IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 7, 2019, accepted September 23, 2019, date of publication September 30, 2019,
date of current version November 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944426

Low-Rank Tensor Thresholding Ridge Regression

KAILING GUO, (Member, IEEE), TONG ZHANG, (Member, IEEE), XIANGMIN XU, (Senior Member, IEEE), AND XIAOFEN XING
South China University of Technology, Guangzhou 510640, China

Corresponding author: Xiangmin Xu (xmxu@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U1801262, Grant U1636218, Grant 61702192, and Grant 61802131, in part by the China Postdoctoral Science Foundation under Grant 2018M630944, in part by the Natural Science Foundation of Guangdong Province, China, under Grant 2018A030313474, and in part by the Fundamental Research Funds for the Central Universities, under Grant 2018MS79, Grant 2019PY21, and Grant 2019MS028.

ABSTRACT In the area of subspace clustering, methods combining self-representation and spectral clustering are predominant in recent years. For dealing with tensor data, most existing methods vectorize them into vectors and lose most of the spatial information. For removing noise of the data, most existing methods focus on the input space and lack consideration of the projection space. Aiming at preserving the spatial information of tensor data, we incorporate tensor mode-d product with low-rank matrices for self-representation. At the same time, we remove noise of the data in both the input space and the projection space, and obtain a robust affinity matrix for spectral clustering. Extensive experiments on several popular subspace clustering datasets show that the proposed method outperforms both traditional subspace clustering methods and recent state-of-the-art deep learning methods.

INDEX TERMS Tensor, low-rank, subspace clustering.

I. INTRODUCTION

Subspace clustering is an important clustering learning problem and attracts increasing interests in recent years. It segments data by fitting each group of points to the subspace they are drawn from. Spectral-based subspace clustering is the most popular subspace clustering method in recent years. Encouraged by the promising performance of two seminal works, sparse subspace clustering (SSC) [2] and low-rank representation (LRR) [3], methods combining self-representation and spectral clustering have become very popular. A lot of variants of SSC and LRR have been proposed, e.g., least square regression (LSR) [4], non-negative low-rank and sparse (NNLRS) graph [5], structured sparse subspace clustering (S^3C) [6], and low-rank coding-based balanced (LRCB) graph [7]. Under some data structure and noise assumptions, they use the data matrix itself as the dictionary for representing data and the representation coefficients can be used to construct affinity matrix for spectral clustering. Although being very successful, the issues of data structure and noise need further consideration.

Subspace clustering has lots of applications on 2-dimensional (2D) data in computer vision. Traditionally, a common approach for dealing with 2D data is to vectorize them [2], [3]. Obviously, it is not a good strategy since it destroys the spatial structure of the 2D data. Peng *et al.* [8] took hand-designed image features (e.g., SIFT [9] and HOG [10]) as input of deep fully connected auto-encoder for sparse self-representation. Deep convolutional auto-encoder networks with sparse self-representation layers have been developed for subspace clustering and have achieved unprecedented results [11]–[14]. However, training deep neural networks is time-consuming. In order to retain the spatial information of 2D data, variance regularized ridge regression (VR3) [15] seeks two projection matrices of each direction of 2D data to obtain new representation of the original data. Nevertheless, VR3 vectorizes the projected data into vectors for the final representation and still harms the spatial structure of the original data. Since tensors are usually used to exploit spatial structure of the data, several tensor low-rank representation methods [19]–[21] have been proposed. However, they handle data vectors with spatial [19], [20] or multi-view [21] relationship rather than 2D data. Besides, they all use nuclear norm for modeling low-rankness, which needs large amounts of computation for optimization.

Most existing methods remove noise from the input space by formulating error regularization term in their objective functions. Thus, they can only handle specific noise induced

The associate editor coordinating the review of this manuscript and approving it for publication was Chong Leong Gan.

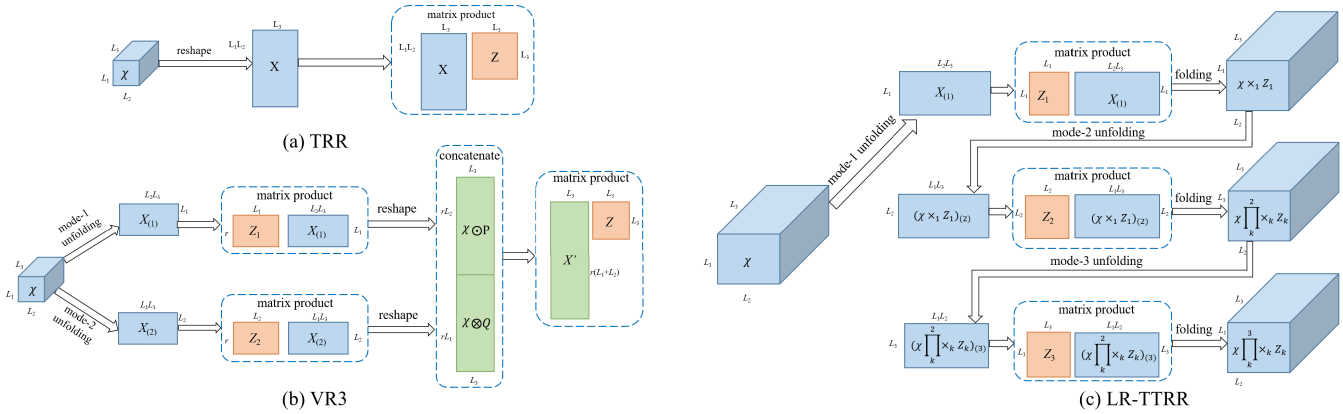

FIGURE 1. Comparison of the insights of TRR, VR3, and LR-TTRR.

by the error regularization term, e.g., Frobenius norm for Gaussian noise [4], combination of Frobenius and $\ell 1$ norm for mixture of Gaussian noise and sparse error [2], and $\ell_{2,1}$ norm for sample-wise corruptions [3]. To alleviate the effect of noise, some recent works introduce more robust loss functions, e.g., correntropy induced metric [22], Cauchy loss function [23], mixture of Gaussian regression [24], which reduce the dependence on the distribution of the noise. From another insight, thresholding ridge regression (TRR) [25], [26] eliminates the noise from the projection space and show that it does not require specific noise assumptions for several common linear projection space. Even though an error regularization term does not exactly fit the real noise distribution, it can always remove the noise to some extent. Thus, we combine removing noise in input space and projection space.

TRR has shown that simple ridge regression can lead to good subspace clustering performance with proper post-processing. To exploit spatial structure of the data, we generalize TRR to low-rank tensor thresholding ridge regression (LR-TTRR) by incorporating tensor mode-d product with low-rank matrices. A dataset of images is represented by a third-order tensor, of which each frontal slice is an image. Mode-d product with low-rank matrices is applied to each mode of the tensor for tensor self-representation. Here we constrain the mode-d product factor matrix to be low-rank since it captures global structure of the data [3]. Figure 1 gives a comparison of traditional TRR, VR3, and LR-TTRR to show this idea more clearly. The blue cubes and rectangles denote data or feature in tensor and matrix form, respectively, and the yellow rectangles denote the representation coefficients. The green ones are projection matrices, and the operators $\odot$ and $\otimes$ are defined as in [15] and can be found in Section II. Compared with the others, the proposed method takes the tensor data as an entire entity and learns richer spatial information than the others. We factorize the low-rank matrices into product of small matrices to accelerate optimization. The noise in the input space is modeled by the mixture of Gaussian noise and sparse error. In the projection space, we remove the noise with hard thresholding following TRR and further utilize power strategy to enlarge the difference between connection and disconnection in the affinity matrix.

The main contributions of LR-TTRR lies in the following: 1) Since tensor mode-d product is introduced for tensor self-representation, the spatial information of the original data is retained. 2) Low-rankness is employed to catch the global structure of the data and to accelerate the optimization procedure. 3) Noise removing is taken in both the input space and the projection space, and a modified hard thresholding post-processing strategy is proposed, which leads to a cleaner affinity for spectral clustering.

II. RELATED WORK

In this paper, we review the relevant prior work of this paper. From the aspect of exploiting data structure, we divide the related subspace clustering methods into vector space methods, multilinear methods, and deep learning methods. We also review noise removal in subspace clustering and acceleration of low-rank methods.

A. VECTOR SPACE METHODS

In traditional spectral-based subspace clustering methods, each sample is reshape into a long vector for self-representation, i.e., using all the data as dictionary to encode each sample. The encoded coefficient matrix can be used to construct the affinity matrix since each entry reflects the similarity of two samples. A lot of methods add different types of regularization terms or structure constraints to approach the assumption that each sample is only represented by data in the same subspace. SSC [2] utilize ℓ_1 norm to find sparse coefficients. LRR [3] adopt nuclear norm, which focus more on global structure of the data. To convert the coefficients to weights directly, non-negativity is introduced in NNLRS [5]. Nonnegative sparse Laplacian regularized low-rank representation (NSLLRR) [27] incorporates Laplacian regularization to encode the manifold structure of the data. In [30],

Sui *et al.* leveraged cascade low-rank and spare representation to achieve a more robust description of the clustering structure. For dealing with non-linear structure, Kernel trick [22], [31], [32] is also employed for sparse and low-rank self-representation to explore the intrinsic high-dimensional structure.

B. MULTINEAR METHODS

In traditional spectral-based subspace clustering methods, spatial structure of 2D data is destroyed because of vectorization. To retain spatial information, VR3 [15] defines two projection operator for a third-order tensor $\mathcal{X} \in \mathbb{R}^{L_1 \times L_2 \times L_3}$ as below.

$$\mathcal{X} \odot P := [\text{vec}(X_1 P), \cdots, \text{vec}(X_{L_3} P)]$$
$$\mathcal{X} \otimes Q := [\text{vec}(X_1^T Q), \cdots, \text{vec}(X_{L_3}^T Q)],$$

where $\text{vec}(\cdot)$ denotes vectorization operator. Ridge regression of the concatenation of $\mathcal{X} \odot P$ and $\mathcal{X} \otimes Q$ with orthogonal constraints on P and Q is adopt to seek the self-representation coefficients. Though it enhances the capability of retaining spatial information, the vectorization operation still damages the spatial structure. Tensor low-rank representation (TLRR) [19] its combination with sparse coding (TLRRSC) [20] represent the data as tensors to keep spatial structure for hyperspectral image clustering. The data samples to be clustered in hyperspectral image are still vectors, i.e., the pixels with multiple channels. The spatial structure in TLRR and TLRRSC refers to spatial relationship among vector samples, which is totally different from the proposed method. In LR-TTRR, each sample is a 2D tensor and the spatial relationship is among the elements within one data sample. Low-rank tensor constrained multi-view subspace clustering (LT-MSC) [21] incorporate spatial information by using hand-designed image features as data samples. LT-MSC applies matrix self representation on each view of the data and forms all the representation coefficients as a low-rank tensor. In contrast, the data for clustering in our method are in tensor form and the representation coefficients are represented by low-rank matrices corresponding to the modes of the tensor.

C. DEEP LEARNING METHODS

Deep subspace clustering with sparsity prior [8] utilize representation coefficients of SSC to guide the self-representation training of the middle layer of fully connected auto-encoder. Deep subspace clustering networks (DSC-Net) [11] design a new self-representation layer to convolutional auto-encoder to directly learn the affinities. Deep cognitive subspace clustering (DeepCogSC) [12] proposes a more robust deep subspace clustering framework by combining self-pace learning with DSC-Net to learn the samples from easy to hard. Distribution Preserving Subspace Clustering (DPSC) [13] introduces a distribution consistency loss to guide the learning of latent representation in DSC-Net. Self-Supervised Convolutional Subspace Clustering Network (S^2ConvSCN) [14] introduces a dual self-supervision that achieves simultaneous feature learning and subspace clustering. The above deep learning methods achieve state-of-the-art subspace clustering performance, but training deep networks is very time-consuming. Compared to deep learning methods, LR-TTRR is quite that represents the data in tensor form without exploiting complex features, but we outperform the deep learning methods.

D. REMOVING NOISE

Adding regularization term in self-representation model is a common approach for removing noise [3], [4], [22]–[24]. However, they only work when the noise distribution satisfies the corresponding assumption. To alleviate the effect of noise, [35] and [36] turn to pre-processing. They first remove noise via low-rank approximation methods [36], [37] and then apply self-representation models [3], [38] on the data matrix after de-noising. Some methods [3], [25], [39] removes noise in the projection subspace, i.e., post-pressing on the representation coefficients. LRR [3] first weights the left singular vector of representation coefficients matrix by multiplying the square root of singular values, and then normalizes each row of the weighted singular vector, and finally applies element-wise powering. However, the weighting and normalization operations do not have clear theoretical explanation. TRR [25] proves that the coefficients over intra-subspace data points are larger than those over inter-subspace data points under several linear projections, and thus keeps the given number of top values of each column in the coefficient matrix and zeros out the others to remove noise in the projection space. However, hard-thresholding may still keep inter-subspace connection. Finding good neighbors in the subspace clustering (FGNSC) [39] finds key connections among samples within a subspace and eliminates other entries to avoid inappropriate connection. However, such post-processing is more complex than TRR. LR-TTRR modifies the hard-thresholding procedure in TRR by introducing element-wise powering, leads to simple but effective post-precessing strategy.

E. ACCELERATION OF LOW-RANK METHODS

Low-rankness is usually modeled by minimizing nuclear norm [3], [37]. However, optimizing nuclear norm is time consuming since it typically requires singular value decomposition (SVD) at each iteration. In order to avoid optimizing nuclear norm, Go Decomposition (GoDec) [40] and its variant [36] model low-rankness by factorizing the low-rank matrix into product of small matrices. In order to accelerate LRR, fixed-rank representation (FRR) [42] introduces an auxiliary self-representation coefficient matrix and forces it to be as close as possible to a low-rank matrix induced by matrix factorization. Since auxiliary variable may introduce extra computation, we factorize the self-representation coefficient matrices directly of the proposed tensor method following GoDec [40].

III. TENSOR THRESHOLDING RIDGE REGRESSION

In this section, we first give some notations and preliminaries, and then introduce LR-TTRR, its optimization and convergence analysis. Finally, we show how to apply LR-TTRR in subspace clustering.

A. NOTATIONS AND PRELIMINARIES

In this section, we introduce some notations and basic definitions relevant to this paper [43], [44].

A tensor is a multi-dimensional array of numbers denoted by a calligraphy letter, e.g., $\mathcal{X}$. We use upper case letters for matrices, e.g., X; bold lower case letters for vectors, e.g., $\mathbf{x}$; lower case letters for scalars, e.g., x; and letters with subscripts for the corresponding entries, e.g., $\mathcal{X}_{l_1 l_2 \cdots l_N}$ for entries in tensors and $X_{l_1 l_2}$ for entries in matrices. We denote the variables in a sequence by bracketed superscripts, e.g., $X^{(k)}$. The vectorization function vec($\mathcal{X}$) refers to transforming a tensor into a vector by concatenating all its elements in the natural order.

Definition 1 (Mode-d Matrix Unfolding [44]): The mode-d matrix unfolding of a Nth-order tensor $\mathcal{X} \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_N}$ is the set of vectors in $\mathbb{R}^{L_d}$ obtained by keeping the index i_d fixed while varying the other indices. Therefore, the mode-d matrix unfolding of a Nth-order tensor is a matrix $X_{(d)} \in \mathbb{R}^{L_d \times \bar{L}_d}$, where $\bar{L}_d = \prod_{i \neq d} L_i$.

Definition 2 (Mode-d Product [43], [44]): The mode-d product of a tensor $\mathcal{X} \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_N}$ with a matrix $Z \in \mathbb{R}^{L'_d \times L_d}$ is an $L_1 \times L_2 \times \cdots \times L_{d-1} \times L'_d \times L_{d+1} \times \cdots \times L_N$ tensor defined by

$$(\mathcal{X} \times_d Z)_{l_1 l_2 \cdots l_{d-1} l'_d l_{d+1} \cdots l_N} = \sum_{l_d} (\mathcal{X})_{l_1 l_2 \cdots l_{d-1} l_d l_{d+1} \cdots l_N} Z_{l'_d l_d},$$

for all indexed values. It can be expressed as matrix multiplication in terms of unfolded tensors: $\mathcal{Y} = \mathcal{X} \times_d Z \Leftrightarrow Y_{(d)} = ZX_{(d)}$.

Based on the definition, mode-d product is commutative and we denote

$$\mathcal{X} \times_1 Z_1 \times_2 Z_2 \cdots \times_N Z_N \triangleq \mathcal{X} \prod_{k=1}^{N} \times_k Z_k,$$

and

$$\mathcal{X} \times_1 Z_1 \times_2 Z_2 \times \cdots \times_{i-1} Z_{i-1} \times_{i+1} Z_{i+1} \times \cdots \times_N Z_N = \mathcal{X} \prod_{k=1; k \neq i}^{N} \times_k Z_k \triangleq \mathcal{X} \bar{\times}_i Z_i.$$

Definition 3 (Frobenius Norm [44]): The Frobenius norm of a tensor $\mathcal{X} \in R^{L_1 \times L_2 \times \cdots \times L_N}$ is given by $\|\mathcal{X}\|_F = \sqrt{\sum_{l_1=1}^{L_1} \cdots \sum_{l_N=1}^{L_N} \mathcal{X}^2_{l_1 \cdots l_N}}$.

B. THE PROPOSED MODEL

Given a data matrix X, most existing self-representation methods including can be formulated as

$$\min_{Z,S} \frac{1}{2}\|X - XZ - S\|_p + \lambda \Omega_1(S) + \beta \Omega_2(Z), \tag{1}$$

where $\|\cdot\|_p$ is some kind of similarity measurement, $\Omega_1(S)$ is an error regularization term, Z is the self-representation, and $\Omega_2(Z)$ denotes a prior structured regularization on Z. Here each column of X denotes a sample. When the sample is represented by tensor, e.g., 2D tensor for image, these methods have to reshape the tensor into a vector and thus the spatial information of the original data is lost.

In order to reserve the spatial information, we propose a tensor form self-representation method. Note that tensor mode-d product can be computed by matrix multiplication after unfolding the tensor. We use tensor mode-d product for tensor self-representation to replace matrix product in matrix-based self-representation. Since LRR shows that low-rank representation coefficients are useful for capturing global structure of the data, we also constrain the representation coefficients to be low-rank. Motivated by GoDec, we adopt fixed-rank constraint instead nuclear norm. For dealing with noise in the input space, we model the noise as "Gaussian noise + sparse error" following GoDec. Since it is difficult for GoDec to tune the cardinality of the sparse component for optimal result, ℓ_1 norm is employed for regularizing of the sparse error as in semi-soft GoDec (SSGoDec) [45]. The proposed LR-TTRR is given as follows

$$\min_{Z_i, \mathcal{S}} \frac{1}{2}\|\mathcal{X} - \mathcal{X} \prod_{i=1}^{N} \times_i Z_i - \mathcal{S}\|_F^2 + \lambda \|\mathcal{S}\|_1 + \frac{1}{2} \sum_{i}^{N} \beta_i \|Z_i\|_F^2,$$
$$s.t.\ \mathrm{rank}(Z_i) \leq r_i, \quad i = 1, 2, , \cdots, N, \tag{2}$$

where $\|\mathcal{S}\|_1$ is the sum of absolute values of all entries in $\mathcal{S}$, Z_i's are the low-rank representation coefficients for each mode of the tensor, and λ and β_i are trade-off parameters. The third term in the objective function adopts Frobenius norm regularization for stable solution. It is a quite simple strategy but TRR has shown that it is very effective for subspace clustering when combined with coefficient thresholding strategy.

To tackle the rank function, we follow GoDec and FRR and factorize Z_i as $Z_i = U_i V_i$, where $U_i \in \mathbb{R}^{L_i \times r_i}$ and $V_i \in \mathbb{R}^{r_i \times \prod_{k \neq i} L_k}$. The low-rank constraint is implicit satisfied since $\mathrm{rank}(U_i V_i) \leq \min(\mathrm{rank}(U_i), \mathrm{rank}(V_i)) \leq r$. The proposed model changes into

$$\min_{U_i, V_i, \mathcal{S}} \frac{1}{2}\|\mathcal{X} - \mathcal{X} \prod_{i=1}^{N} \times_i U_i V_i - \mathcal{S}\|_F^2 + \lambda \|\mathcal{S}\|_1 + \frac{1}{2} \sum_{i}^{N} \beta_i \|U_i V_i\|_F^2. \tag{3}$$

The U_i's and V_i's in Eq. (3) can be solved by QR decomposition, which we will clarify in Subsection III-C.

C. OPTIMIZATION

Since the proposed model is non-convex, it is difficult to optimize all the variables simultaneously. We solve the problem by alternately optimizing $\mathcal{S}$, U_i's and V_i's.

When U_i's and V_i's are given, $\mathcal{S}$ can be optimized via soft-thresholding (shrinkage) [48]. The soft-thresholding operator

is defined as $\mathbb{S}_\lambda(x) = \text{sgn}(x)\max(|x| - \lambda, 0)$, where $\text{sgn}(\cdot)$ is the sign function. This operator can be extended to tensors by applying it element-wise. Then, the solution $\mathcal{S}$ is given by

$$\mathcal{S} = \mathbb{S}_\lambda(\mathcal{X} - \mathcal{X}\prod_{i=1}^{N} \times_i U_i V_i) \tag{4}$$

Given $\mathcal{S}$, we unfold the tensors in i^{th} mode for optimizing U_i's and V_i's. Then, we have the sub-problem of U_i and V_i.

$$\min_{U_i,V_i,\mathcal{S}} \frac{1}{2}\|X_{(i)} - U_i V_i(\mathcal{X}\bar{\times}_i(U_i V_i))_{(i)} - S_{(i)}\|_F^2 + \frac{1}{2}\sum_i^N \beta_i \|U_i V_i\|_F^2. \tag{5}$$

For notation simplification, we define $A_i = X_{(i)} - S_{(i)}$ and $B_i = (\mathcal{X}\bar{\times}_i(U_i V_i))_{(i)}$. Correspondingly, the sub-problem (5) changes into the following form:

$$\min_{U_i,V_i} \|A_i - U_i V_i B_i\|_F^2 + \beta_i \|U_i V_i\|_F^2. \tag{6}$$

Alternately optimizing U_i and V_i in (6) yields the following updating rules for the t^{th} iteration:

$$\begin{cases} U_i = A_i B_i^T V_i^{(t-1)T}[V_i^{(t-1)}(B_i B_i^T + \beta_i I)V_i^{(t-1)T}]^{-1}, \\ V_i = (U_i^T U_i)^\dagger U_i^T A_i B_i^T (B_i B_i^T + \beta_i I)^{-1}, \end{cases} \tag{7}$$

where I is an identity matrix, and $(\cdot)^\dagger$ stands for the Moore-Penrose pseudo-inverse. Note that we ignore the bracketed superscripts for the t^{th} iteration in order to simplify the notations. It follows that

$$\begin{aligned} U_i V_i &= U_i (U_i^T U_i)^\dagger U_i^T A_i B_i^T (B_i B_i^T + \beta_i I)^{-1} \\ &= P_{U_i}(A_i B_i^T (B_i B_i^T + \beta_i I)^{-1}), \end{aligned} \tag{8}$$

where P_{U_i} is the orthogonal projection onto the column space of U_i. Since the objective value in (6) is determined by the matrix product $U_i V_i$ rather than individual U_i or V_i, it is a considerable strategy to find a pair of (U_i, V_i) that has the same product $U_i V_i$ but with less computation. From (7), we know that the column space of U_i is the same as the column space of $A_i B_i^T V_i^{(t-1)T}$. We can compute its orthogonal basis via QR decomposition, i.e., $A_i B_i^T V_i^{(t-1)T} = Q_i R_i$. Then, the product $U_i V_i$ can be equivalently computed as $U_i V_i = P_{Q_i}(A_i B_i^T (B_i B_i^T + \beta_i I)^{-1}) = Q_i Q_i^T A_i B_i^T (B_i B_i^T + \beta_i I)^{-1}$. Thus, a faster updating procedure is given as

$$\begin{cases} A_i B_i^T (V_i^{(t-1)})^T = Q_i R_i, \quad U_i = Q_i \\ V_i = Q_i^T A_i B_i^T (B_i B_i^T + \beta_i I)^{-1}. \end{cases} \tag{9}$$

Although this strategy accelerates the computation, computing the matrix inversion in (9) may still take time. Motivated by LRR [3], the computational cost can be further reduced. We then have the following theorem.

Theorem 1: For any optimal solution (U_i^*, V_i^*) to problem (6), we have $(V_i^*)^T \in span(B_i)$.

Proof: According to Eq. (7), we have

$$(V_i^*)^T = (B_i B_i^T + \beta_i I)^{-1} B_i A_i^T ((U_i^*)^T U_i^*)^\dagger. \tag{10}$$

Thus, $(V_i^*)^T \in \text{span}((B_i B_i^T + \beta_i I)^{-1} B_i)$. Suppose the skinny SVD of B_i is $U_B \Sigma_B V_B^T$ and $U_{B\perp}$ is the orthogonal complement of U_B; then we have

$$\begin{aligned} &(B_i B_i^T + \beta_i I)^{-1} B_i \\ &= [U_B, U_{B\perp}]\text{bd}((\Sigma_B^2 + \beta_i I)^{-1}, \beta_i^{-1} I)[U_B, U_{B\perp}]^T U_B \Sigma_B V_B^T \\ &= [U_B, U_{B\perp}]\text{bd}((\Sigma_B^2 + \beta_i I)^{-1}, \beta_i^{-1} I)[I, \mathbf{0}]^T \Sigma_B V_B^T \\ &= U_B(\Sigma_B^2 + \beta_i I)^{-1}\Sigma_B V_B^T, \end{aligned}$$

where $\text{bd}(\cdot)$ produces the block diagonal concatenation matrix of its input arguments, and $\mathbf{0}$ is a zero matrix. Since $\text{span}(U_B) = \text{span}(B_i)$, we have $(V_i^*)^T \in \text{span}(B_i)$. □

According to Theorem 1, V_i can be factorized into $V_i = \widetilde{V}_i Q_{B_i}^T$, where Q_{B_i} is computed from the QR decomposition $B_i = Q_{B_i} R_{B_i}$. Define $\widetilde{B}_i = Q_{B_i}^T B_i$, after which problem (6) can be transformed into the following simpler problem:

$$\min_{U_i,\widetilde{V}_i} \|A_i - U_i \widetilde{V}_i \widetilde{B}_i\|_F^2 + \beta_i \|U_i \widetilde{V}_i\|_F^2. \tag{11}$$

This problem is equivalent to problem (6) because $\|U_i \widetilde{V}_i Q_{B_i}^T\|_F = \|U_i \widetilde{V}_i\|_F$ when Q_{B_i} is column orthogonal. The final algorithm for solving (2) is summarized in Algorithm 1. Since B_i is usually low-rank in practice, this strategy can accelerate the computation.

Algorithm 1 LR-TTRR

Input: $\mathcal{X}, r_i, \lambda, \beta_i$

Output: $\mathcal{S}, Z_i$

1: Initialize $\mathcal{S}^{(0)} = \mathbf{0}, Z_i^{(0)} = I, t := 1$.
2: Generate standard Gaussian matrices $V_i^{(0)} \in \mathbb{R}^{r_i \times L_i}, \forall i = 1, 2, \cdots, N$;
3: **while** not converged **do**
4: $\quad \mathcal{A}^{(t)} = \mathcal{X} - \mathcal{S}^{(t-1)}$;
5: $\quad$ **for** $i = 1$ to N **do**
6: $\quad\quad A_i^{(t)} = \mathcal{A}_{(i)}^{(t)}, B_i^{(t)} = (\mathcal{X}\bar{\times}_i Z_i^{(t-1)})_{(i)}$;
7: $\quad\quad$ Compute the QR decomposition $B_i^{(t)} = Q_{B_i}^{(t)} R_{B_i}^{(t)}$;
8: $\quad\quad \widetilde{V}_i^{(t-1)} = V_i^{(t-1)} Q_{B_i}^{(t)}, \widetilde{B}_i = Q_{B_i}^{(t)T} B_i^{(t)}$;
9: $\quad\quad$ Compute the QR decomposition $A_i^{(t)}\widetilde{B}_i^{(t)T}(\widetilde{V}_i^{(t-1)})^T = Q_i^{(t)} R_i^{(t)}$;
10: $\quad\quad U_i^{(t)} = Q_i^{(t)}$; $\widetilde{V}_i^{(t)} = Q_i^{(t)T} A_i^{(t)} \widetilde{B}_i^{(t)T} (\widetilde{B}_i^{(t)} \widetilde{B}_i^{(t)T} + \beta_i I)^{-1}$;
11: $\quad\quad V_i^{(t)} = \widetilde{V}_i^{(t)} Q_{B_i}^{(t)T}, Z_i^{(t)} = U_i^{(t)} V_i^{(t)}$;
12: $\quad$ **end for**
13: $\quad \mathcal{S}^{(t)} = \mathbb{S}_\lambda(\mathcal{X} - \mathcal{X}\prod_{i=1}^N \times_i Z_i^{(t)})$;
14: $\quad$ t := t + 1;
15: **end while**

The computational complexity of Algorithm 1 mainly comes from tensor mode-d product, QR decomposition, matrix multiplication, and matrix inversion in steps 6 to 13. The complexity of computing the tensor mode-d product $\mathcal{X} \times_d Z_d$ is $O(\prod_{i=1}^N L_i r_d)$ by replacing Z_d with small matrices $U_d V_d$. The computational complexity of QR decomposition of a $m \times n$ matrix with rank r is $O(mnr)$. The complexity of matrix multiplication of matrices with size $m \times n$ and

$n \times k$ is $O(mnk)$, and the complexity of matrix inversion of a $n \times n$ matrix is $O(n^3)$. Summing up the complexity of all the operations in the inner iteration, the computational complexity of steps 6 to 11 for iteration i is $O(\prod_{j=1}^{N} L_j r_{Bi} + r_{Bi}^3)$, where r_{Bi} is the rank of matrix B_i. Denote the maximum of all r_{Bi} as r, by summing up all the operation complexity from steps 6 to 13, we have the computational complexity of Algorithm 1 $O(\prod_{j=1}^{N} L_j(\sum_i^N r_i + r) + r^3)$. Note that r_{Bi} is at most $\min(L_i, \prod_{j\neq i} L_j)$. The value r is some L_i or $\prod_{j\neq i} L_j$. Suppose the N-th way of the tensor corresponds to the number of samples and the sample number L_N is large enough, i.e., $L_N \geq \prod_{i=1}^{N-1} L_i$, then r is L_N and the computational complexity of Algorithm 1 is $O(\prod_{j=1}^{N} L_j L_N + L_N^3)$, which is at the same order as LRR [3].

D. CONVERGENCE ANALYSIS

Based on some existing theorems from [49] and [50], we show that Algorithm 1 converges and give an estimation of the convergence rate. We have the following theorems.

Theorem 2: Alternately optimizing (3) via (4) and (7) yields a sequence $\{U_i^{(t)}, V_i^{(t)}, \mathcal{S}^{(t)} | i = 1, 2, \cdots, N\}$ *that converges to a local minimizer* $\{U_i^*, V_i^*, \mathcal{S}^* | i = 1, 2, \cdots, N\}$ *of (3). Denote* $\mathbf{u} = [vec(U_1^{(t)}); \cdots; vec(U_N^{(t)}); vec(V_1^{(t)}); \cdots; vec(V_N^{(t)}); vec(\mathcal{S}^{(t)})]$, *then we have* $\|\mathbf{u}^{(t)} - \bar{\mathbf{u}}\| \leq Ct^{-(1-\theta)/(2\theta-1)}$ *for all* $t > t_0$, *for cetrain* $t_0 > 0$, $C > 0$, *and* θ *is given by*

$$\theta = 1 - (d(3d-3)^{n-1})^{-1}. \tag{12}$$

We give a sketch of the proof here and the details can be found in the Appendix.

Step 1, we estimate the Lojasiewicz exponent in the Kurdyka-Lojasiewicz (KL) inequality of the objective function in (3). For a function $\psi(\mathbf{x})$, the corresponding KL inequality is

$$\phi'(|\psi(\mathbf{x}) - \psi(\bar{\mathbf{x}})|)\text{dist}(\mathbf{0}, \partial\psi(\mathbf{x})) \leq 1, \tag{13}$$

where $\phi'(\cdot)$ is the derivative of function $\phi(s) = cs^{1-\theta}$, and $\text{dist}(\mathbf{0}, \partial\psi(\mathbf{x}))$ is the distance of the subgradient$\partial\psi(\mathbf{x})$ to the original. Formal definition of KL inequality and KL property can be found in [50]. The parameter θ is called Lojasiewicz exponent. KL inequality is for vector function, but we can modify the objective function in (3) by vectorizing the variables. Theorem 4.2 in [49] provides an explicit estimation of θ for polynominal functions. The objective function in (3) is sum of polynominal function and absolute value functions. By studying the subgradient of absolute value function, we prove that the conclusion for polynominal functions still holds for the objective function in (3).

Step 2, we prove that the alternative optimization procedure converges to a critical point based on KL inequality by means of Theorem 2.8 in [50]and estimate the convergence rate based on the Lojasiewicz exponent by means of Theorem 2.9 in [50]. What we need to do is to verify that the objective function in (3) satisfies the assumptions of Theorem 2.8 and 2.9 in [50].

Theorem 3: Algorithm 1 generates a sequence $\{Z_i^{(t)}, \mathcal{S}^{(t)} | i = 1, 2, \cdots, N\}$ *that converges to* $\{Z_i^*, \mathcal{S}^* | i = 1, 2, \cdots, N\}$, *where* $Z_i^* = U_i^* V_i^*$ *and* $\{U_i^*, V_i^*, \mathcal{S}^* | i = 1, 2, \ldots, N\}$ *is a local minimizer of (3). The convergence rate of* $(vec(Z_i^{(t)}); \cdots; vec(Z_N^{(t)}); vec(\mathcal{S}^{(t)}))$ *is given by Eq. (12).*

Proof: We first show that updating U_i and V_i by (7) and (9) generates the same product $U_i V_i$ at each iteration. For clarity, we denote U_i and V_i updated by (9) as $\bar{U}_i$ and $\bar{V}_i$. For notation simplification, we ignore the superscript for the t^{th} iteration temporally. Define $K_i = A_i B_i^T (B_i B_i^T + \beta I)^{-1}$. Since Q_i is the basis of the column space of U_i, we have $U_i = Q_i D_i$, where D_i is an $r \times r$ full rank matrix. Therefore, $V_i = (U_i^T U_i)^\dagger U_i^T K_i = (D_i^T D_i)^\dagger D_i^T Q_i K_i$. According to (8), we have $U_i^{(t+1)} V_i^{(t+1)} = P_{U_i^{(t+1)}}(A_i^{(t+1)} B_i^{(t+1)T} (B_i^{(t+1)} B_i^{(t+1)T} + \beta I)^{-1})$. The column space of $U_i^{(t+1)}$ is the same as the column space of $A_i^{(t+1)} B_i^{(t+1)T} V_i^T$. From the solution of V_i, we have $A_i^{(t+1)} B_i^{(t+1)T} V_i^T = A_i^{(t+1)} B_i^{(t+1)T} K_i^T Q_i^T D_i (D_i^T D_i)^{\dagger T}$. Thus, the column space of $U_i^{(t+1)}$ is the same as the column space of $A_i^{(t+1)} B_i^{(t+1)T} K_i^T Q_i^T$. When the variables are updated by (9), we have $\bar{U}_i^{(t+1)} \bar{V}_i^{(t+1)} = P_{\bar{U}_i^{(t+1)}}(A_i^{(t+1)} B_i^{(t+1)T} (B_i^{(t+1)} B_i^{(t+1)T} + \beta I)^{-1})$. The column space of $\bar{U}_i^{(t+1)}$ is the same as the column space of $A_i^{(t+1)} B_i^{(t+1)T} \bar{V}_i^T = A_i^{(t+1)} B_i^{(t+1)T} K_i^T Q_i^T$. Thus, $\bar{U}_i^{(t+1)} \bar{V}_i^{(t+1)} = U_i^{(t+1)} V_i^{(t+1)}$. By Theorem 2, U_i and V_i converges, and thus their product $U_i V_i$ converges to $U_i^* V_i^*$. Therefore, $\bar{U}_i^{(t+1)} \bar{V}_i^{(t+1)}$ also converges to $U_i^* V_i^*$. By Theorem 1, the sequence $\{Z_i, \mathcal{S} | i = 1, 2, \cdots, N\}$ generated by Algorithm 1 converges to $\{U_i^* V_i^*, \mathcal{S}^* | i = 1, 2, \ldots, N\}$. Since

$$\begin{aligned}
&\|U_i V_i - U_i^* V_i^*\| \\
&\quad = \|U_i V_i - U_i V_i^* + U_i V_i^* - U_i^* V_i^*\| \\
&\quad = \|U_i (V_i - V_i^*) + (U_i - U_i^*) V_i^*\| \\
&\quad \leq \|U_i\| \|V_i - V_i^*\| + \|U_i - U_i^*\| \|V_i^*\| \\
&\quad \leq (\|U_i - U_i^*\| + \|U_i^*\|) \|V_i - V_i^*\| + \|U_i - U_i^*\| \|V_i^*\|,
\end{aligned}$$

there exists some constant $C > 0$ such that

$$\|U_i V_i - U_i^* V_i^*\| \leq C(\|V_i^- V_i^*\| + \|U_i - U_i^*\|).$$

Thus, the conclusion for the convergence rate is proved. □

IV. AFFINITY MATRIX CONSTRUCTION

Recently, self-representation is popularly used for constructing an affinity matrix in spectral clustering-based subspace clustering [2], [3], [27]. LRR computes a weighted shape iteration matrix as the affinity matrix. Suppose the representation coefficients are denoted by matrix Z whose skinny SVD is $Z = \hat{U}\hat{\Sigma}\hat{V}$. The affinity matrix A constructed by LRR is defined as $[G]_{ij} = [\widetilde{U}\widetilde{U}^T]_{ij}^{\phi}$, where $\widetilde{U}$ is formed by $\hat{U}(\hat{\Sigma})^{\frac{1}{2}}$ with normalized rows and ϕ is an even integer parameter that controls the sharpness of the affinity matrix. TRR proves that small values of the representation coefficients always corresponds to the projections over noise and applies hard thresholding on the coefficients to construct an affinity matrix.

The construction method in LRR is based on the property that the column space of Z is the recovery of the row space of the clean data. This property holds for LRR but may be not true for other methods. In contrast, hard thresholding strategy in TRR does not require extra properties of Z and thus is a more reasonable choice. However, TRR simply choses the k largest entries of each column of the coefficient matrix and zeros the others, which may remove useful information or may keep coefficients corresponding to noise. Note that the power strategy in LRR enlarges the relative rate between large values and small values of the coefficients, which can be considered as a soft denoising method. Thus, we can use large k in hard threshoding and combine it with power strategy. Based on these analyses, we propose to construct the affinity matrix from the representation coefficients Z with the following steps:

1) Obtain $\bar{Z}$ by normalizing the columns of Z with ℓ_∞ norm;

2) Compute $\bar{G} = |\bar{Z}| + |\bar{Z}|^T$;

3) Keep the k largest entries of each column of $\bar{G}$ and zero the others;

4) Obtain $\hat{G}$ by normalizing the columns of $\hat{G}$ with ℓ_∞ norm;

5) Compute $\widetilde{G} = |\hat{G}| + |\hat{G}|^T$;

6) Construct the affinity matrix G as $[G]_{ij} = [\widetilde{G}_{ij}]^2$.

For LR-TTRR, suppose the tensor $\mathcal{X}$ is formed by concatenating all the samples and the N-th way of the tensor corresponds to the number of samples. After obtaining Z_i's, the low-n-rank tensor is unfolded in the N^{th} mode and we have $(\mathcal{X}\prod_{i=1}^{N}\times_i Z_i)_{(N)} = Z_N(\mathcal{X}\prod_{i=1}^{N}\bar{\times}_i Z_i)_{(N)}$. After transposing, Z_N^T can be considered as the encoding coefficients of the data matrix $X_{(N)}$ with the dictionary, $[(\mathcal{X}\prod_{i=1}^{N}\bar{\times}_i Z_i)_{(N)}]^T$, which can be used as the coefficients for constructing affinity matrix, i.e., we set $Z = Z_{(N)}^T$. Compared to existing self-representation methods which use the data matrix itself as the dictionary, LR-TTRR constructs a dictionary that encodes the spatial information of the original data.

V. EXPERIMENTS

In this section, we conduct experiments on 2D image datasets for subspace clustering to demonstrate the superiority of the proposed LR-TTRR.

A. METHODS FOR COMPARISON

We demonstrate the effectiveness of LR-TTRR for subspace clustering by comparing with several typical or recent developed methods including SSC [2], S^{3}C [6], SSC by orthogonal matching pursuit (SSC-OMP) [51], kernel sparse subspace clustering (KSSC) [52], LSR [4], LRR [3], kernel low-rank representation [32], low-rank subspace clustering (LRSC) [53], FRR [42], NNLRS [54], nonnegative sparse Laplacian regularized low-rank representation (NSLLRR) [27], LRCB [7], GoDec+ [36], TRR [25], VR3 [15], nonlinear VR3 (NVR3) [15], DSC-Net [11], DeepCogSC [12], DPSC [13], S^{2}ConvSCN [14], structured autoencoders (StructAE) [55], nonlinear subspace clustering via adaptive graph regularized autoencoder (NSC-AGA) [56], FGNSC [39], and iterative reconstrained LRR with weighted nonconvex regularization (IRWNR) [57].

B. PARAMETER SETTINGS

The parameters of the compared methods are set as suggested in the original papers or tuned for optimal results. For LR-TTRR, we tune the parameters for optimal results in the following experiments. At first glance, LR-TTRR has too many parameters (β_i, r_i's, λ, and k) that need to be tuned, but we will show that most of the parameters can be jointly tunes. For solving Z_i, if we set $\mathcal{S}$ as zero tensor $\mathbf{0}$ and Z_j with $j \neq i$ as identity matrices, and ignore the low-rank constraints temporarily, problem (2) becomes a ridge regression problem $\min_{Z_i} \|X_{(i)} - X_{(i)}Z_i\|_F^2 + \beta_i\|Z_i\|_F^2$ with analytical solution given by $Z_i = (X_{(i)}X_{(i)}^T + \beta_i I)^{-1}X_{(i)}$. It is well known that the effect of β_i is to add the singular values of $X_{(i)}X_{(i)}^T$ to make the matrix inversion operation stable. This motivates us to set the value of β_i according to the singular values of $X_{(i)}X_{(i)}^T$. In this paper, we set as $\beta_i = \gamma_1\|X_{(i)}X_{(i)}^T\|_2$, where $\|\cdot\|_2$ is the matrix 2-norm, i.e., the largest singular value of the matrix, and γ_1 is a hyper parameter. Thus, we only need to tune one parameter γ_1 for the parameters β_i's. For the rank r_i, it has closed relationship to the dimension L_i of the corresponding mode. We set r_i as $r_i = \gamma_2 L_i$ for the modes that account for spatial relationship, where γ_2 is a hyper parameter. For the mode corresponding to sample numbers in subspace clustering, we set the rank as kC, where k is the thresholding parameter and C is the number of clusters.

C. FACE CLUSTERING

The extended Yale B [58] and ORL [59] face datasets are adopted for evaluating the subspace clustering performance.

The extended Yale B dataset [58] contains face images under nine poses and 64 illumination conditions for 38 individuals. We use the cropped data containing only the frontal face. Following the experimental setup of [36], we resize the images to 48 × 42 pixels. The first 10, 20, 30, and 38 subjects are chosen for the experiments. The parameters $(\gamma_1, \gamma_2, \beta, k)$ for LR-TTRR are set as $(2e-4, 0.7, 9e-3, 8)$. Table 1 shows the clustering errors of the compared methods on extended Yale B with different number of subjects. LR-TTRR achieves the best performance in all the cases except that it is a little worse than NVR3 and S2ConvSCN with 10 and 38 subjects, respectively. Though VR3 and NVR3 utilize spatial information to some extent, they only show their superiority over other methods with small number of subjects. The optimization of VR3 and NVR3 need to compute pairwise matrix multiplication of all the data samples and thus they become very time-consuming as the sample number increase. With all the 38 subjects, they will take more than 3 days. In contrast, LR-TTRR only takes around 700 seconds for all the 38 subjects. FGNSC focuses on post-preprocessing on self-representation coefficients and

TABLE 1. Clustering error (%) on extended Yale B.

Subjects	10	20	30	38
SSC	12.19	25.45	25.57	29.95
S^3C	8.28	12.52	14.41	21.96
SSC-OMP	-	-	-	16.05
KSSC	-	-	-	19.64
LSR	38.97	30.81	30.33	30.65
LRR	5.47	10.06	8.38	9.27
KLRR	-	-	-	27.36
LRSC	-	-	-	25.84
FRR	18.28	19.49	21.03	26.76
NNLRS	5.19	6.59	8.48	19.70
NSLLRR	3.92	6.39	5.51	8.41
LRCB	7.20	12.60	8.57	12.39
GoDec+	1.56	7.25	6.99	7.75
TRR	3.91	11.09	3.94	13.17
VR3	1.41	10.39	10.68	-
NVR3	**0.06**	9.30	6.56	-
DSC-Net	-	-	-	2.67
DeepCogSC	-	-	-	2.18
S^2ConvSCN	-	-	-	**1.52**
StructAE	-	-	-	5.3
NSC-AGA	-	-	-	2.16
FGNSC	-	-	-	5.76
LR-TTRR	1.25	**1.74**	**1.63**	1.99

TABLE 2. Computation time (seconds) on the first ten classes of extended Yale B for subspace clustering.

Methods	Time	Methods	Time	Methods	Time
SSC	55.25	NNLRS	103.16	VR3	620.94
S^3C	343.92	NSLLRR	101.79	NVR3	2682.71
LSR	0.07	LRCB	1185.55	LR-TTRR	52.97
LRR	26.85	GoDec+	44.83		
FRR	6.87	TRR	0.59		

shows great improvement. LR-TTRR utilizes a simpler post-processing strategy but still outperforms FGNSC. Recent state-of-the-art subspace clustering results are all achieved by deep learning methods, which outperform traditional methods significantly. LR-TTRR achieves amazing result that outperforms all deep learning methods except S2ConvSCN. Note that LR-TTRR does not need expensive training and it outperforms S2ConvSCN on other datasets in the following experiments. To show the efficiency of LR-TTRR, we also give a computational time comparison in Table 2 for the first ten classes. LR-TTRR is relatively fast among all the compared methods. Though the performance of NVR3 is the best for the first ten classes, it is the slowest method.

The ORL face dataset [59] contains face images of 40 subjects with 10 samples per subject. Following [11], each image is down-sampled from 112 × 92 to 32 × 32. It is more challenging than extended Yale B since the images were taken under varying lighting conditions with different facial expressions and details. LR-TTRR's parameters $(\gamma_1, \gamma_2, \beta, k)$ are set to $(1e-5, 0.8, 1e-4, 5)$. Figure 2 illustrates the clustering errors of the compared methods. LR-TTRR outperforms all traditional methods and deep learning methods, and gains 0.75% performance improvement over the second best method. This again verifies that spatial information is very useful for image subspace clustering and the noise removing strategy is very robust.

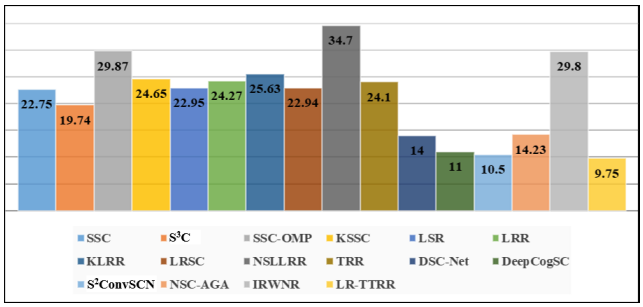

FIGURE 2. Clustering error (%) on ORL with 40 subjects.

D. OBJECT CLUSTERING

The COIL20 and COIL100 datasets [60] are both object image datasets. They contain 1440 images from 20 classes and 7200 images from 100 classes, respectively. Each object was taken with varying pose degree of 5, resulting in 72 images per object. In contrast to well aligned face images, these datasets are more challenging due to view changing. We downsample the images to 32 × 32. We set the parameters $(\gamma_1, \gamma_2, \beta, k)$ to (9e-3, 0.3, 0.02, 5) and (5e-3, 0.3, 0.05, 5) for COIL20 and COIL100, respectively. The results are given in Table 3. TR-LLRR significantly outperforms all traditional methods and is also better than all deep learning methods on COIL20. TR–LLRR is the second best on COIL100. It is only a little worse (0.14%) than the best method DPSC on COIL100, but outperforms DPSC by 1% on COIL20. The performance improvement against TRR is significant, which verifies that tensor space handles view change well.

E. HANDWRITTEN DIGIT CLUSTERING

The Alphadigits dataset[1] is also employed for evaluating the subspace clustering performance of LR-TTRR. The dataset consists of 36 clusters (binary digits '0-9' and capital letters 'A-Z') with 39 images per cluster. Each image is in the size of 20 × 16 pixels. We divide the dataset into four groups, consisting of '0-9', 'A-J', 'K-T', and 'U-Z', respectively. All possible combinations of subsets with cluster number $K \in \{2, 5, 8, 10\}$ are collected within each group. We set the parameters $(\gamma_1, \gamma_2, \beta, k)$ of LR-TTRR to (0.04, 0.7, 0.9, 8). The average and median clustering errors are reported

[1] http://www.cs.nyu.edu/~roweis/data.html

TABLE 3. Clustering error (%) on COIL20 and COIL100.

Methods	SSC	S^3C	SSC-OMP	KSSC	LSR	LRR	KLRR	LRSC	NSLLRR	TRR
COIL20	13.68	7.87	29.86	19.65	31.77	18.82	31.17	25.84	38.00	40.30
COIL100	44.90	33.57	58.77	46.18	49.33	46.82	49.83	48.59	-	47.6
Methods	DSC-Net	DeepCogSC	DPSC	S^2ConvSCN	StructAE	SC-LALRG	FGNSC	NSC-AGA	IWPNR	TR-LLRR
COIL20	5.14	4.89	2.46	2.14	8.96	11.32	9.93	4.65	17.4	**1.46**
COIL100	30.96	-	**24.6**	26.67	-	-	-	25.62	-	24.74

TABLE 4. Clustering error (%) on Alphadigits.

Clusters (#)	2		5		8		10	
Error (%)	AVG	MED	AVG	MED	AVG	MED	AVG	MED
SSC	5.70	2.56	23.26	25.12	30.00	30.01	32.14	32.82
LRR	7.76	3.84	23.34	23.59	30.50	30.44	33.67	32.56
LRSC	15.81	8.97	37.77	37.95	47.98	48.08	50.77	51.03
KSSC	5.42	2.56	22.64	23.08	31.06	32.05	33.85	34.36
S^3C	6.66	5.13	27.11	28.72	35.83	34.94	37.18	38.21
TRR	4.40	2.56	18.98	16.41	27.94	27.56	31.62	30.51
VR3	3.90	**1.28**	18.81	16.92	26.87	25.96	26.15	23.33
NVR3	3.79	**1.28**	18.43	16.41	26.73	25.96	23.59	23.08
LR-TTRR	**3.49**	**1.28**	**13.46**	**10.77**	**19.68**	**19.55**	**21.79**	**18.21**

in Table 4. It can be seen that LR-TTRR works well for different clustering numbers and is the best among the compared methods. NVR3 and VR3 are the second and third best methods, which also confirms the importance of spatial information for subspace clustering of tensor data.

VI. CONCLUSION

In this paper, we propose LR-TTRR for subspace clustering of tensor data. By incorporating tensor mode-d product for tensor self-representation and removing noise from both the input space and the projected space, LR-TTRR is very robust and efficient in subspace clustering. Comprehensive experiments on several benchmark image datasets show significant improvement of LR-TTRR over existing methods.

APPENDIX

PROOF OF THEOREM 2

A. PRELIMINARIES

To analyze the convergence property, we first give some relevant definitions, assumptions and lemmas from [49] and [50].

Definition 4: A function $\psi(\mathbf{x})$ *satisfies the Kurdyka-Lojasiewicz (KL) property at point* $\bar{\mathbf{x}} \in dom(\partial\psi)$ *if there exists* $\theta \in [0, 1)$ *such that* $\frac{|\psi(\mathbf{x})-\psi(\bar{\mathbf{x}})|^\theta}{dist(\mathbf{0},\partial\psi(\mathbf{x}))}$ *is bounded around* $\bar{\mathbf{x}}$ *under the following notations:* $0^0 = 1$, $\infty/\infty = 0/0 = 0$. *In other words, in a certain neighborhood* $\mathcal{U}$ *of* $\bar{\mathbf{x}}$, *there exists* $\phi(s) = cs^{1-\theta}$ *for some* $c > 0$ *and* $\theta \in [0, 1)$ *such that the KL inequality holds:*

$$\phi'(|\psi(\mathbf{x}) - \psi(\bar{\mathbf{x}})|)dist(\mathbf{0}, \partial\psi(\mathbf{x})) \geq 1 \tag{14}$$

for any $\mathbf{x} \in \mathcal{U} \cap dom(\partial\psi)$ *and* $\psi(\mathbf{x}) \neq \psi(\bar{\mathbf{x}})$, *where* $dom(\partial(\psi)) \triangleq \{\mathbf{x} : \partial\psi(\mathbf{x}) \neq \emptyset\}$ *and* $dist(\mathbf{0}, \partial\psi(\mathbf{x}) \triangleq \min\{\|\mathbf{y}\| : \mathbf{y} \in \partial\psi(\mathbf{x})\}$.

Definition 5: We call a set $\mathcal{X}$ *block multiconvex if its projection to each block of variable is convex, namely, for each* i *and fixed* $(s-1)$ *blocks* $\mathbf{x}_1, \cdots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_s$, *the set* $\mathcal{X}_i(\mathbf{x}_1, \cdots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_s) \triangleq \{\mathbf{x}_i \in \mathbb{R}^{n_i} : (\mathbf{x}_1, \cdots, \mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}, \cdots, \mathbf{x}_s) \in \mathcal{X}\}$ *is convex.*

The optimization problem to be considered is given by

$$\min_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x}_1, \cdots, \mathbf{x}_s) = f(\mathbf{x}_1, \cdots, \mathbf{x}_s) + \sum_{i=1}^{s} r_i(\mathbf{x}_i), \tag{15}$$

where variable $\mathbf{x}$ is decomposed into s blocks $\mathbf{x}_1, \cdots, \mathbf{x}_s$, the set $\mathcal{X}$ of feasible points is assumed to be a closed and block multiconvex subset of $\mathbb{R}^n$, f is assumed to be a differentiable and block multiconvex function, and $r_i, i = 1, \cdots, s$ are extended-value convex functions. The definition of Nash point is given as follows.

Definition 6: If $F(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i-1}, \bar{\mathbf{x}}_i, \bar{\mathbf{x}}_{i+1}, \cdots, \bar{\mathbf{x}}_s) \leq F(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i-1}, \mathbf{x}_i, \bar{\mathbf{x}}_{i+1}, \cdots, \bar{\mathbf{x}}_s) \quad \forall \mathbf{x}_i \in \bar{\mathcal{X}}_i, i = 1, \cdots, s,$ *where* $\bar{\mathcal{X}}_i = \mathcal{X}_i(\bar{\mathbf{x}}_1, \cdots, \bar{\mathbf{x}}_{i-1}, \bar{\mathbf{x}}_{i+1}, \cdots, \bar{\mathbf{x}}_s)$, *we call* $\bar{\mathbf{x}}$ *a Nash point or block coordinate-wise minimizer.*

The following assumptions are introduced for analyzing the convergence property of optimizing problem (15).

Assumption 1: F *is continuous in* $dom(F)$ *and* $inf_{x\in dom(F)}F(x) > -\infty$. *Problem (15) has a Nash point.*

Assumption 2: Each block function f_i *is strongly convex, namely,* $f_i(\mathbf{u}) - f_i(\mathbf{v}) \geq \langle\nabla f_i(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle + \frac{L}{2}\|\mathbf{u} - \mathbf{v}\|^2$.

We summarize Theorems 2.8 and 2.9 in [50] as Lemma 1 in the following.

Lemma 1: If the sequence $\{\mathbf{x}^{(t)}\}$ *generated by alternatively optimizing each block of problem (15) has a finite limit point* $\bar{\mathbf{x}}$ *and* F *satisfies the KL inequality (14) with* $\phi(s) = cs^{1-\theta}$ *for* $c > 0$ *and* $\theta \in [0, 1)$*, the sequence* $\{\mathbf{x}^{(t)}\}$ *converges to* $\bar{\mathbf{x}}$*, which is a critical point of (15). The convergence rate satisfies: 1) If* $\theta = 0$*, then* $\mathbf{x}^{(t)}$ *converges to* $\bar{\mathbf{x}}$ *in finitely many iterations. 2) If* $\theta \in (0, \frac{1}{2}]$*,* $\|\mathbf{x}^{(t)} - \bar{\mathbf{x}}\| \leq C\tau^t$ *for all* $t > t_0$*, for certain* $t_0 > 0, C > 0, \tau \in [0, 1)$*. 3) If* $\theta \in (\frac{1}{2}, 1)$*,* $\|\mathbf{x}^{(t)} - \bar{\mathbf{x}}\| \leq Ct^{-(1-\theta)/(2\theta-1)}$ *for all* $t > t_0$*, for certain* $t_0 > 0$*,* $C > 0$*.*

Lemma 1 provides an estimation of convergence based on Lojasiewicz exponent θ. The following lemma (Theorem 4.2 in [49]) gives explicit bounds of Lojasiewicz exponent for polynomial function.

Lemma 2: For any polynomial $f : \mathbb{R}^n \to \mathbb{R}$ *of degree* d *the Lojasiewicz exponent at* $\mathbf{0}$ *is less than or equal to* $1 - (d(3d-3)^{n-1})^{-1}$*. More precisely, if* $f(\mathbf{0}) = 0$ *and* $\nabla f(\mathbf{0}) = \mathbf{0}$*, then for any* $r_0 > 0$ *there exist* $\epsilon > 0$ *and* $C > 0$ *such that*

$$|\nabla f(\mathbf{x})| \geq C|f(\mathbf{x})|^\theta \tag{16}$$

for any $\mathbf{x} \in B^n(r_0)$ *with* $|f(\mathbf{x})| < \epsilon$*, where* θ *is given by*

$$\theta = 1 - (d(3d-3)^{n-1})^{-1}. \tag{17}$$

Note that for function f such that $f(\mathbf{0}) \neq 0$ and $\nabla f(\mathbf{0}) = \mathbf{0}$, Eq. (16) changes into

$$|\nabla f(\mathbf{x})| \geq C|f(\mathbf{x}) - f(\bar{\mathbf{x}})|^\theta, \tag{18}$$

where $\bar{\mathbf{x}}$ is a critical point of f and $\mathbf{x}$ is in the neighbourhood of $\bar{\mathbf{x}}$.

B. MAIN PROOF

Define vectorized functions

$$f(\text{vec}(U_1), \text{vec}(V_1), \ldots, \text{vec}(U_N), \text{vec}(V_N), \text{vec}(\mathcal{S})) = \frac{1}{2}\|\mathcal{X} - \mathcal{X}\prod_{i=1}^{N} \times_i U_i V_i - \mathcal{S}\|_F^2 + \frac{1}{2}\sum_{i}^{N} \beta_i \|U_i V_i\|_F^2$$

and

$$r(\text{vec}(\mathcal{S})) = \lambda\|\mathcal{S}\|_1.$$

The objective function of T-GoDec (Eq. (3) in the main manuscript)

$$\frac{1}{2}\|\mathcal{X} - \mathcal{X}\prod_{i=1}^{N} \times_i U_i V_i - \mathcal{S}\|_F^2 + \lambda\|\mathcal{S}\|_1 + \frac{1}{2}\sum_{i}^{N} \beta_i \|U_i V_i\|_F^2$$

can be rewritten as

$$F(\text{vec}(U_1), \text{vec}(V_1), \ldots, \text{vec}(U_N), \text{vec}(V_N), \text{vec}(\mathcal{S})) = f(\text{vec}(U_1), \text{vec}(V_1), \ldots, \text{vec}(U_N), \text{vec}(V_N), \text{vec}(\mathcal{S})) + r(\text{vec}(\mathcal{S})).$$

For notation simplification, we denote the concatenation of the vector variables as $\mathbf{x}$ and the corresponding minimizer as $\bar{\mathbf{x}}$. Note that $f(\mathbf{x})$ is a polynominal function. If values correspond to absolute functions are not zeros, $F(\mathbf{x})$ is a polynominal function. By Lemma 2, the following inequality holds

$$|\nabla F(\mathbf{x})| \geq C|F(\mathbf{x}) - F(\bar{\mathbf{x}})|^\theta, \tag{19}$$

where $\bar{\mathbf{x}}$ is the critical point of $F(\mathbf{x})$. When some values correspond to absolute functions are zeros, denote such vector variable as $\mathbf{x}'$. By letting the set Ω denote the indicator of the zero values, we have $P_\Omega(\mathbf{x}') = \mathbf{0}$, where $P_\Omega(\cdot)$ is to project the vector to the set Ω. By assuming that $P_{\bar{\Omega}}(\mathbf{x}') = P_{\bar{\Omega}}(\mathbf{x})$, we have

$$|F(\mathbf{x}) - F(\bar{\mathbf{x}})| > |F(\mathbf{x}') - F(\bar{\mathbf{x}})| \tag{20}$$

From the definition of subgradient, we have

$$\nabla f(\mathbf{x}) \in \partial f(\mathbf{x}'). \tag{21}$$

Thus, $|\partial F(\mathbf{x}')| \geq C|F(\mathbf{x}') - F(\bar{\mathbf{x}})|^\theta$. By Lemma 2, the explicit estimation of the Lojasiewicz exponent of $F(\mathbf{x})$ is given by Eq. (17). It is also the exponent in KL inequality.

Since the alternative optimization procedure yields a sequence of decreasing values, the limit point of $\mathbf{x}$ is a Nash point. When fixing the others, for blocks U_i and V_i, the function f is strongly convex. Since d is 2 and n is usually a large number, we have $\theta \in (\frac{1}{2}, 1)$. By Lemma 1, the conclusion for Theorem 2 is proved.

REFERENCES

[1] T. Zhang, G. Su, C. Qing, X. Xu, B. Cai, and X. Xing, "Hierarchical lifelong learning by sharing representations and integrating hypothesis," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.
[2] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.
[3] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[4] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 347–360.
[5] L. Zhuang, S. Gao, J. Tang, J. Wang, Z. Lin, Y. Ma, and N. Yu, "Constructing a nonnegative low-rank and sparse graph with data-adaptive features," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3717–3728, Nov. 2015.
[6] C.-G. Li and R. Vidal, "Structured sparse subspace clustering: A unified optimization framework," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Oct. 2015, pp. 277–286.
[7] S. Li and Y. Fu, "Learning balanced and unbalanced graphs via low-rank coding," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1274–1287, May 2015.
[8] X. Peng, S. Xiao, J. Feng, W.-Y. Yau, and Z. Yi, "Deep subspace clustering with sparsity prior," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1925–1931.
[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.
[11] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 24–33.
[12] Y. Jiang, Z. Yang, Q. Xu, X. Cao, and Q. Huang, "When to learn what: Deep cognitive subspace clustering," in *Proc. ACM Multimedia*, 2018, pp. 718–726.
[13] L. Zhou, X. Bai, D. Wang, X. Liu, J. Zhou, and E. Hancock, "Latent distribution preserving deep subspace clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 4440–4446.
[14] J. Zhang, C.-G. Li, C. You, X. Qi, H. Zhang, J. Guo, and Z. Lin, "Self-supervised convolutional subspace clustering network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5473–5482.

[15] C. Peng, Z. Kang, and Q. Cheng, "Subspace clustering via variance regularized ridge regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2931–2940.
[16] J. M. Landsberg, *Tensors: Geometry and Applications*. Providence, RI, USA: AMS, 2012.
[17] M. Yang, W. Li, and M. Xiao, "On identiability of higher order block term tensor decompositions of rank rank-1," *Linear Multilinear Algebra*, p. 123, Jul. 2018. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/03081087.2018.1502251
[18] X. Ben, P. Zhang, Z. Lai, R. Yan, X. Zhai, and W. Meng, "A general tensor representation framework for cross-view gait recognition," *Pattern Recognit.*, vol. 90, pp. 87–98, Jun. 2019.
[19] Y. Fu, J. Gao, D. Tien, and Z. Lin, "Tensor LRR based subspace clustering," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2014, pp. 1877–1884.
[20] Y. Fu, J. Gao, D. Tien, Z. Lin, and X. Hong, "Tensor LRR and sparse coding-based subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 2120–2133, Spe. 2016.
[21] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1582–1590.
[22] X. Zhang, B. Chen, H. Sun, Z. Liu, Z. Ren, and Y. Li, "Robust low-rank kernel subspace clustering based on the schatten p-norm and correntropy," *IEEE Trans. Knowl. Data Eng.*, to be published.
[23] X. Li, Q. Lu, Y. Dong, and D. Tao, "Robust subspace clustering by cauchy loss function," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 2067–2078, Jul. 2019.
[24] B. Li, H. Lu, Y. Zhang, Z. Lin, and W. Wu, "Subspace clustering under complex noise," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 930–940, Apr. 2019.
[25] X. Peng, Z. Yi, and H. Tang, "Robust subspace clustering via thresholding ridge regression," in *Proc. Adv. Artif. Intell.*, 2015, pp. 3827–3833.
[26] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2016.
[27] M. Yin, J. Gao, and Z. Lin, "Laplacian regularized low-rank representation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 504–517, Mar. 2016.
[28] Z. Kang, H. Pan, S. C. H. Hoi, and Z. Xu, "Robust graph learning from noisy data," *IEEE Trans. Cybern.*, to be published.
[29] T. Zhang, X. Wang, X. Xu, and C. L. P. Chen, "GCB-Net: Graph convolutional broad network and its application in emotion recognition," *IEEE Trans. Affect. Comput.*, to be published.
[30] Y. Sui, G. Wang, and L. Zhang, "Sparse subspace clustering via low-rank structure propagation," *Pattern Recognit.*, vol. 95, pp. 261–271, Nov. 2019.
[31] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowl.-Based Syst.*, vol. 163, pp. 510–517, Jan. 2019.
[32] S. Xiao, M. Tan, D. Xu, and Z. Y. Dong, "Robust kernel low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2268–2281, Nov. 2016.
[33] Z. Kang, C. Peng, and Q. Cheng, "Kernel-driven similarity learning," *Neurocomputing*, vol. 267, pp. 210–219, Dec. 2017.
[34] Z. Kang, H. Xu, B. Wang, H. Zhu, and Z. Xu, "Clustering with similarity preserving," *Neurocomputing*, vol. 365, pp. 211–218, Nov. 2019.
[35] G. Liu, Q. Liu, and P. Li, "Blessing of dimensionality: Recovering mixture data via dictionary pursuit," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 47–60, Jan. 2017.
[36] K. Guo, L. Liu, X. Xu, D. Xu, and D. Tao, "GoDec+: Fast and robust low-rank matrix decomposition based on maximum correntropy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2323–2336, Jun. 2018.
[37] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, May 2011.
[38] C. Lu, J. Tang, M. Lin, L. Lin, S. Yan, and Z. Lin, "Correntropy induced L2 graph for robust subspace clustering," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1801–1808.
[39] J. Yang, J. Liang, K. Wang, P. Rosin, and M.-H. Yang, "Subspace clustering via good neighbors," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[40] T. Zhou and D. Tao, "GoDec: Randomized low-rank & sparse matrix decomposition in noisy case," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 33–40.
[41] K. Guo, X. Xu, and D. Tao, "Discriminative GoDec+ for classification," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3414–3429, Jul. 2017.
[42] R. Liu, Z. Lin, F. De la Torre, and Z. Su, "Fixed-rank representation for unsupervised visual learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 598–605.
[43] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[44] D. Tao, X. Li, X. Wu, W. Hu, and S. J. Maybank, "Supervised tensor learning," *Knowl. Inf. Syst.*, vol. 13, no. 1, pp. 1–42, 2007.
[45] T. Zhou and D. Tao, "Shifted subspaces tracking on sparse outlier for motion segmentation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 1946–1952.
[46] E. S. Georgi, *Linear Algebra*. New York, NY, USA: Dover, 1997.
[47] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[48] T. Zhou and D. Tao, "Greedy bilateral sketch, completion & smoothing," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2013, pp. 650–658.
[49] D. D'Acunto and K. Kurdyka, "Explicit bounds for the Łojasiewicz exponent in the gradient inequality for polynomials," *Annales Polonici Math.*, vol. 1, no. 87, pp. 51–61, 2005.
[50] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1758–1789, 2013.
[51] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3918–3927.
[52] V. M. Patel and R. Vidal, "Kernel sparse subspace clustering," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 2849–2853.
[53] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognit. Lett.*, vol. 43, pp. 47–61, Jul. 2013.
[54] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu, "Non-negative low rank and sparse graph for semi-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2328–2335.
[55] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.
[56] Q. Ji, Y. Sun, J. Gao, Y. Hu, and B. Yin, "Nonlinear subspace clustering via adaptive graph regularized autoencoder," *IEEE Access*, vol. 7, pp. 74122–74133, 2019.
[57] J. Zheng, C. Lu, H. Yu, W. Wang, and S. Chen, "Iterative reconstrained low-rank representation via weighted nonconvex regularizer," *IEEE Access*, vol. 6, pp. 51693–51707, 2018.
[58] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.
[59] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Dec. 1994, pp. 138–142.
[60] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library," Columbia Univ., New York, NY, USA, Tech. Rep., 1996.

KAILING GUO received the B.S. and Ph.D. degrees from the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China, in 2011 and 2017, respectively. From 2015 to 2017, he was a Visiting Ph.D. Student with the Center for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney. He is currently an Assistant Professor with the School of Electronics and Information, and a Postdoctoral Fellow with the School of Computer Science and Engineering, South China University of Technology. His current research interests include computer vision and machine learning.

TONG ZHANG (S'12–M'16) received the B.S. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2009, and the M.S. degree in applied mathematics and the Ph.D. degree in software engineering from the University of Macau, Macau, China, in 2011 and 2016, respectively. He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, China. His current research interests include affective computing, evolutionary computation, neural networks, and other machine learning techniques and their applications. He has been involving in publication matters for many IEEE conferences.

XIANGMIN XU (M'13–SM'19) received the Ph.D. degree from the South China University of Technology, Guangzhou, China. He is currently a Full Professor with the School of Electronic and Information Engineering, South China University of Technology. His current research interests include image/video processing, human–computer interaction, computer vision, and machine learning.

XIAOFEN XING received the B.S., M.S., and Ph.D. degrees from the South China University of Technology, China, in 2001, 2004, and 2013, respectively. She has been an Associate Professor with the School of Electronic and Information Engineering, South China University of Technology, since 2007. Her current research interests include image/video processing, human–computer interaction, and video surveillance.