^{n} (\zeta_i + \xi_u^*) v_i v_i^T,
\] (61)

with all eigenvalues \( \zeta_i + \xi_u^* \) are larger than 0 due to that \( \zeta_u^* > -\zeta_n \), and \( \zeta_1 \geq \zeta_2 \geq \cdots \geq \zeta_n \), which implies the matrix \( A_k + \xi_u^* I_n \) is positive definite.

Since for any positive definite matrix \( 2A_k + 2\xi_u I_n \), we have

\[
\xi_u^T(2A_k + 2\xi_u I_n)\xi_u > 0, \quad \xi_u \neq 0_{n \times 1}.
\] (62)

Therefore, either \( \alpha > 0 \) or \( z = 0_{n \times 1} \) must be met to ensure the positive definiteness assumption on \( G \).

According to the definition of eigenvectors, we have

\[
G \begin{bmatrix} z \\ s \end{bmatrix} = \alpha \begin{bmatrix} z \\ s \end{bmatrix}.
\] (63)

Additionally, we obtain the following equation with the definition of \( G \):

\[
G \begin{bmatrix} z \\ s \end{bmatrix} = \begin{bmatrix} B s \\ 0_{n \times 1} \end{bmatrix},
\] (64)

If \( z = 0_{n \times 1} \), combining (63) and (64), we have

\[
Bs = 0_{n \times 1}.
\] (65)

Since \( B \) is linearly independent and with full rank, it follows that \( s = 0_{n \times 1} \). This contradicts our earliest assumption that \( z, s \neq 0_{(n_e+n_x) \times 1} \). Consequently, we must have \( \alpha > 0 \). Thus, each eigenvector of \( G \) is strictly positive and hence \( G \) is positive definite.

**REFERENCES**

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[2] R. A. Horn, and C. A. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[3] B. Bhanu, C. V. Ravishankar, A. K. Roy-Chowdury, H. Aghajan, and D. Terzopoulos, *Distributed Video Sensor Networks*. London, U.K.: Springer-Verlag, 2011.

[4] S. Soro and W. Heinzelman, “A survey of visual sensor networks,” *Adv. Multimedia*, vol. 2009, Jul. 2009, Art. no. 640386.

[5] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[6] S. Boyd, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[7] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[8] D. P. Bertsekas, and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.

[9] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY, USA: Springer-Verlag, 2002.

[10] J. P. Cunningham and Z. Ghahramani, “Linear dimensionality reduction: Survey, insights, and generalizations,” *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 2859–2900, 2015.

[11] X. Jiang, “Linear subspace learning-based dimensionality reduction,” *IEEE Signal Proc. Mag.*, vol. 28, no. 2, pp. 16–26, Mar. 2011.

[12] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis,” *J. ACM*, vol. 58, no. 3, pp. 1–37, May 2011.

[13] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, “Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization,” in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 2080–2088.

[14] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, “Multilinear discriminant analysis for face recognition,” *IEEE Trans. Image Proc.*, vol. 16, no. 1, pp. 212–220, Jan. 2007.

[15] D. Xu, S. Yan, L. Zhang, S. Lin, H.-J. Zhang, and T. S. Huang, “Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 36–47, Jan. 2008.

[16] X. Li, S. Lin, S. Yan, and D. Xu, “Discriminant locally linear embedding with high-order tensor data,” *IEEE Trans. Syst., Man, Cybern., B (Cybern.),* vol. 38, no. 2, pp. 342–352, Apr. 2008.

[17] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, “Graph embedding and extensions: A general framework for dimensionality reduction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[18] H.-L. Wu, J.-F. Nie, Y.-J. Yu, and R.-Q. Yu, “Multi-way chemometric methodologies and applications: A central summary of our research work,” *Anal. Chim. Acta*, vol. 650, no. 1, pp. 131–142, Sep. 2009.

[19] D. Letexier, S. Bourennane, and J. Blanc-Talon, “Nonorthogonal tensor matricization for hyperspectral image filtering,” *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 1, pp. 3–7, Jan. 2008.

[20] T.-K. Kim and R. Cipolla, “Canonical correlation analysis of video volume tensors for action categorization and detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 8, pp. 1415–1428, Aug. 2009.

[21] D. Tao, X. Li, X. Wu, and S. J. Maybank, “General tensor discriminant analysis and Gabor features for gait recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

[22] H. Fronthaler, K. Kollreider, J. Bigun, J. Ferrer, F. Alonso-Fernandez, J. Ortega-Garcia, and J. Gonzalez-Rodriguez, “Fingerprint image-quality estimation and its application to multithreaded verification,” *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 2, pp. 331–338, Jun. 2008.
[23] Y. Zhang, M. Chen, S. Mao, L. Hu, and V. Leung, “CAP: Community activity prediction based on big data analysis,” *IEEE Netw.*, vol. 28, no. 4, pp. 52–57, Jul. 2014.

[24] K. Maruhashi, F. Guo, and C. Faloutsos, “MultiAspectForensics: Mining large heterogeneous networks using tensor,” *Int. J. Web Eng. Technol.*, vol. 7, no. 4, p. 302, 2012.

[25] Z. Meng, A. Wiesel, and A. O. Hero, “Distributed principal component analysis on networks via directed graphical models,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.* (ICASSP), Mar. 2012, pp. 2877–2880.

[26] A. Aduroja, I. D. Schizas, and V. Maroulas, “Distributed principal components analysis in sensor networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 5850–5854.

[27] F. Jerome, D. Picard, and P. H. Gosselin, “Dimensionality reduction in decentralized networks by Gossip aggregation of principal components analyzers,” in *Proc. ESANN*, 2014, pp. 171–176.

[28] D. Gabay, “Applications of the method of multipliers to variational inequalities,” in *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*. Amsterdam, The Netherlands: North Holland, 1983.

[29] J. Eckstein and D. P. Bertsekas, “On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators,” *Math. Program.*, vol. 55, pp. 293–318, Apr. 1992.

[30] S. Zhang and A. G. Constantinides, “Lagrange programming neural networks,” in *Proc. IEEE Int. Conf. Acoust.*, Speech Signal Process., Mar. 2012, pp. 441–452, Jul. 1992.

[31] C. S. Leung, J. Sum, H. C. So, A. G. Constantinides, and F. K. W. Chan, “Lagrange programming neural networks for time-of-arrival-based source localization,” *Neural Comput. Appl.*, vol. 12, pp. 109–116, Jan. 2004.

[32] Y. Wang, J. Yang, W. Yin, and Y. Zhang, “A new alternating minimization algorithm for total variation image reconstruction,” *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 248–272, Jan. 2008.

[33] T. Erseghe, “A distributed and maximum-likelihood sensor network localization algorithm based upon a nonconvex problem formulation,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 4, pp. 247–258, Dec. 2015.

[34] J. Liang, M. Zhang, X. Zeng, and G. Yu, “Distributed dictionary learning for sparse representation in sensor networks,” *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2528–2541, Jun. 2014.

[35] J. Liang, G. Yu, B. Chen, and M. Zhao, “Decentralized dimensionality reduction for distributed tensor data across sensor networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2174–2186, Nov. 2016.

[36] J. Liang, H. C. So, C. S. Leung, J. Li, and A. Farina, “Waveform design with unit modulus and spectral shape constraints via Lagrange programming neural network,” *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 8, pp. 1377–1386, Dec. 2015.

[37] Handwritten Digit Database. [Online]. Available: http://www- i6.informatik.rwth-aachen.de/keysers/~usps.html

[38] Olivetti & Oracle Research Laboratory. (1994). *The Olivetti & Oracle Research Laboratory Face Database of Faces*. [Online]. Available: http://www.cam-olr.co.uk/facedatabase.html

ZHONGHUA YE was born in China. He is currently pursuing the Ph.D. degree with the School of Automation and Information Engineering, Xi’an University of Technology, Xi’an, China. He is currently a Lecturer with the Xi’an University of Finance and Economics.

His research interests include image processing, pattern recognition, and their applications.

HONG ZHU received the Ph.D. degree from Fukui University, Fukui, Japan, in 1999. She is currently a Professor with the Institute of Automation and Information Engineering, Xi’an University of Technology, Xi’an, China. Her research interests include image analysis, intelligent video surveillance, and pattern recognition.