Pattern Recognition 106 (2020) 107441

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Multi-view subspace clustering via simultaneously learning the representation tensor and affinity matrix

Yongyong Chen[a], Xiaolin Xiao[b], Yicong Zhou[a,*]

[a] Department of Computer and Information Science, University of Macau, Macau 999078, China
[b] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.

ARTICLE INFO

Article history:
Received 27 June 2019
Revised 1 March 2020
Accepted 7 May 2020
Available online 16 May 2020

Keywords:
Multi-view subspace clustering
Low-rank tensor representation
Tensor-singular value decomposition
Adaptive weights
Local manifold

ABSTRACT

Multi-view subspace clustering aims at separating data points into multiple underlying subspaces according to their multi-view features. Existing low-rank tensor representation-based multi-view subspace clustering algorithms are robust to noise and can preserve the high-order correlations of multi-view features. However, they may suffer from two common problems: (1) the local structures and different importance of each view feature are often neglected; (2) the low-rank representation tensor and affinity matrix are learned separately. To address these issues, we propose a unified framework to learn the Graph regularized Low-rank representation Tensor and Affinity matrix (GLTA) for multi-view subspace clustering. In the proposed GLTA framework, the tensor singular value decomposition-based tensor nuclear norm is adopted to explore the high-order cross-view correlations. The manifold regularization is exploited to preserve the local structures embedded in high-dimensional space. The importance of different features is automatically measured when constructing the final affinity matrix. An iterative algorithm is developed to solve GLTA using the alternating direction method of multipliers. Extensive experiments on seven challenging datasets demonstrate the superiority of GLTA over the state-of-the-art methods.

1. Introduction

Subspace clustering [1] has gained increasing attention in pattern recognition and machine learning communities [2–4]. According to the available sources, subspace clustering methods can be roughly grouped into two categories: single-view subspace clustering and multi-view subspace clustering.

Single-view subspace clustering: Single-view subspace clustering is the clustering of data points into multiple subspaces while finding a low-dimensional subspace to fit each group of data points [2]. Sparse subspace clustering (SSC) [2] and low-rank representation (LRR) [3] are two representative works of single-view subspace clustering. Many variants of SSC and LRR have been proposed [5,6]. However, these methods perform the clustering task using only single-view feature and fail to explore the correlation among the features of different sources.

Multi-view subspace clustering: “Feature” refers to “an individual measurable property or characteristic of an object”. For example, three typical features of images are color, textures, and edges. “View” usually refers to the sources of feature acquisition or the perspectives of feature estimation. For example, views may refer to Local Binary Pattern (LBP), Gabor, and Histogram of Oriented Gradients (HOG). In real applications, the data characteristics can be modeled from different views (or sources). For example, documents can be translated into different languages for natural language processing; for action recognition, action sequences may be captured by RGB, depth, and skeleton sensors. An intuitive example of multi-view features is shown in Fig. 1(b). The features from different views are complementary to each other since each view usually characterizes partial knowledge of the original object or data. This is the reason why multi-view clustering methods would achieve better performance than single-view clustering ones. For single-view clustering methods, there are two ways to handle multi-view features. They perform single-view clustering methods either on each feature individually, or on the concatenated features. However, these above two schemes may fail to make full use of the correlation among multiple features. Multi-view subspaces refer to multiple subspaces with multi-view features for a set of data points. Considering that a set of data points are usually drawn from a union of several subspaces, multi-view subspace clustering refers to the problem of separating data into multiple underlying subspaces according to their multi-view features. The main difference between the single-view subspace clustering and multi-view subspace clustering is that the former obtains the clustering results using the single feature while the later

[*] Corresponding author.
E-mail address: yicongzhou@um.edu.mo (Y. Zhou).

https://doi.org/10.1016/j.patcog.2020.107441
0031-3203/© 2020 Elsevier Ltd. All rights reserved.

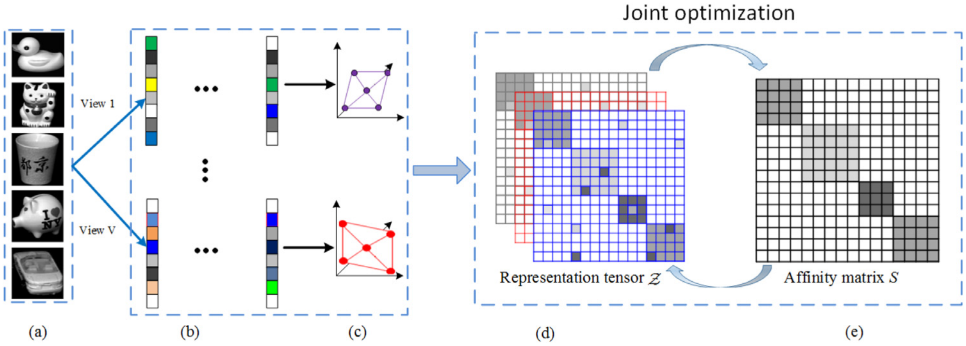


Fig. 1. The flowchart of the proposed GLTA. Multi-view features (b) are extracted from (a) original images. Unlike those existing multi-view clustering methods which learn the representation tensor (d) and affinity matrix (e) in two separate steps, the proposed GLTA not only learns (d) and (e) simultaneously, but also takes the local structures (c) into consideration in a unified manner.

uses multiple complementary features for clustering. Therefore, it is of vital importance to design efficient methods to learn the underlying intrinsic information hidden in different views for improving clustering performance.

Considerable efforts have been made to develop efficient multi-view clustering (MVC) algorithms, such as the multi-view k-means clustering [7], co-regularized MVC [8], canonical correlation analysis-based clustering [9], and low-rank representation-based MVC [10–14]. Due to the high efficiency and excellent performance, the low-rank representation-based multi-view subspace clustering has become the mainstream [12,13]. Generally, the procedures of these methods can be roughly divided into three steps: **Step 1:** learn the representation matrix or tensor using different subspace learning approaches, such as SSC [2], LRR [3], and others [11,12,15]; **Step 2:** construct the affinity matrix by averaging all representation matrices, where the affinity matrix (also called similarity matrix) aims at measuring the similarity between two data points; **Step 3:** obtain the clustering results using the spectral clustering algorithm [16] with the affinity matrix. The core of clustering is to construct an informative affinity matrix. This is mainly because the clustering performance highly depends on the affinity matrix. Many works focus on how to directly learn a well-designed representation matrix or tensor for the construction of the affinity matrix. For example, Maria et al. [4] proposed to learn a low-rank and sparse representation matrix. Wang et al. [17] used the intact space learning technique to learn an informative intactness-aware representation matrix. Zhu et al. [18] developed a structured multi-view subspace clustering method to learn general and specific representation matrices. The representation tensor was encoded either by the tensor nuclear norm [11,12] or by the diversity of all representations [19].

Although the above low-rank matrix or tensor representation-based MVC methods have achieved satisfactory performance, they still suffer from the following limitations: (1) they often ignore the local structures as shown in Fig. 1(c). The methods in [4,10–13,17,18] extend the (tensor) robust principal component analysis or (tensor) LRR [3,20,21] for multi-view subspace clustering by considering only the global low-rank property of the representation matrix or tensor. Thus, the locality and similarity information of samples may be ignored in the learning processes [22,23]; (2) they separately learn the low-rank representation and the affinity matrix. To obtain the clustering results, the methods in [11,12] first pursue the representation tensor in Step 1 by using different tensor rank approximations and then construct the final affinity matrix in Step 2 by averaging all representation matrices. In such a way, two critical factors in clustering, *i.e.*, the representation tensor and affinity matrix, are learned separately. This would ignore the high dependence between them. Thus, there is no guarantee of recovering an overall optimal clustering result. Meanwhile, the existing scheme of constructing the affinity matrix treats the representation matrices of different views equally. This may lead to unsatisfactory performance in real applications. It is mainly because different features characterize specific and partly independent information of the data and thus may have different contributions for the final clustering results [24]. An intuitive example is reported in Table 9 in Section 5.3. We can see that the clustering results of different views vary.

Such two concerns are not well solved in existing low-rank tensor representation-based MVC methods. In this paper, we propose a novel multi-view subspace clustering method by learning the Graph regularized Low-rank representation Tensor and Affinity matrix (GLTA) in a unified framework as shown in Fig. 1. Given the multi-view features as shown in Fig. 1(b), the representation tensor (Fig. 1(d)) and the affinity matrix (Fig. 1(e)) are learned simultaneously. Considering the fact that different view features may have various contributions to the clustering performance, GLTA can automatically assign corresponding weight to each view by a constrained quadratic programming. Meanwhile, the local structures as shown in Fig. 1(c) are also preserved in the construction of the representation tensor. Therefore, the main purpose of this manuscript is to propose an efficient multi-view subspace clustering method not only to learn the low-rank representation tensor and affinity matrix simultaneously, but also to integrate the local structures into a unified manner. In such a way, the global and local structures of multi-view features can be well explored. The main contributions of this work are summarized as follows.

- Different from the existing low-rank representation-based MVC which learns the representation matrix/tensor and affinity matrix in a sequential way, the proposed GLTA not only learns the representation tensor and affinity matrix simultaneously, but also considers the local structures and the various contributions of multi-view features.
- GLTA exploits the tensor singular value decomposition-based tensor nuclear norm to encode the low-rank property, adopts the manifold regularization to depict the local structures, and adaptively assigns various weights for multi-view features when constructing the final affinity matrix. In this way, the high-order correlations among different views and the local structures can be explicitly captured.
- An iterative algorithm is developed to solve GLTA using the alternating direction method of multipliers. Extensive experiment evaluations on seven challenging datasets demonstrate that GLTA outperforms the state-of-the-art clustering approaches.

The remaining of this paper is structured as follows. Section 2 briefly reviews related works on single-view and multi-view subspace clustering. Section 3 discusses the mathematical background. In Section 4, we introduce the proposed method for multi-view subspace clustering and design an iterative algorithm. We evaluate the performance of the proposed method in Section 5 and conclude the whole paper in Section 6.

2. Related work

In this section, we briefly review several popular single-view and multi-view subspace clustering methods based on low-rank matrix or tensor approximation.

2.1. Single-view subspace clustering

Under the basic assumption that the observed data points are generally drawn from low-dimensional subspaces, the popular methods for single-view subspace clustering can be generally formulated as follows:

$$\min_{Z,E} \Psi(Z) + \lambda\varphi(E) \quad s.t. \quad X = XZ + E, \quad diag(Z) = 0, \tag{1}$$

where $X \in \mathbb{R}^{d\times n}$ is the feature matrix with n samples and d-dimension feature. $\Psi(Z)$ is the regularization which imposes desired property on the representation matrix $Z \in \mathbb{R}^{n\times n}$. $\varphi(E)$ is to remove noise E. $\lambda > 0$ is a trade-off parameter. The main difference among works in [2,3,15] is how to choose Ψ. For instance, SSC minimizes the l_1-norm [2] on Z to learn the local structure of data, while LRR [3] enforces the low-rank constraint on Z to capture the global structure. Least squares regression (LSR) [15] exploits the Frobenius-norm to model both Z and E.

Subspace clustering methods have achieved great success in various applications including face and scene image clustering [3], motion segmentation [3,25], object detection [26], community clustering in social networks [27], biclustering [28] and so on. For more information of subspace clustering, please refer to [1,29]. Since these methods assumed that the data lie in multiple linear subspaces, they may not be able to handle the nonlinear data. On the other hand, for multi-view data, especially the heterogeneous features, LRR, SSC, LSR and their variants [5,23,30] may cause a significant performance degradation [11,12,19] since they focus on single-view feature. Thus they cannot deal with the multi-view clustering task.

2.2. Multi-view subspace clustering

To tackle the above problem, multi-view subspace clustering [11,12,19,22,31–33] takes advantages of multi-view features to boost the clustering performance, and has shown superiority over its single-view counterpart [12]. Most of existing multi-view subspace clustering methods can be summarized as

$$\min_{Z^{(v)},E^{(v)}} \sum_{v=1}^{V} \Psi(Z^{(v)}) + \lambda\varphi(E^{(v)}) \quad s.t.$$
$$X^{(v)} = X^{(v)}Z^{(v)} + E^{(v)}, \; diag(Z^{(v)}) = 0, \; (v = 1, 2, \cdots, V), \tag{2}$$

where $X^{(v)}$ denotes the vth feature matrix, its corresponding representation matrix is denoted as $Z^{(v)} \in \mathbb{R}^{n\times n}$. d_v is the dimension of a sample vector in the vth feature matrix, n and V are the numbers of data points and views, respectively. Similar to Eq. (1), the studies in [4,11–13,18] used different regularizers to obtain different characteristics. For example, works in [4,18] used the nuclear norm and l_1 norm to preserve the consistency and diversity. Zhang et al. [11] exploited the unfolding-based tensor nuclear norm and $l_{2,1}$ norm to capture the high-order correlations. However, it may yield unsatisfactory performance in real applications, since the unfolding-based tensor nuclear norm is a loose approximation of Tucker rank [20,21,34]. To deal with this issue, a newly proposed tensor nuclear norm [20] was exploited in [12] to ensure the consensus among multiple views. The work in [35] proposed to learn the latent representation to overcome the noise interference. As previously mentioned, one essential limitation of these low-rank tensor representation-based MVC methods is that the representation tensor and affinity matrix are learned in a separate way.

This paper is an extension of our conference work [36]. In [36], we used Tucker decomposition to encode the low-rank property of the representation tensor. Due to the difficulty of the desired rank of Tucker decomposition, we utilize the tensor singular value decomposition-based tensor nuclear norm instead of the Tucker decomposition in this paper.

3. Preliminaries

Before further discussions, we introduce the fundamental formulas used in the paper and the tensor singular value decomposition (t-SVD)-based tensor nuclear norm (see *Definition* 3.3). For details of tensor and its applications, please see [37].

Following [37], calligraphy letters, capital letters, and lowercase letters (*e.g.*, $\mathcal{X}$, X, x) denote tensors, matrices, and vectors, respectively. The inner product of $\mathcal{X}$ and $\mathcal{Y}$ in $\mathbb{R}^{N_1\times N_2\times N_3}$ is defined as $\langle\mathcal{X},\mathcal{Y}\rangle = \mathbf{vec}(\mathcal{X})^T\mathbf{vec}(\mathcal{Y})$, and the Frobenius norm of $\mathcal{X}$ is defined as $\|\mathcal{X}\|_F = \sqrt{\langle\mathcal{X},\mathcal{X}\rangle}$. Operator $\mathbf{vec}(X)$ is to stacking all columns of the matrix X into a vector. We denote the l_1-norm of $\mathcal{X}$ as $\|\mathcal{X}\|_1 = \|\mathbf{vec}(\mathcal{X})\|_1$ and the infinity norm of $\mathcal{X}$ as $\|\mathcal{X}\|_\infty = \max_{i,j,k}|\mathcal{X}(i,j,k)|$. The kth frontal slice of tensor $\mathcal{X}$ is denoted as $\mathcal{X}^{(k)}$. Performing the fast Fourier transformation (FFT) along the tube fibers of $\mathcal{X}$ is denoted as $\hat{\mathcal{X}} = \mathbf{fft}(\mathcal{X}, [\,], 3)$. Likewise, we can obtain $\mathcal{X}$ from $\hat{\mathcal{X}}$ by the inverse FFT, *i.e.*, $\mathscr{X}$ =$\mathbf{ifft}(\hat{\mathcal{X}}, [\,], 3)$.

We first introduce several block operators which are the foundation of t-SVD [20,38]. For a tensor $\mathcal{X} \in \mathbb{R}^{N_1\times N_2\times N_3}$, its block circular matrix $\mathbf{bcirc}(\mathcal{X})$ and block diagonal matrix $\mathbf{bdiag}(\mathcal{X})$ are defined as

$$\mathbf{bcirc}(\mathcal{X}) = \begin{bmatrix} \mathcal{X}^{(1)} & \mathcal{X}^{(N_3)} & \cdots & \mathcal{X}^{(2)} \\ \mathcal{X}^{(2)} & \mathcal{X}^{(1)} & \cdots & \mathcal{X}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{X}^{(N_3)} & \mathcal{X}^{(N_3-1)} & \cdots & \mathcal{X}^{(1)} \end{bmatrix},$$

$$\mathbf{bdiag}(\mathcal{X}) = \begin{bmatrix} \mathcal{X}^{(1)} & & & \\ & \mathcal{X}^{(2)} & & \\ & & \ddots & \\ & & & \mathcal{X}^{(N_3)} \end{bmatrix}.$$

The block vectorization is defined as $\mathbf{bvec}(\mathcal{X}) = [\mathcal{X}^{(1)}; \cdots; \mathcal{X}^{(N_3)}]$. The inverse operations of **bvec** and **bdiag** are defined as $\mathbf{bvfold}(\mathbf{bvec}(\mathcal{X})) = \mathcal{X}$ and $\mathbf{bdfold}(\mathbf{bdiag}(\mathcal{X})) = \mathcal{X}$, respectively. Let $\mathcal{Y} \in \mathbb{R}^{N_2\times N_4\times N_3}$. The **t-product** $\mathcal{X} * \mathcal{Y}$ is an $N_1 \times N_4 \times N_3$ tensor,

$$\mathcal{X} * \mathcal{Y} = \mathbf{bvfold}(\mathbf{bcirc}(\mathcal{X}) * \mathbf{bvec}(\mathcal{Y})).$$

The **transpose** of $\mathcal{X}$ is $\mathcal{X}^T \in \mathbb{R}^{N_2\times N_1\times N_3}$ by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through N_3. The **identity tensor**$\mathcal{I} \in \mathbb{R}^{N_1\times N_1\times N_3}$ is a tensor whose first frontal slice is an $N_1 \times N_1$ identity matrix and the remaining frontal slices are zero. A tensor $\mathcal{X} \in \mathbb{R}^{N_1\times N_1\times N_3}$ is **orthogonal** if it satisfies

$$\mathcal{X}^T * \mathcal{X} \; = \mathcal{X} * \mathcal{X}^T = \mathcal{I}.$$

Based on the above knowledge, we can obtain the definition of t-SVD.

Definition 3.1 ((t-SVD))**.** For $\mathcal{X} \in \mathbb{R}^{N_1\times N_2\times N_3}$, its t-SVD is given by

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T,$$

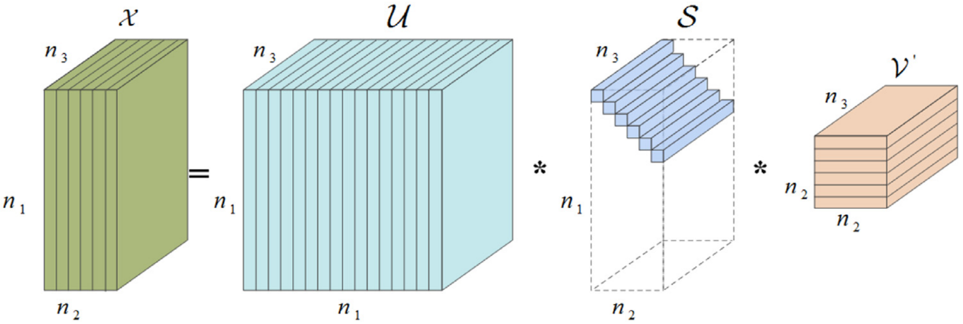


Fig. 2. The t-SVD of a tensor of size $N_1 \times N_2 \times N_3$.

where $\mathcal{U} \in \mathbb{R}^{N_1 \times N_1 \times N_3}$ and $\mathcal{V} \in \mathbb{R}^{N_2 \times N_2 \times N_3}$ are orthogonal tensors, $\mathcal{S} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is an f-diagonal tensor. Each of its frontal slices is a diagonal matrix.

Fig. 2 shows the t-SVD of a third-order tensor. As discussed in [38], t-SVD can be efficiently computed by matrix SVD in the Fourier domain. Based on t-SVD, the tensor multi-rank is given as follows.

Definition 3.2 (Tensor multi-rank)**.** The tensor multi-rank of a tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is a vector $r \in \mathbb{R}^{N_3 \times 1}$ with its ith element being the rank of the ith frontal slice of $\hat{\mathcal{X}}$.

Derived from the relationship between the rank function and nuclear norm in the matrix case, the following t-SVD-based tensor nuclear norm (t-SVD-TNN) is obtained.

Definition 3.3 (t-SVD-TNN)**.** The t-SVD-TNN of a tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, denoted as $\|\mathcal{X}\|_{\circledast}$, is defined as the sum of singular values of all the frontal slices of $\hat{\mathcal{X}}$, *i.e.*,

$$\|\mathcal{X}\|_{\circledast} = \sum_{i=1}^{\min\{N_1,N_2\}} \sum_{k=1}^{N_3} |\hat{\mathcal{S}}(i,i,k)|. \tag{3}$$

Note that t-SVD-TNN is a valid norm which is the tightest convex relaxation to l_1-norm of the tensor multi-rank [20].

4. GLTA for MVC

In this section, we first propose a novel MVC method to learn the Graph regularized Low-rank representation Tensor and the Affinity matrix (GLTA) in a unified framework. Afterward, we design an iterative algorithm to solve GLTA by the alternating direction method of multipliers (ADMM).

4.1. The proposed GLTA

To address those concerns as discussed in Section 1 whlie learning a reliable affinity matrix, we consider the following three aspects:

- According to the self-representation property [2,3], each data point in the vth view can be represented as a linear combination of other points, *i.e.*, $X^{(v)} = X^{(v)}Z^{(v)} + E^{(v)}$, where $E^{(v)}$ denotes noise. Clearly, it is in general ill-posed without any restriction to obtain the representation matrix $Z^{(v)}$ and noise $E^{(v)}$ from $X^{(v)}$. Inspired by Liu et al. [3], Zhang et al. [11], Xie et al. [12], we introduce the low-rank tensor approximation to explore the high-order correlations between multiple views.
- Following the assumption of graph embedding in [39] that if two data points $x_i^{(v)}$ and $x_j^{(v)}$ are close in the original space, their low-dimensional representations $z_i^{(v)}$ and $z_j^{(v)}$ should be close to each other. This can be illustrated by the following manifold regularization:

$$\sum_i \sum_j \|z_i^{(v)} - z_j^{(v)}\|_2^2 W_{ij}^{(v)} = \frac{1}{2} tr(Z^{(v)} L^{(v)} Z^{(v)^T}), \tag{4}$$

where $W_{i,j}^{(v)}$ is the similarity between $x_i^{(v)}$ and $x_j^{(v)}$. The vth view graph Laplacian matrix $L^{(v)}$ [23] is constructed in the k-nearest neighbor fashion and defined as $L^{(v)} = D^{(v)} - W^{(v)}$, where $D^{(v)}$ is a diagonal matrix and $D_{i,i}^{(v)} = \sum_j W_{i,j}^{(v)}$. The trace of a matrix is denoted as $tr(\ \cdot\ )$. Using Eq. (4), the local structures hidden in high-dimensional space can be well preserved [23].
- As in [11,12,25], an intuitive way to construct the affinity matrix is $S = \frac{1}{V}\sum_v \left(|Z^{(v)}| + |Z^{(v)^T}|\right)$. This means that all representation matrices are treated equally. As discussed before, we need to assign different weights according to the importance of each view. Therefore, we learn S by minimizing the linear combination of the residual $\|Z^{(v)} - S\|_F^2$ for each view.

Considering the above three aspects, the proposed GLTA is formulated as

$$\left\{ \begin{array}{l} \min\limits_{\mathcal{Z},E,S,\omega} \underbrace{\Psi(\mathcal{Z})}_{\text{low-rank tensor representation}} + \sum\limits_{v=1}^{V} \left(\underbrace{\lambda_1 \|E^{(v)}\|_{2,1}}_{\text{noise}} + \underbrace{\lambda_2 tr\left(Z^{(v)} L^{(v)} Z^{(v)^T}\right)}_{\text{local manifold}} + \underbrace{\lambda_3 \omega_v \|Z^{(v)} - S\|_F^2}_{\text{consensus representation}} \right) + \gamma \|\omega\|_2^2 \\ s.t.\ X^{(v)} = X^{(v)}Z^{(v)} + E^{(v)},\ (v = 1, 2, \cdots, V),\ \mathcal{Z} = \Phi(Z^{(1)}, Z^{(2)}, \cdots, Z^{(V)}),\ \omega \geq 0,\ \Sigma_v \omega_v = 1, \end{array} \right\} \tag{5}$$

where $\|\ \cdot\ \|_{2,1}$ is the $l_{2,1}$-norm to remove the sample-specific corruptions. $E = [E^{(1)}; E^{(2)}; \cdots; E^{(V)}]$. Using $\Phi(\ \cdot\ )$, all representation matrices $\{Z^{(v)}\}$ are merged to construct a 3-order representation tensor $\mathcal{Z} \in \mathbb{R}^{n \times n \times V}$ as shown in Fig. 1(b). The regularization $\Psi(\mathcal{Z})$ is to depict the low-rank property of $\mathcal{Z}$. S is the final affinity matrix to be learned. λ_1, λ_2, λ_3 and γ are nonnegative parameters. $\omega = [\omega_1, \omega_2, \cdots, \omega_V]$ is the weight vector, whose entry ω_v is the relative weight of the vth view.

Remarks:

- From Eq. (5), we can see that the low-rank representation tensor $\mathcal{Z}$ and the affinity matrix S can be simultaneously learned in a unified framework, such that a meaningful affinity matrix can be obtained as the input of the spectral clustering algorithm in [16] to yield the clustering results;

- The first term in Eq. (5) is to depict the low-rank property (the global structure and high-order correlations) of the representation tensor $\mathcal{Z}$ while the second term can model the sample-specific corruptions [3];
- Using the manifold regularization, *i.e.*, the third term in Eq. (5), the local structures of multi-view data can be preserved such that the data points being close in the original space still have similar representations;
- The last term in Eq. (5) enforces different weights on different views, so as to obtain an informative affinity matrix. To overcome the difficulty of weight allocation, Eq. (5) can adaptively assign weights on different features by a constrained quadratic programming.

The proposed unified framework GLTA can cover several existing state-of-the-art MVC methods. For example, LT-MSC [11] and t-SVD-MSC [12] can be regarded as two special cases if $\Psi(\mathcal{Z})$ is selected as the unfolding-based tensor nuclear norm and t-SVD-TNN respectively, and $\lambda_2 = \lambda_3 = \gamma = 0$. The works in [21,34] have pointed out that t-SVD-TNN has achieved superior performance than the other tensor decomposition forms including CANDECOMP/PARAFAC decomposition and Tucker decomposition in computer vision. Inspired by this, in the following subsection, we will exploit the t-SVD-TNN to encode the low-rank tensor property of $\mathcal{Z}$.

4.2. Optimization of GLTA

Using the t-SVD-TNN defined in Eq. (3), *i.e.*, $\Psi(\mathcal{Z}) = \|\mathcal{Z}\|_{\circledast}$, the model in Eq. (5) can be formulated as:

$$\begin{aligned}
&\min_{\mathcal{Z},E,S,\omega} \|\mathcal{Z}\|_{\circledast} + \sum_{v=1}^{V} \Big(\lambda_1 \|E^{(v)}\|_{2,1} \\
&\quad + \lambda_2 tr\big(Z^{(v)} L^{(v)} Z^{(v)^T}\big) + \lambda_3 \omega_v \|Z^{(v)} - S\|_F^2 \Big) + \gamma \|\omega\|_2^2 \\
&\quad s.t. \;\; X^{(v)} = X^{(v)} Z^{(v)} + E^{(v)}, \quad (v = 1, 2, \cdots, V), \\
&\mathcal{Z} = \Phi(Z^{(1)}, Z^{(2)}, \cdots, Z^{(V)}), \;\; \omega \geq 0, \;\; \Sigma_v \omega_v = 1.
\end{aligned} \tag{6}$$

Since we impose both global low-rankness and local structure priors on $\mathcal{Z}$, Eq. (6) is coupled with respect to $\mathcal{Z}$. To make $\mathcal{Z}$ separable, we adopt the variable-splitting technique [40,41] and introduce one auxiliary tensor variable $\mathcal{Y}$. Then, Eq. (6) can be reformulated as the following optimization problem:

$$\begin{aligned}
&\min_{\mathcal{Y},\mathcal{Z},E,S,\omega} \|\mathcal{Z}\|_{\circledast} + \sum_{v=1}^{V} \Big(\lambda_1 \|E^{(v)}\|_{2,1} \\
&\quad + \lambda_2 tr\big(Y^{(v)} L^{(v)} Y^{(v)^T}\big) + \lambda_3 \omega_v \|Y^{(v)} - S\|_F^2 \Big) + \gamma \|\omega\|_2^2 \\
&\quad s.t. \;\; X^{(v)} = X^{(v)} Y^{(v)} + E^{(v)}, \quad (v = 1, 2, \cdots, V), \\
&\mathcal{Z} = \Phi(Z^{(1)}, Z^{(2)}, \cdots, Z^{(V)}), \omega \geq 0, \;\; \Sigma_v \omega_v = 1, \;\; \mathcal{Z} = \mathcal{Y}.
\end{aligned} \tag{7}$$

The corresponding augmented Lagrangian function of Eq. (7) is

$$\begin{aligned}
&\mathcal{L}_\rho(\mathcal{Y}, \mathcal{Z}, E, S, \omega; \{\Theta^{(v)}\}, \Pi) = \|\mathcal{Z}\|_{\circledast} \\
&\quad + \sum_{v=1}^{V} \Big(\lambda_1 \|E^{(v)}\|_{2,1} + \lambda_2 tr\big(Y^{(v)} L^{(v)} Y^{(v)^T}\big) \\
&\quad + \lambda_3 \omega_v \|Y^{(v)} - S\|_F^2 \Big) + \gamma \|\omega\|_2^2 \\
&\quad + \tfrac{\rho}{2} \Big(\sum_{v=1}^{V} \|X^{(v)} - X^{(v)} Y^{(v)} - E^{(v)} + \tfrac{\Theta^{(v)}}{\rho}\|_F^2 + \|\mathcal{Z} - \mathcal{Y} + \tfrac{\Pi}{\rho}\|_F^2 \Big),
\end{aligned} \tag{8}$$

where $\{\Theta^{(v)} \in \mathbb{R}^{d_v \times n}\}$ and $\Pi \in \mathbb{R}^{n \times n \times V}$ are Lagrange multipliers with respect to two equality constraints, respectively. $\rho > 0$ is the penalty parameter. Borrowing the idea of alternative update strategy [11,19,42], Eq. (8) can be divided into the following six subproblems:

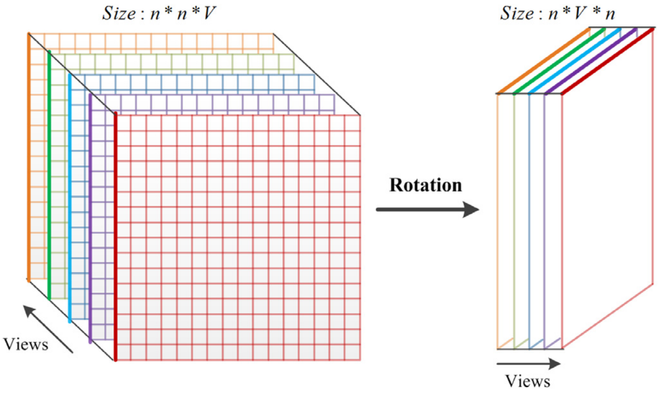


Fig. 3. Rotation of the representation tensor $\mathcal{Z}$.

Step 1 Update $\mathcal{Y}$: Given other variables in their previous iteration, we can update $\mathcal{Y}$ by solving the following problem:

$$\begin{aligned}
&\min_{\mathcal{Y}} \sum_{v=1}^{V} \Big(\lambda_2 tr\big(Y^{(v)} L^{(v)} Y^{(v)^T}\big) + \lambda_3 \omega_{t,v} \|Y^{(v)} - S_t\|_F^2 \Big) \\
&\quad + \tfrac{\rho_t}{2} \Big(\sum_{v=1}^{V} \|X^{(v)} - X^{(v)} Y^{(v)} - E_t^{(v)} + \tfrac{\Theta_t^{(v)}}{\rho_t}\|_F^2 + \|\mathcal{Z}_t - \mathcal{Y} + \tfrac{\Pi_t}{\rho_t}\|_F^2 \Big).
\end{aligned} \tag{9}$$

Eq. (9) can be separated into V independent minimization problems and the vth minimization problem is

$$\begin{aligned}
&\min_{Y^{(v)}} \lambda_2 tr\big(Y^{(v)} L^{(v)} Y^{(v)^T}\big) + \lambda_3 \omega_{t,v} \|Y^{(v)} \\
&\quad - S_t\|_F^2 + \tfrac{\rho_t}{2} \Big(\|X^{(v)} Y^{(v)} - A_t^{(v)}\|_F^2 + \|Y^{(v)} - B_t^{(v)}\|_F^2 \Big),
\end{aligned} \tag{10}$$

where $A_t^{(v)} = X^{(v)} - E_t^{(v)} + \frac{\Theta_t^{(v)}}{\rho_t}$ and $B_t^{(v)} = Z_t^{(v)} + \frac{\Pi_t^{(v)}}{\rho_t}$. By setting the derivative of Eq. (10) with respect to $Y^{(v)}$ to zero, we can yield a Sylvester equation $M * Y^{(v)} + Y^{(v)} * N = C$, where $M = (2\lambda_3 \omega_{t,v} + \rho_t) I + \rho_t X^{(v)^T} X^{(v)}$, $N = \lambda_2 (L^{(v)} + L^{(v)^T})$, and $C = 2\lambda_3 \omega_{t,v} S_t + \rho_t \Big(X^{(v)^T} A_t^v + B_t^{(v)} \Big)$. Then, the optimal solution of $Y_{t+1}^{(v)}$ can be obtained by the off-the-shelf solver, such as the Matlab function *lyap*.

Step 2 Update $\mathcal{Z}$: Fixing other variables, $\mathcal{Z}$ can be updated by solving

$$\min_{\mathcal{Z}} \frac{1}{\rho_t} \|\mathcal{Z}\|_{\circledast} + \frac{1}{2} \|\mathcal{Z} - \mathcal{F}_t\|_F^2, \tag{11}$$

where $\mathcal{F}_t = \mathcal{Y}_{t+1} - \Pi_t / \rho_t$. Note that we need to rotate $\mathcal{Z}$ from size $n \times n \times V$ to $n \times V \times n$ as shown in Fig. 3. This is because of two reasons: (1) as in Eq. (3), t-SVD-TNN performs the SVD on each frontal slice of $\hat{\mathcal{Z}}$, leading to capturing of the "spatial-shifting" correlation [34,38]. This means that t-SVD-TNN preserves only the low-rank property of intra-view. However, we hope to capture the low-rank property of inter-views. (2) the rotation operation can significantly reduce the computation cost [12].

The closed-form solution of Eq. (11) can be obtained by the tensor tubal-shrinkage operator [12,43]:

$$\mathcal{Z}_{t+1} = \mathcal{C}_{\frac{V}{\rho}}(\mathcal{F}_t) = \mathcal{U} * \mathcal{C}_{\frac{V}{\rho_t}}(\mathcal{S}) * \mathcal{V}^T, \tag{12}$$

where $\mathcal{F}_t = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$, and $\mathcal{C}_{\frac{V}{\rho_t}}(\mathcal{S}) = \mathcal{S} * \mathcal{J}$, in which $\mathcal{J}$ is an f-diagonal tensor whose diagonal element in the Fourier domain is $\mathcal{J}(i, i, k) = \max\{1 - \frac{V/\rho_t}{\mathcal{S}(i,i,k)}, 0\}$. The details on the update of $\mathbf{Z}_{t+1}$ are summarized in Algorithm 1.

Step 3 Update E: Minimizing the augmented Lagrangian function in Eq. (8) with respect to E, we have

Algorithm 1 : Update of $\mathcal{Z}$ based on t-SVD.

```
Input: tensor: $\mathcal{F}_t \in \mathbb{R}^{n\times V\times n}$; parameter: $\tau = \frac{V}{\rho_t}$;
$\hat{\mathcal{F}}_t = \textbf{fft}(\mathcal{X}, [], 3)$;
for $k = 1$ to $n$ do
    $[\mathcal{U}^{(k)}, \mathcal{S}^{(k)}, \mathcal{V}^{(k)}]$ = SVD$(\hat{\mathcal{F}}_t^{(k)})$;
    $\mathcal{J}^{(k)} = diag(\max\{1 - \frac{\tau}{\mathcal{S}(i,i,k)}, 0\}), i = 1, \cdots, V$;
    $\Sigma^{(k)} = \mathcal{S}^{(k)}\mathcal{J}^{(k)}$;
    $\hat{\mathcal{Z}}^{(k)} = \mathcal{U}^{(k)}\Sigma^{(k)}\mathcal{V}^{(k)^T}$;
end for
$\mathcal{Z}_{t+1} = \textbf{ifft}(\hat{\mathcal{Z}}, [], 3)$;
Output: tensor $\mathcal{Z}_{t+1}$.
```

$$
\begin{aligned}
E_{t+1} &= \underset{E}{\operatorname{argmin}} \sum_{v=1}^{V} \lambda_1 \|E^{(v)}\|_{2,1} + \frac{\rho_t}{2}\|E^{(v)} - T_t^{(v)}\|_F^2 \\
&= \underset{E}{\operatorname{argmin}} \frac{\lambda_1}{\rho_t}\|E\|_{2,1} + \frac{1}{2}\|E - T_t\|_F^2, \qquad (13)
\end{aligned}
$$

where $T_t^{(v)} = X^{(v)} - X^{(v)}Y_{t+1}^{(v)} + \frac{\Theta_t^{(v)}}{\rho_t}$ and $T_t = [T_t^{(1)}; T_t^{(2)}; \cdots; T_t^{(V)}]$. The jth column of E_{t+1} can be obtained by

$$
E_{t+1}(:, j) = \begin{cases} \frac{\|T_t(:, j)\|_2 - \frac{\lambda_1}{\rho_t}}{\|T_t(:, j)\|_2} T_t(:, j), & \text{if } \frac{\lambda_1}{\rho_t} < \|T_t(:, j)\|_2; \\ 0, & \text{otherwise.} \end{cases} \qquad (14)
$$

Step 4 Update S: To obtain the optimal solution S_{t+1}, we can minimize the augmented Lagrangian function in Eq. (8) with respect to S as

$$
\min_S \sum_{v=1}^{V} \omega_{t,v}\|Y_{t+1}^{(v)} - S\|_F^2. \qquad (15)
$$

We also set the derivative of Eq. (15) with respect to S to zero. The closed-form solution S_{t+1} is

$$
S_{t+1} = \frac{\sum_v \omega_{t,v} Y_{t+1}^{(v)}}{\sum_v \omega_{t,v}} = \sum_v \omega_{t,v} Y_{t+1}^{(v)}, \qquad (16)
$$

which is based on the constraint $\sum_v \omega_{t,v} = 1$.

Step 5 Update ω: The optimization of ω is transformed into the following problem

$$
\min_\omega \sum_{v=1}^{V} \omega_v a_t^{(v)} + \gamma\|\omega\|_2^2, \quad s.t.\ \omega \geq 0,\ \Sigma_v\omega_v = 1, \qquad (17)
$$

where $a_t^{(v)} = \|Y_{t+1}^{(v)} - S_{t+1}\|_F^2$. $\gamma\|\omega\|_2^2$ is used to smoothen the weight distribution and avoid the futile solution [22]. Then, Eq. (17) can be rewritten into the following quadratic programming formulation

$$
\min_\omega \|\omega + \frac{a_t}{2\gamma}\|_2^2, \quad s.t.\ \omega \geq 0,\ \Sigma_v\omega_v = 1. \qquad (18)
$$

The above formula can be efficiently solved by any off-the-shelf quadratic programming solver, such as *quadprog*.

Step 6 Update $\{\Theta^{(v)}\}, \Pi$, and ρ: The Lagrangian multipliers $\{\Theta^{(v)}\}, \Pi$ and the penalty parameter ρ can be updated by

$$
\begin{aligned}
&\Theta_{t+1}^{(v)} = \Theta_t^{(v)} + \rho_t(X^{(v)} - X^{(v)}Y_{t+1}^{(v)} - E_{t+1}^{(v)}); \\
&\Pi_{t+1} = \Pi_t + \rho_t(\mathcal{Z}_{t+1} - \mathcal{Y}_{t+1}); \qquad (19) \\
&\rho_{t+1} = \min\{\beta * \rho_t, \rho_{max}\},
\end{aligned}
$$

where $\beta > 1$ is to facilitate the convergence speed [44]. $\rho_{\max}$ is the maximum value of the penalty parameter ρ. The whole procedure of solving Eq. (7) is summarized in Algorithm 2, in which the stopping criterion is defined as follows:

$$
\max\begin{Bmatrix} \|X^{(v)} - X^{(v)}Y_{t+1}^{(v)} - E_{t+1}^{(v)}\|_\infty, v = 1, \cdots, V \\ \|\mathcal{Z}_{t+1} - \mathcal{Y}_{t+1}\|_\infty \end{Bmatrix} \leq tol, \qquad (20)
$$

where $tol > 0$ is a pre-defined tolerance. Once the affinity matrix S is obtained by GLTA (Algorithm 2), the spectral clustering algorithm [16] is carried out to yield the final clustering results.

Algorithm 2 : GLTA for multi-view subspace clustering.

```
Input: multi-view features: $\{X^{(v)}\}$; parameters: $\lambda_1, \lambda_2, \lambda_3, \gamma = 10$;
    nearest neighbors number 5; graph Laplacian matrices $\{L^{(v)}\}$;
    cluster number $K$;
Initialize: $\mathcal{Y}_1, \mathcal{Z}_1, E_1, S_1, \Theta_1, \Pi_1$ initialized to $\mathbf{0}$; weight $\omega_{1,v} = \frac{1}{V}$;
    $\rho_1 = 10^{-3}, \beta = 1.5, \epsilon = 10^{-7}, t = 1$;
while not converged do
    for $v = 1$ to $V$ do
        Update $Y_{t+1}^{(v)}$ according to Eq. (10);
    end for
    Update $\mathcal{Z}_{t+1}$ according to Algorithm 1;
    Update $E_{t+1}$ according to Eq. (14);
    Update $S_{t+1}$ according to Eq. (16);
    Update $\omega_{t+1}$ according to Eq. (18);
    Update $\Theta_{t+1}^{(v)}, \Pi_{t+1}$, and $\rho_{t+1}$ according to Eq. (19);
    Check the convergence condition in Eq. (20);
end while
Output: Affinity matrix $S_{t+1}$.
```

4.3. Computation complexity

The computation cost of Algorithm 2 is dominated by updating $\mathcal{Y}$, $\mathcal{Z}$, and E. For Step 1, the computation cost of solving the Sylvester equation is $O(n^3)$. For Step 2, updating $\mathcal{Z}$ needs $\mathcal{O}(2Vn^2\log(n))$ operations to calculate 3D FFT and inverse FFT, and $\mathcal{O}(V^2n^2)$ operations for performing SVD on V number of $n \times V$ matrices. For Step 3, it costs $\mathcal{O}(Vn^2)$ operations. As for the remaining steps, their computation costs can be ignored since they contain only the basic operations, such as matrix addition, subtraction, and multiplication. Thus, the computation complexity of Algorithm 2 is $\mathcal{O}\big(T(Vn^3 + 2Vn^2log(n) + V^2n^2)\big)$, where T is the number of iterations. As shown in Section 5.4, the proposed GLTA can converge within 30 ~ 45 iterations.

5. Experiments results

To verify the effectiveness of the proposed GLTA, in this section, we first conduct experiments to compare with twelve state-of-the-art clustering methods. Seven challenging datasets from three different application areas are selected as the testing data. To provide a comprehensive study of the proposed GLTA, we analyze GLTA with respect to three important parameters and report the empirical convergence of GLTA.

5.1. Datasets

Following [11,12,22], we evaluate the performance of GLTA on seven challenging multi-view datasets, including:

- **BBC4view dataset** and **BBCSport dataset**[1]: BBC4view and BBCSport are news stories datasets. They contains 685 and 544 documents from BBC Sport website about sports news on 5 topics, respectively. For each document, four different types of features are extracted in BBC4view while two different types of features are extracted in BBCSport.

[1] http://mlg.ucd.ie/datasets/segment.html

Table 1
Summary of seven challenging multi-view databases.

Category	Dataset	Instance	View	cluster
News stories	BBC4view	685	4	5
	BBCSport	544	2	5
	3Sources	169	3	6
Scene	MSRC-V1	210	5	7
	Scene-15	4485	3	15
	MITIndoor-67	5360	4	67
Generic object	COIL-20	1440	3	20

- **3Sources dataset**[2]: It is a news stories dataset, which was collected from three online news sources: BBC, Reuters, and Guardian. It contains 416 distinct news stories from 6 classes. Of them, 169 news documents are reported in all three sources and each source serves as one view.
- **MSRC-V1 dataset**: It contains 210 images in 7 classes, including tree, building, airplane, cow, face, car, and bicycle. Following [24], five-view features, including 24-D (dimension, D) colour moment (CM), 576-D histogram of oriented gradients (HOG), 512-D GIST, 254-D CENTRIST feature, and 256-D local binary pattern (LBP) are extracted.
- **Scene-15 dataset** [45]: It contains 4485 outdoor and indoor scene images from 15 categories. Following [12], three kinds of image features, including 1800-D PHOW, 1180-D PRI-CoLBP, and 1240-D CENTRIST are extracted to represent Scene-15.
- **MITIndoor-67 dataset** [46]: It consists of 15 thousand indoor pictures spanning 67 different categories. We select one training subset including 5360 images for clustering. As in [12], except three features used in Scene-15, one handcrafted feature from VGG-VD [47] is subtracted to serve as a new view to pursuit better performance.
- **COIL-20 dataset**[3]: There are 20 object categories and 1440 generic object images with 32 × 32 pixels. Similar to [12], we also extract three view features, including 1024-D intensity, 3304-D LBP, and 6750-D Gabor.

The statistics of these datasets are summarized in Table 1.

5.2. Compared methods and evaluation measures

We compare GLTA with the following state-of-the-art methods, including **SSC$_{\mathbf{best}}$**[4] [2]: single-view clustering via the l_1-norm regularized representation matrix construction; **LRR$_{\mathbf{best}}$**[5] [3]: single-view clustering via the nuclear norm regularized representation matrix construction; **RSS$_{\mathbf{best}}$**[6] [30]: single-view clustering via simultaneously learning data representations and their affinity matrix; **MLAP**[7] [25]: MVC by concatenating subspace representations of different views and imposing low-rank constraint to explore the complementarity; **DiMSC**[8] [19]: MVC with the Hilbert-Schmidt Independence criterion; **LT-MSC**[9] [11]: MVC with low-rank tensor constraint; **MVCC**[10] [22]: MVC via concept factorization with local manifold regularization; **ECMSC**[11] [33]: exclusivity-consistency regularized MVC; **MLAN**[12] [48]: MVC with adaptive neighbors; **t-SVD-MSC**[13] [12]: MVC via tensor multi-rank minimization; **MLRSSC**[14] [4]: MVC via low-rank sparse subspace clustering; **MSC_IAS**[15] [17] : MVC with intactness-aware similarity. The first three methods belong to single-view clustering baselines while others belong to multi-view clustering ones. We choose these methods due to their popularity and code availability. We also follow their experiment settings for fair comparison. Moreover, the deep feature is imposed on MITIndoor-67 dataset. We also compare the proposed GLTA with GSNMF-CNN [49] in Table 8. For **SSC$_{\mathbf{best}}$**, **LRR$_{\mathbf{best}}$**, and **RSS$_{\mathbf{best}}$**, each feature is used independently and the best clustering result is reported. For a full comparison, we also perform SSC, LRR, and RSS with the joint view feature which is concatenated by all features. They are denoted as SSC$_{\mathbf{Con}}$, LRR$_{\mathbf{Con}}$, and RSS$_{\mathbf{Con}}$, respectively. Since there exists one random parameter in MLAN, we run MLAN 10 trials and report the best clustering result. For DiMSC, LT-MSC, t-SVD-MSC, MLRSSC, and MSC_IAS, they all first learn the representation matrix or tensor, and then construct the affinity matrix. For all methods except MLAN, the spectral clustering algorithm [16] is performed to obtain the clustering result. Our previous conference paper [36] used the Tucker decomposition to encode the low-rank property, denoted as GLTA_Tucker.

Following [11,12], we exploit six popular clustering measures [50], *i.e.*, accuracy (ACC), normalized mutual information (NMI), adjusted rank index (AR), F-score, Precision, and Recall, to evaluate the clustering performance. One can refer to [12] for more details of these six measures. Generally, the higher values these six measures have, the better the clustering quality is. Since the spectral clustering is based on K-means for all methods and different initializations may yield different results, we run 10 trials for each experiment and report their average performance with standard deviations.

5.3. Clustering performance comparison

All clustering results on seven benchmark datasets are reported in Tables 2–8. The best results for each index are highlighted in boldface and the second-best results are underlined.

We reach the following observations from these experiment results:

- In most cases, the performance of GLTA is better than or comparable to those of all competing methods, especially on BBC4View, Scene-15, MITIndoor-67, and COIL-20 datasets. GLTA with t-SVD-NN outperforms GLTA_Tucker in all cases. This indicates that the singular value decomposition-based tensor nuclear norm may be the better candidate for the low-rank property of the representation tensor over the Tucker decomposition. The improvement of the proposed GLTA is around 16.7, 11.6, 19.6, 18.1, 22.7, and 12.9 percentage points with respect to six measures over the second-best method t-SVD-MSC on Scene-15 dataset, and around 23.8, 22.3, 36.1, 35.6, 34.9, and 36.3 percentage points on MITIndoor-67 dataset, respectively. The main reason is that DiMSC, LT-MSC, t-SVD-MSC-MLRSSC, and MSC_IAS construct the representation matrix or tensor and affinity matrix in two separate steps without the consideration of the various contributions of different features. However, the proposed GLTA learns the representation tensor and affinity matrix in a synchronous way such that the high dependence between them can be well exploited. More importantly, the promising performance of GLTA also benefits from the preservation of the local geometrical structures;

[2] http://mlg.ucd.ie/datasets/3sources.html

[3] http://www.cs.columbia.edu/CAVE/software/softlib/

[4] http://www.ccis.neu.edu/home/eelhami/codes.htm

[5] https://sites.google.com/site/guangcanliu/

[6] https://sites.google.com/view/xjguo

[7] https://github.com/canyilu/LibADMM/tree/master/algorithms

[8] http://cs.tju.edu.cn/faculty/zhangchangqing/code.html

[9] http://cs.tju.edu.cn/faculty/zhangchangqing/code.html

[10] https://github.com/vast-wang/Clustering

[11] http://www.cbsr.ia.ac.cn/users/xiaobowang/codes/Demo_ECMSC.zip

[12] http://www.escience.cn/people/fpnie/papers.html

[13] https://www.researchgate.net/profile/Yuan_Xie4/publications

[14] https://github.com/mbrbic/Multi-view-LRSSC

[15] http://www.cbsr.ia.ac.cn/users/xiaobowang/codes/MSC_IAS_Released.zip

Table 2
Clustering results (mean ± standard deviation) on BBC4view.

Method	ACC	NMI	AR	F-score	Precision	Recall
$\text{SSC}_{\mathbf{best}}$	0.660 ± 0.002	0.494 ± 0.005	0.470 ± 0.001	0.599 ± 0.001	0.578 ± 0.001	0.622 ± 0.001
$\text{SSC}_{\mathbf{Con}}$	0.848 ± 0.001	0.667 ± 0.002	0.702 ± 0.002	0.770 ± 0.002	0.787 ± 0.002	0.754 ± 0.002
$\text{LRR}_{\mathbf{best}}$	0.802 ± 0.000	0.568 ± 0.000	0.621 ± 0.000	0.712 ± 0.000	0.697 ± 0.000	0.727 ± 0.000
$\text{LRR}_{\mathbf{Con}}$	0.804 ± 0.000	0.611 ± 0.000	0.609 ± 0.000	0.700 ± 0.000	0.710 ± 0.000	0.690 ± 0.000
$\text{RSS}_{\mathbf{best}}$	0.837 ± 0.000	0.621 ± 0.000	0.665 ± 0.000	0.747 ± 0.000	0.720 ± 0.000	0.775 ± 0.000
$\text{RSS}_{\mathbf{Con}}$	0.877 ± 0.001	0.738 ± 0.002	0.758 ± 0.002	0.812 ± 0.001	0.834 ± 0.001	0.792 ± 0.002
MLAP	0.872 ± 0.000	0.725 ± 0.000	0.751 ± 0.000	0.808 ± 0.000	0.824 ± 0.000	0.793 ± 0.000
DiMSC	0.892 ± 0.001	0.728 ± 0.002	0.752 ± 0.002	0.810 ± 0.002	0.811 ± 0.002	0.810 ± 0.002
LT-MSC	0.591 ± 0.000	0.442 ± 0.005	0.400 ± 0.001	0.546 ± 0.000	0.525 ± 0.000	0.570 ± 0.001
MVCC	0.745 ± 0.001	0.587 ± 0.001	0.550 ± 0.000	0.656 ± 0.001	0.654 ± 0.001	0.658 ± 0.000
ECMSC	0.308 ± 0.028	0.047 ± 0.009	0.008 ± 0.018	0.322 ± 0.017	0.239 ± 0.009	0.497 ± 0.064
MLAN	0.853 ± 0.007	0.698 ± 0.010	0.716 ± 0.005	0.783 ± 0.004	0.776 ± 0.003	0.790 ± 0.004
t-SVD-MSC	0.858 ± 0.001	0.685 ± 0.002	0.725 ± 0.002	0.789 ± 0.001	0.800 ± 0.001	0.778 ± 0.002
MLRSSC	0.888 ± 0.074	0.761 ± 0.036	0.788 ± 0.073	0.837 ± 0.056	0.845 ± 0.053	0.830 ± 0.061
MSC_IAS	0.820 ± 0.001	0.632 ± 0.001	0.647 ± 0.002	0.728 ± 0.001	0.741 ± 0.001	0.715 ± 0.002
GLTA_Tucker	0.910 ± 0.000	0.771 ± 0.000	0.810 ± 0.000	0.854 ± 0.000	0.864 ± 0.000	0.845 ± 0.000
GLTA	**0.996** ± 0.000	**0.983** ± 0.000	**0.990** ± 0.000	**0.993** ± 0.000	**0.996** ± 0.000	**0.990** ± 0.000

Bold fonts denote the best performance; underlined ones represent the second-best results in all tables.

Table 3
Clustering results (mean ± standard deviation) on BBCSport.

Method	ACC	NMI	AR	F-score	Precision	Recall
$\text{SSC}_{\mathbf{best}}$	0.627 ± 0.003	0.534 ± 0.008	0.364 ± 0.007	0.565 ± 0.005	0.427 ± 0.004	0.834 ± 0.004
$\text{SSC}_{\mathbf{Con}}$	0.666 ± 0.011	0.590 ± 0.024	0.440 ± 0.088	0.609 ± 0.046	0.494 ± 0.062	0.819 ± 0.061
$\text{LRR}_{\mathbf{best}}$	0.836 ± 0.001	0.698 ± 0.002	0.705 ± 0.001	0.776 ± 0.001	0.768 ± 0.001	0.784 ± 0.001
$\text{LRR}_{\mathbf{Con}}$	0.853 ± 0.000	0.738 ± 0.000	0.760 ± 0.000	0.818 ± 0.000	0.807 ± 0.000	0.830 ± 0.000
$\text{RSS}_{\mathbf{best}}$	0.878 ± 0.000	0.714 ± 0.000	0.717 ± 0.000	0.784 ± 0.000	0.787 ± 0.000	0.782 ± 0.000
$\text{RSS}_{\mathbf{Con}}$	0.870 ± 0.001	0.731 ± 0.001	0.758 ± 0.001	0.815 ± 0.001	0.822 ± 0.001	0.809 ± 0.001
MLAP	0.868 ± 0.001	0.763 ± 0.003	0.791 ± 0.003	0.842 ± 0.002	0.827 ± 0.002	0.858 ± 0.003
DiMSC	0.922 ± 0.000	0.785 ± 0.000	0.813 ± 0.000	0.858 ± 0.000	0.846 ± 0.000	0.872 ± 0.000
LT-MSC	0.460 ± 0.046	0.222 ± 0.028	0.167 ± 0.043	0.428 ± 0.014	0.328 ± 0.028	0.629 ± 0.053
MVCC	0.928 ± 0.000	0.816 ± 0.000	0.831 ± 0.000	0.870 ± 0.000	0.889 ± 0.000	0.853 ± 0.000
ECMSC	0.285 ± 0.014	0.027 ± 0.013	0.009 ± 0.011	0.267 ± 0.020	0.244 ± 0.007	0.297 ± 0.045
MLAN	0.721 ± 0.000	0.779 ± 0.000	0.591 ± 0.000	0.714 ± 0.000	0.567 ± 0.000	0.962 ± 0.000
t-SVD-MSC	0.879 ± 0.000	0.765 ± 0.000	0.784 ± 0.000	0.834 ± 0.000	0.863 ± 0.000	0.807 ± 0.000
MLRSSC	0.815 ± 0.020	0.681 ± 0.005	0.678 ± 0.007	0.753 ± 0.004	0.775 ± 0.015	0.732 ± 0.007
MSC_IAS	0.948 ± 0.000	0.854 ± 0.000	0.861 ± 0.000	0.894 ± 0.000	0.892 ± 0.000	0.897 ± 0.000
GLTA_Tucker	0.939 ± 0.000	0.825 ± 0.000	0.849 ± 0.000	0.885 ± 0.000	0.890 ± 0.000	0.880 ± 0.000
GLTA	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000

Table 4
Clustering results (mean ± standard deviation) on 3Sources.

Method	ACC	NMI	AR	F-score	Precision	Recall
$\text{SSC}_{\mathbf{best}}$	0.762 ± 0.003	0.694 ± 0.003	0.658 ± 0.004	0.743 ± 0.003	0.769 ± 0.001	0.719 ± 0.005
$\text{SSC}_{\mathbf{Con}}$	0.670 ± 0.006	0.632 ± 0.009	0.511 ± 0.009	0.643 ± 0.007	0.556 ± 0.004	0.762 ± 0.014
$\text{LRR}_{\mathbf{best}}$	0.647 ± 0.033	0.542 ± 0.018	0.486 ± 0.028	0.608 ± 0.033	0.594 ± 0.031	0.636 ± 0.096
$\text{LRR}_{\mathbf{Con}}$	0.607 ± 0.019	0.605 ± 0.016	0.440 ± 0.026	0.554 ± 0.021	0.635 ± 0.022	0.491 ± 0.019
$\text{RSS}_{\mathbf{best}}$	0.722 ± 0.000	0.601 ± 0.000	0.533 ± 0.000	0.634 ± 0.000	0.679 ± 0.000	0.595 ± 0.000
$\text{RSS}_{\mathbf{Con}}$	0.731 ± 0.007	0.693 ± 0.006	0.591 ± 0.013	0.678 ± 0.010	0.738 ± 0.016	0.627 ± 0.006
MLAP	0.805 ± 0.000	**0.756** ± 0.000	0.688 ± 0.000	0.762 ± 0.000	0.751 ± 0.000	0.773 ± 0.000
DiMSC	0.795 ± 0.004	0.727 ± 0.010	0.661 ± 0.005	0.748 ± 0.004	0.711 ± 0.005	0.788 ± 0.003
LT-MSC	0.781 ± 0.000	0.698 ± 0.003	0.651 ± 0.003	0.734 ± 0.002	0.716 ± 0.008	0.754 ± 0.005
MVCC	0.761 ± 0.016	0.698 ± 0.016	0.626 ± 0.010	0.731 ± 0.008	0.607 ± 0.009	**0.916** ± 0.008
ECMSC	0.346 ± 0.025	0.132 ± 0.029	0.011 ± 0.031	0.295 ± 0.013	0.240 ± 0.019	0.391 ± 0.043
MLAN	0.775 ± 0.015	0.676 ± 0.005	0.580 ± 0.008	0.666 ± 0.007	0.756 ± 0.003	0.594 ± 0.009
t-SVD-MSC	0.781 ± 0.000	0.678 ± 0.000	0.658 ± 0.000	0.745 ± 0.000	0.683 ± 0.000	0.818 ± 0.000
MLRSSC	0.697 ± 0.034	0.604 ± 0.012	0.562 ± 0.041	0.660 ± 0.030	0.690 ± 0.050	0.633 ± 0.025
MSC_IAS	0.797 ± 0.017	0.641 ± 0.009	0.576 ± 0.026	0.666 ± 0.022	0.729 ± 0.014	0.613 ± 0.028
GLTA_Tucker	0.846 ± 0.000	0.728 ± 0.000	0.665 ± 0.000	0.736 ± 0.000	0.805 ± 0.000	0.678 ± 0.000
GLTA	**0.859** ± 0.008	0.753 ± 0.015	**0.713** ± 0.014	**0.775** ± 0.011	**0.827** ± 0.009	0.730 ± 0.013

- In general, multi-view clustering approaches achieve better clustering performance than the single-view clustering approaches SSC_{best}, LRR_{best}, and RSS_{best}. This is mainly because single-view clustering methods focus on specific view feature while the high-order cross information among multiple views is well captured by these multi-view clustering approaches;
- LT-MSC achieves unsatisfactory results on the first two datasets which may come from the fact that the unfolding-based tensor nuclear norm is a loose surrogate of Tucker rank. Moreover, t-SVD-MSC has achieved better performance than LT-MSC. The main reason is that t-SVD-TNN can better uncover the global structure of the representation tensor than the unfolding-based tensor nuclear norm;
- MLAN performs worse than three single-view clustering methods on BBCSport and Scene-15 datasets. The main reason may

Table 5
Clustering results (mean ± standard deviation) on MSRC-V1.

Method	ACC	NMI	AR	F-score	Precision	Recall
$\mathrm{SSC}_{\mathbf{best}}$	0.791 ± 0.007	0.750 ± 0.005	0.651 ± 0.006	0.701 ± 0.005	0.670 ± 0.008	0.736 ± 0.003
$\mathrm{SSC}_{\mathbf{Con}}$	0.762 ± 0.000	0.748 ± 0.002	0.658 ± 0.001	0.707 ± 0.002	0.673 ± 0.002	0.748 ± 0.001
$\mathrm{LRR}_{\mathbf{best}}$	0.695 ± 0.000	0.590 ± 0.000	0.491 ± 0.000	0.562 ± 0.000	0.560 ± 0.000	0.564 ± 0.002
$\mathrm{LRR}_{\mathbf{Con}}$	0.694 ± 0.004	0.553 ± 0.009	0.470 ± 0.007	0.545 ± 0.006	0.535 ± 0.006	0.556 ± 0.007
$\mathrm{RSS}_{\mathbf{best}}$	0.751 ± 0.002	0.634 ± 0.003	0.538 ± 0.004	0.604 ± 0.004	0.587 ± 0.004	0.621 ± 0.003
$\mathrm{RSS}_{\mathbf{Con}}$	0.801 ± 0.040	0.692 ± 0.030	0.625 ± 0.047	0.678 ± 0.041	0.670 ± 0.040	0.686 ± 0.041
MLAP	0.857 ± 0.000	0.750 ± 0.000	0.704 ± 0.000	0.746 ± 0.000	0.741 ± 0.000	0.751 ± 0.000
DiMSC	0.759 ± 0.009	0.622 ± 0.015	0.548 ± 0.015	0.611 ± 0.013	0.606 ± 0.013	0.616 ± 0.012
LT-MSC	0.831 ± 0.003	0.743 ± 0.004	0.665 ± 0.004	0.712 ± 0.004	0.699 ± 0.004	0.725 ± 0.003
MVCC	0.622 ± 0.018	0.588 ± 0.013	0.458 ± 0.015	0.538 ± 0.014	0.510 ± 0.012	0.569 ± 0.020
ECMSC	0.795 ± 0.002	0.750 ± 0.002	0.681 ± 0.001	0.727 ± 0.001	0.705 ± 0.001	0.750 ± 0.001
MLAN	0.859 ± 0.003	0.751 ± 0.003	0.709 ± 0.004	0.750 ± 0.003	0.727 ± 0.004	0.776 ± 0.002
t-SVD-MSC	$\underline{0.991}$ ± 0.000	$\underline{0.982}$ ± 0.000	$\underline{0.978}$ ± 0.000	$\underline{0.981}$ ± 0.000	$\underline{0.980}$ ± 0.000	$\underline{0.982}$ ± 0.000
MLRSSC	0.521 ± 0.051	0.411 ± 0.041	0.285 ± 0.052	0.386 ± 0.044	0.379 ± 0.045	0.392 ± 0.042
MSC_IAS	0.909 ± 0.000	0.844 ± 0.000	0.802 ± 0.000	0.830 ± 0.000	0.820 ± 0.000	0.840 ± 0.000
GLTA_Tucker	0.878 ± 0.006	0.783 ± 0.009	0.737 ± 0.010	0.774 ± 0.010	0.763 ± 0.000	0.785 ± 0.009
GLTA	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000

Table 6
Clustering results (mean ± standard deviation) on COIL-20.

Method	ACC	NMI	AR	F-score	Precision	Recall
$\mathrm{SSC}_{\mathbf{best}}$	0.803 ± 0.022	0.935 ± 0.009	0.798 ± 0.022	0.809 ± 0.013	0.734 ± 0.027	0.804 ± 0.028
$\mathrm{SSC}_{\mathbf{Con}}$	0.851 ± 0.000	0.960 ± 0.000	0.833 ± 0.000	0.843 ± 0.000	0.757 ± 0.000	0.949 ± 0.000
$\mathrm{LRR}_{\mathbf{best}}$	0.761 ± 0.003	0.829 ± 0.006	0.720 ± 0.020	0.734 ± 0.006	0.717 ± 0.003	0.751 ± 0.002
$\mathrm{LRR}_{\mathbf{Con}}$	0.766 ± 0.020	0.866 ± 0.008	0.722 ± 0.013	0.737 ± 0.012	0.694 ± 0.024	0.787 ± 0.016
$\mathrm{RSS}_{\mathbf{best}}$	0.837 ± 0.012	0.930 ± 0.006	0.789 ± 0.005	0.800 ± 0.005	0.717 ± 0.012	0.897 ± 0.017
$\mathrm{RSS}_{\mathbf{Con}}$	0.757 ± 0.011	0.836 ± 0.008	0.711 ± 0.016	0.725 ± 0.016	0.717 ± 0.016	0.732 ± 0.015
MLAP	0.738 ± 0.020	0.825 ± 0.009	0.685 ± 0.023	0.701 ± 0.021	0.688 ± 0.027	0.715 ± 0.016
DiMSC	0.778 ± 0.022	0.846 ± 0.002	0.732 ± 0.005	0.745 ± 0.005	0.739 ± 0.007	0.751 ± 0.003
LT-MSC	0.804 ± 0.011	0.860 ± 0.002	0.748 ± 0.004	0.760 ± 0.007	0.741 ± 0.009	0.776 ± 0.006
MVCC	0.732 ± 0.018	0.845 ± 0.007	0.675 ± 0.022	0.692 ± 0.021	0.647 ± 0.034	0.744 ± 0.013
ECMSC	0.782 ± 0.001	0.942 ± 0.001	0.781 ± 0.001	0.794 ± 0.001	0.695 ± 0.002	0.925 ± 0.001
MLAN	0.862 ± 0.011	**0.961** ± 0.004	0.835 ± 0.006	0.844 ± 0.013	0.758 ± 0.008	**0.953** ± 0.007
t-SVD-MSC	0.830 ± 0.000	0.884 ± 0.005	0.786 ± 0.003	0.800 ± 0.004	0.785 ± 0.007	0.808 ± 0.001
MLRSSC	0.859 ± 0.007	$\underline{0.960}$ ± 0.001	0.835 ± 0.004	0.843 ± 0.003	0.758 ± 0.001	$\underline{0.952}$ ± 0.007
MSC_IAS	0.845 ± 0.009	0.958 ± 0.005	0.849 ± 0.010	0.839 ± 0.012	0.803 ± 0.008	0.910 ± 0.006
GLTA_Tucker	$\underline{0.878}$ ± 0.008	0.945 ± 0.001	$\underline{0.869}$ ± 0.007	$\underline{0.875}$ ± 0.007	$\underline{0.856}$ ± 0.013	0.895 ± 0.001
GLTA	**0.903** ± 0.006	0.946 ± 0.001	**0.891** ± 0.007	**0.897** ± 0.006	**0.893** ± 0.013	0.900 ± 0.001

Table 7
Clustering results (mean ± standard deviation) on Scene-15.

Method	ACC	NMI	AR	F-score	Precision	Recall
$\mathrm{SSC}_{\mathbf{best}}$	0.444 ± 0.003	0.470 ± 0.002	0.279 ± 0.001	0.337 ± 0.002	0.292 ± 0.001	0.397 ± 0.001
$\mathrm{SSC}_{\mathbf{Con}}$	0.436 ± 0.010	0.527 ± 0.003	0.317 ± 0.008	0.371 ± 0.007	0.324 ± 0.009	0.434 ± 0.013
$\mathrm{LRR}_{\mathbf{best}}$	0.445 ± 0.013	0.426 ± 0.018	0.272 ± 0.015	0.324 ± 0.010	0.316 ± 0.015	0.333 ± 0.015
$\mathrm{LRR}_{\mathbf{Con}}$	0.523 ± 0.001	0.532 ± 0.001	0.375 ± 0.002	0.418 ± 0.002	0.419 ± 0.001	0.418 ± 0.002
$\mathrm{RSS}_{\mathbf{best}}$	0.468 ± 0.008	0.441 ± 0.003	0.310 ± 0.004	0.357 ± 0.003	0.358 ± 0.003	0.356 ± 0.004
MLAP	0.568 ± 0.005	0.563 ± 0.002	0.405 ± 0.002	0.447 ± 0.002	0.439 ± 0.001	0.455 ± 0.003
DiMSC	0.300 ± 0.010	0.269 ± 0.009	0.117 ± 0.012	0.181 ± 0.010	0.173 ± 0.016	0.190 ± 0.010
LT-MSC	0.574 ± 0.009	0.571 ± 0.011	0.424 ± 0.010	0.465 ± 0.007	0.452 ± 0.003	0.479 ± 0.008
MVCC	0.469 ± 0.001	0.496 ± 0.002	0.318 ± 0.002	0.369 ± 0.001	0.342 ± 0.002	0.400 ± 0.001
ECMSC	0.457 ± 0.001	0.463 ± 0.002	0.303 ± 0.001	0.357 ± 0.001	0.318 ± 0.001	0.408 ± 0.001
MLAN	0.332 ± 0.000	0.475 ± 0.000	0.151 ± 0.000	0.248 ± 0.000	0.150 ± 0.000	0.731 ± 0.000
t-SVD-MSC	$\underline{0.812}$ ± 0.007	$\underline{0.858}$ ± 0.007	$\underline{0.771}$ ± 0.003	$\underline{0.788}$ ± 0.001	$\underline{0.743}$ ± 0.006	$\underline{0.839}$ ± 0.003
MLRSSC	0.484 ± 0.026	0.463 ± 0.011	0.313 ± 0.015	0.362 ± 0.014	0.355 ± 0.015	0.368 ± 0.013
MSC_IAS	0.583 ± 0.003	0.603 ± 0.003	0.429 ± 0.006	0.472 ± 0.006	0.438 ± 0.009	0.512 ± 0.013
GLTA	**0.979** ± 0.027	**0.974** ± 0.007	**0.967** ± 0.022	**0.969** ± 0.020	**0.970** ± 0.024	**0.968** ± 0.017

be that MLAN learns the affinity matrix directly from the raw data that may contain noise and outliers;

- The low-rank matrix-based multi-view subspace clustering methods, *i.e.*, MLRSSC and MSC_IAS have unstable performance. For example, they outperform almost competing methods on BBC4view, BBCSport and COIL-20 datasets but achieve worse performance than SSC and LRR on MITIndoor-67 dataset.

In summary, these experiment results indicate that learning the representation tensor and affinity matrix in a synchronous way has the potential to the improvement of the clustering performance.

5.4. Model analysis

In this section, we aim to present a comprehensive study of the proposed GLTA. We first analyze the parameter sensitivity and empirical convergence, and then explain why the proposed GLTA can obtain superiority over all competing methods.

(1) Parameter selection: We set the number of the nearest neighbors as 5 and $\gamma = 10$ for all experiments. Here, we investigate how to tune parameters in the proposed GLTA. Three free parameters λ_1, λ_2, and λ_3 in GLTA should be tuned. Specifically, they are

Table 8
Clustering results (mean ± standard deviation) on MITIndoor-67.

Method	ACC	NMI	AR	F-score	Precision	Recall
SSC$_{\mathbf{best}}$	0.475 ± 0.008	0.615 ± 0.003	0.332 ± 0.006	0.343 ± 0.006	0.314 ± 0.007	0.377 ± 0.007
SSC$_{\mathbf{Con}}$	0.411 ± 0.009	0.528 ± 0.003	0.258 ± 0.005	0.270 ± 0.005	0.255 ± 0.007	0.286 ± 0.005
LRR$_{\mathbf{best}}$	0.120 ± 0.004	0.226 ± 0.006	0.031 ± 0.007	0.045 ± 0.004	0.044 ± 0.006	0.047 ± 0.004
LRR$_{\mathbf{Con}}$	0.358 ± 0.010	0.492 ± 0.004	0.223 ± 0.005	0.234 ± 0.005	0.230 ± 0.005	0.239 ± 0.004
RSS$_{\mathbf{best}}$	0.490 ± 0.013	0.603 ± 0.005	0.338 ± 0.008	0.348 ± 0.008	0.337 ± 0.008	0.359 ± 0.008
DiMSC	0.246 ± 0.000	0.383 ± 0.003	0.128 ± 0.005	0.141 ± 0.004	0.138 ± 0.001	0.144 ± 0.002
LT-MSC	0.431 ± 0.002	0.546 ± 0.004	0.280 ± 0.008	0.290 ± 0.002	0.279 ± 0.006	0.306 ± 0.005
ECMSC	0.353 ± 0.002	0.489 ± 0.001	0.216 ± 0.002	0.228 ± 0.001	0.213 ± 0.001	0.247 ± 0.002
MLAN	0.468 ± 0.010	0.611 ± 0.003	0.312 ± 0.006	0.323 ± 0.006	0.299 ± 0.008	0.352 ± 0.003
GSNMF-CNN	0.517 ± 0.003	0.673 ± 0.003	0.264 ± 0.005	0.372 ± 0.002	0.367 ± 0.004	0.381 ± 0.001
t-SVD-MSC	$\underline{0.684}$ ± 0.005	$\underline{0.750}$ ± 0.007	$\underline{0.555}$ ± 0.005	$\underline{0.562}$ ± 0.008	$\underline{0.543}$ ± 0.005	$\underline{0.582}$ ± 0.004
MSC_IAS	0.333 ± 0.006	0.466 ± 0.002	0.176 ± 0.004	0.189 ± 0.004	0.174 ± 0.004	0.207 ± 0.004
GLTA	**0.922** ± 0.014	**0.973** ± 0.004	**0.916** ± 0.004	**0.918** ± 0.013	**0.892** ± 0.018	**0.945** ± 0.009

MLRSSC runs out of memory in current platform.

Table 9
Comparison among different view features by SSC [2] and LRR [3].

Dataset	SSC (ACC/NMI)				LRR (ACC/NMI)			
	View 1	View 2	View 3	View 4	View 1	View 2	View 3	View 4
BBC4view	0.660/0.494	0.414/0.238	0.542/0.259	0.415/0.236	0.802/0.568	0.769/0.525	0.791/0.550	0.740/0.497
3Sources	0.661/0.568	0.762/0.694	0.695/0.632		0.580/0.516	0.647/0.542	0.618/0.511	
BBCSport	0.589/0.534	0.627/0.534			0.836/0.698	0.816/0.630		

Table 10
Comparison of GLTA and its variants (ACC/NMI).

	ACC/NMI						
	BBC4view	BBCSport	3Sources	MSRC-V1	Scene15	COIL-20	Average
GLTA	0.996/0.983	1.000/1.000	0.859/0.753	1.000/1.000	0.979/0.974	0.903/0.946	0.9562/0.9427
GLTA-p1	0.972/0.909	0.959/0.905	0.749/0.720	0.879/0.828	0.912/0.918	0.891/0.940	0.8937/0.8700
GLTA-p2	0.417/0.371	0.998/0.994	0.291/0.083	1.000/1.000	0.885/0.889	0.835/0.905	0.7377/0.7070

Table 11
Complexity and average running time on all datasets (in seconds).

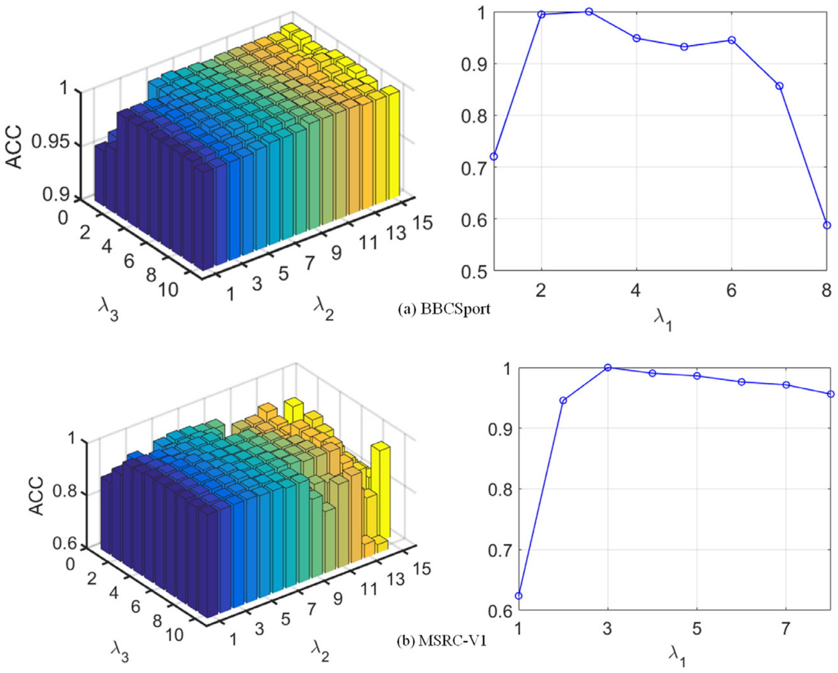
Data	MLAP	DiMSC	LT-MSC	MLAN	t-SVD-MSC	MLRSSC	MSC_IAS	GLTA
Complexity	$\mathcal{O}(Tn^3)$	$\mathcal{O}(TVn^3)$	$\mathcal{O}(TVn^3)$	$\mathcal{O}(dn^2+Tcn^2)$	$\mathcal{O}\Big(TVn^2\log(n)\Big)$	$\mathcal{O}(TVn^3)$	$\mathcal{O}(Tn^3)$	$\mathcal{O}\Big(TV\Big(n^3+n^2log(n)\Big)\Big)$
BBC4view	555.41	207.21	335.51	2.76	97.99	15.59	6.25	192.45
BBCSport	159.65	38.15	77.23	1.89	19.59	6.51	15.16	54.63
3Sources	43.03	4.89	23.45	1.01	8.72	2.59	3.47	10.24
MSRC-V1	29.37	4.73	20.15	1.44	5.96	1.12	3.39	11.12
COIL-20	1826.51	617.29	874.91	31.03	169.10	34.52	41.55	1689.22
Scene-15	13825.53	12449.36	7705.87	3318.62	3429.46	3592.46	185.45	16744.69
MITIndoor-67	33851.14	31254.21	20834.12	429.74	3404.86	-	254.89	25332.23

Table 12
Average running time on BBC4view with different value combinations (in seconds).

(λ_1, λ_2, λ_3)	(0.005,0.01,0.1)	(0.005,0.01,10)	(0.005,0.1,0.1)	(0.005,0.1,10)
Time	195.49	192.45	195.10	187.24
Iteration	46	45	46	44
(λ_1, λ_2, λ_3)	(0.1,0.01,0.1)	(0.1,0.01,10)	(0.1,0.1,0.1)	(0.1,0.1,10)
Time	165.84	162.22	172.12	171.21
Iteration	38	38	39	39

empirically selected from the sets of [0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.4, 0.5], [0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.4, 0.5, 1, 2, 5, 10, 50, 100, 500], and [0.01,0.1,0.5,1,3,5,7,10,50,100], respectively. Due to page limitation, we only show the ACC values of our GLTA with different combinations of λ_1, λ_2, and λ_3 on BBCSport and MSRC-V1 datasets in Fig. 4. It is well known that the error term may have less importance for the objective function [3]. Inspired by this observation, we first fix λ_1 as a relative small constant, and then perform GLTA with different combinations of λ_2 and λ_3 as shown in the left figures of Fig. 4. We can see that GLTA is not sensitive to parameter λ_2 and λ_3. Finally, we fix λ_2 and λ_3, and perform GLTA to investigate the influence of λ_1. We can see that when λ_1 is small, GLTA can yield promising results. Overall, the recommended parameters of GLTA are that λ_1, λ_2, and λ_3 can select from the interval [0.005, 0.2], [0.05,0.2], and [0.01,1], respectively.

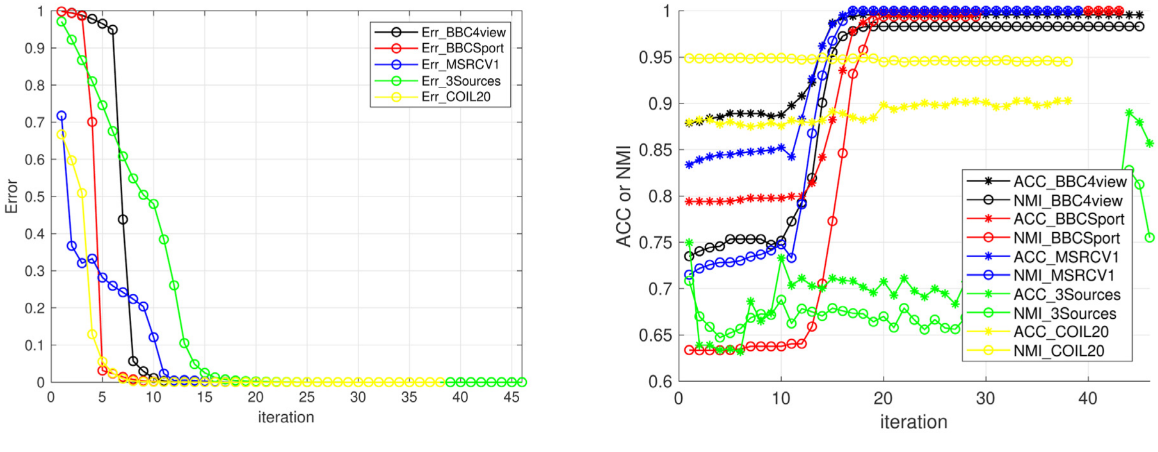
(2) Convergence analysis: It is intractable to derive the theoretical convergence proof of the proposed GLTA. Instead, we provide the empirical convergence analysis on four datasets in Fig. 5(a), in which the vertical axis denotes the error defined as $\sum_v \|X^{(v)} - X^{(v)}Y^{(v)^*} - E^{(v)^*}\|_F / \sum_v \|X^{(v)}\|_F$. After 15 iterations, the error yields



(a) BBCSport

(b) MSRC-V1

Fig. 4. ACC values of GLTA with different combinations of λ_2 and λ_3 by fixing λ_1 on (a) BBCSport and (b) MSRC-V1 datasets.



(a)

(b)

Fig. 5. (a) Empirical convergence versus iterations; (b) ACC and NMI versus iterations.

a stable value. This means that GLTA can converge within a few iterations. We also report the ACC and NMI values with each iteration in Fig. 5(b), since they can reflect the clustering performance to some extent. We can see that when the number of iterations increases, ACC and NMI values consistently increase until approaching the best values. This indicates indirectly that GLTA is convergent on these real datasets.

(3) The necessity of the feature weight: The clustering results by SSC [2] and LRR [3] on each view feature are reported in Table 9. We can see that for the same dataset, different features may yield various clustering results. For example, the values of ACC and NMI on BBC4view by SSC vary from 41.4 to 66.0 and 23.6 to 49.4 percentage points, respectively. For 3Sources, differences among three views by SSC with respect to ACC and NMI are 10.1 and 12.6 percentage points, respectively. In addition, on 3Sources and MSRC-v1 datasets, $\text{SSC}_{\textbf{Con}}$ and $\text{LRR}_{\textbf{Con}}$ perform worse than SSC and LRR. Therefore, we can draw a conclusion that different features have various contributions to clustering results. This is one of the fundamental motivations of this paper. Thus, it is of vital importance to fully consider the different contributions of different features in the multi-view clustering procedure.

(4) Ablation study:

In this section, we aim to investigate the ablation study of GLTA including the roles of local structures and the scheme of simultaneously learning the representation tensor and affinity matrix. From all above experimental results, we can see that only considering the low-rank tensor representation (such as, LT-MSC and t-SVD-MSC) or the local structures (such as, MLAN) cannot achieve satisfactory performance. In addition, existing methods, including DiMSC, LT-MSC, and t-SVD-MSC, learn the representation tensor, and then construct the affinity matrix. They fail to consider the various contributions of different features and the dependence between features. To address these issues, the proposed GLTA improves existing methods in two phases: (1) GLTA learns the representation tensor and affinity matrix simultaneously; (2) GLTA incorporates the local geometrical structures into one unified framework. To investigate the contributions of these above two factors individually, we conduct experiments by performing two tests.

Specifically, the first test sets $\lambda_2,\ \lambda_3 = 0$ and tunes other parameters, while the second test fixes $\lambda_3 = 0$.

The first test, denoted as GLTA-p1, sets parameters λ_2, λ_3 to zero to verify the contribution of Phase (1). In GLTA-p1, $\mathcal{Z}$ and S are learned simultaneously while the local structures are missing. The second test, called GLTA-p2, sets parameter λ_3 to zero to investigate the contribution of Phase (2). These two tests are performed on the BBC4view, BBCSport, 3Sources, MSRC-V1, Scene-15, and COIL-20 databases. The clustering results of GLTA, GLTA-p1, and GLTA-p2 are reported in Table 10. As can be seen, GLTA has achieved superior performance to GLTA-p1 and GLTA-p2 in all cases. In average, GLTA improves GLTA-p1 and GLTA-p2 at 21.85 and 6.25 percentage points with respect to the ACC value and at 23.57 and 7.27 percentage points in terms of the NMI value. These results directly verify that the superiority of GLTA, and indicate that constructing the representation tensor and affinity matrix in a synchronous way and preserving the local geometrical structures can significantly boost the clustering performance.

(5) Comparison of running time: The average running time of different multi-view clustering methods is shown in Table 11. All experiments are implemented in Matlab 2016a on a workstation with 3.50GHz CPU and 16GB RAM. MLAN and MSC_IAS have the shortest processing time among all methods, especially when handling the large-scale datasets. MLAP and DiMSC have the running time comparable with the proposed GLTA. The low-rank tensor-based multi-view clustering methods (including LT-MSC, t-SVD-MSC, and the proposed GLTA) have high computation cost while they have achieved better performance than other competing methods. The underlying reason is that LT-MSC, t-SVD-MSC, and GLTA find the correlation of the representation matrices in a global view via the low-rank tensor approximation. The main shortcoming of the proposed GLTA is the high computation complexity. There are two possible approaches to address this issue. Following [13], the first approach is to learn a flexible affinity matrix that may avoid solving a Sylvester equation. Using the tensor factorization strategy [51], the second one is to factorize the representation tensor into the product of two tensors with small sizes. This approach needs only matrix multiplications and does not compute the tensor singular value decomposition. Our future work will investigate how to use these two approaches to develop efficient and effective multi-view clustering methods (Table 12).

To further investigate the computational complexity of the proposed GLTA, we conduct experiments on BBC4view with different combinations of (λ_1, λ_2, λ_3). The results are shown in Table 12. We can see that different settings of parameters (λ_1, λ_2, λ_3) may slightly influence on the running time of GLTA.

6. Conclusion

In this paper, we developed a novel method for multi-view subspace clustering by learning graph regularized low-rank representation tensor and affinity matrix (GLTA) in a unified framework. GLTA can learn the low-rank representation tensor and affinity matrix simultaneously. The representation tensor is encoded by the t-SVD-based tensor nuclear norm and the local manifolds while the affinity matrix is constructed by assigning different weights to different view features. Extensive experiments on seven challenging datasets demonstrated that our GLTA outperforms the state-of-the-arts.

For the future exploration, the first direction is how to integrate the spectral clustering into the low-rank tensor representation-based methods to learn the common indicator matrix. The second one is, in some real applications like webpage clustering and disease diagnosing, some samples of different views may be missing. Thus, it is natural to consider how to extend the proposed method for incomplete multi-view clustering. The last one is to develop multi-view clustering methods with an unknown number of clusters.

Declaration of Competing Interest

The authors declare that they have no conflict of interest.

Acknowledgments

This work was funded by The Science and Technology Development Fund, Macau SAR (File no. 189/2017/A3), and by University of Macau (File no. MYRG2018-00136-FST).

References

[1] R. Vidal, Subspace clustering, IEEE Signal Process. Mag. 28 (2) (2011) 52–68.
[2] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
[3] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 171–184.
[4] M. Brbić, I. Kopriva, Multi-view low-rank sparse subspace clustering, Pattern Recognit. 73 (2018) 247–258.
[5] R. Vidal, P. Favaro, Low rank subspace clustering (LRSC), Pattern Recognit. Lett. 43 (2014) 47–61.
[6] W. Zhu, J. Lu, J. Zhou, Nonlinear subspace clustering for image clustering, Pattern Recognit. Lett. 107 (2018) 131–136.
[7] X. Cai, F. Nie, H. Huang, Multi-view k-means clustering on big data., in: Proc. Joint Conf. Artif. Intell., 2013, pp. 2598–2604.
[8] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: Proc. Neural Inf. Process. Syst., 2011, pp. 1413–1421.
[9] K. Chaudhuri, S.M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, in: Proc. Int. Conf. Mach. Learn., ACM, 2009, pp. 129–136.
[10] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, in: Proc. AAAI Conf. Artif. Intell., 2014, pp. 2149–2155.
[11] C. Zhang, H. Fu, S. Liu, G. Liu, X. Cao, Low-rank tensor constrained multiview subspace clustering, in: Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1582–1590.
[12] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, Y. Qu, On unifying multi-view self--representations for clustering by tensor multi-rank minimization, Int. J. Comput. Vis. 126 (11) (2018) 1157–1179.
[13] J. Wu, Z. Lin, H. Zha, Essential tensor learning for multi-view spectral clustering, IEEE Trans. Image Process. 28 (12) (2019) 5910–5922.
[14] X. Zhang, H. Sun, Z. Liu, Z. Ren, Q. Cui, Y. Li, Robust low-rank kernel multi-view subspace clustering based on the Schatten p-norm and correntropy, Inf. Sci. 477 (2019) 430–447.
[15] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, S. Yan, Robust and efficient subspace segmentation via least squares regression, in: Proc. Eur. Conf. Comput. Vis., Springer, 2012, pp. 347–360.
[16] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Proc. Neural Inf. Process. Syst., 2002, pp. 849–856.
[17] X. Wang, Z. Lei, X. Guo, C. Zhang, H. Shi, S.Z. Li, Multi-view subspace clustering with intactness-aware similarity, Pattern Recognit. 88 (2019) 50–63.
[18] W. Zhu, J. Lu, J. Zhou, Structured general and specific multi-view subspace clustering, Pattern Recognit. 93 (2019) 392–403.
[19] X. Cao, C. Zhang, H. Fu, S. Liu, H. Zhang, Diversity-induced multi-view subspace clustering, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 586–594.
[20] Z. Zhang, G. Ely, S. Aeron, N. Hao, M. Kilmer, Novel methods for multilinear data completion and de-noising based on tensor-SVD, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 3842–3849.
[21] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 5249–5257.
[22] H. Wang, Y. Yang, T. Li, Multi-view clustering via concept factorization with local manifold regularization, in: Proc. IEEE Int. Conf. Data Min., 2016, pp. 1245–1250.
[23] M. Yin, J. Gao, Z. Lin, Laplacian regularized low-rank representation and its applications, IEEE Trans. Pattern Anal. Mach. Intell. 38 (3) (2016) 504–517.
[24] F. Nie, J. Li, X. Li, Self-weighted multiview clustering with multiple graphs, in: Proc. Joint Conf. Artif. Intell., 2017, pp. 2564–2570.
[25] B. Cheng, G. Liu, J. Wang, Z. Huang, S. Yan, Multi-task low-rank affinity pursuit for image segmentation, in: Proc. IEEE Int. Conf. Comput. Vis., IEEE, 2011, pp. 2439–2446.
[26] X. Zhou, C. Yang, W. Yu, Moving object detection by detecting contiguous outliers in the low-rank representation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (3) (2012) 597–610.
[27] A. Jalali, Y. Chen, S. Sanghavi, H. Xu, Clustering partially observed graphs via convex optimization., in: Proc. Int. Conf. Mach. Learn., vol. 11, 2011, pp. 1001–1008.

[28] R. Henriques, C. Antunes, S.C. Madeira, A structured view on pattern mining-based biclustering, Pattern Recognit. 48 (12) (2015) 3941–3958.

[29] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, ACM SIGKDD Explor. Newsl. 6 (1) (2004) 90–105.

[30] X. Guo, Robust subspace segmentation by simultaneously learning data representations and their affinity matrix, in: Proc. Joint Conf. Artif. Intell., 2015, pp. 3547–3553.

[31] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, arXiv:1304.5634(2013).

[32] C. Lu, S. Yan, Z. Lin, Convex sparse spectral clustering: single-view to multi--view, IEEE Trans. Image Process. 25 (6) (2016) 2833–2843.

[33] X. Wang, X. Guo, Z. Lei, C. Zhang, S.Z. Li, Exclusivity-consistency regularized multi-view subspace clustering, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 923–931.

[34] Y. Chen, S. Wang, Y. Zhou, Tensor nuclear norm-based low-rank approximation with total variation regularization, IEEE J. Sel. Top. Signal Process. 12 (6) (2018) 1364–1377.

[35] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, D. Xu, Generalized latent multi--view subspace clustering, IEEE Trans. Pattern Anal. Mach. Intell. 42 (1) (2018) 86–99.

[36] Y. Chen, X. Xiao, Y. Zhou, Multi-view clustering via simultaneously learning graph regularized low-rank tensor representation and affinity matrix, in: Proc. Int. Conf. Multimedia Expo., IEEE, 2019, pp. 1348–1353.

[37] D. Goldfarb, Z. Qin, Robust low-rank tensor recovery: models and algorithms, SIAM J. Matrix Anal. Appl. 35 (1) (2014) 225–253.

[38] M.E. Kilmer, C.D. Martin, Factorization strategies for third-order tensors, Linear Algebra Appl. 435 (3) (2011) 641–658.

[39] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.

[40] Y. Chen, X. Xiao, Y. Zhou, Low-rank quaternion approximation for color image processing, IEEE Trans. Image Process. (2019) 1–14.

[41] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends® Mach.Learn. 3 (1) (2011) 1–122.

[42] Y. Chen, X. Xiao, Y. Zhou, Jointly learning kernel representation tensor and affinity matrix for multi-view clustering, IEEE Trans. Multimed. 1 (2019) 1–13, doi:10.1109/TMM.2019.2952984.

[43] W. Hu, D. Tao, W. Zhang, Y. Xie, Y. Yang, The twist tensor nuclear norm for video completion, IEEE Trans. Neural Netw. Learn. Syst. 28 (12) (2017) 2961–2973.

[44] Y. Chen, S. Wang, F. Zheng, Y. Cen, Graph-regularized least squares regression for multi-view subspace clustering, Knowl.-Based Syst. (2020) 105482, doi:10.1016/j.knosys.2020.105482.

[45] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., vol. 2, IEEE, 2005, pp. 524–531.

[46] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, 2009, pp. 413–420.

[47] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proc. Int. Conf. Learn. Representations, 2014.

[48] F. Nie, G. Cai, J. Li, X. Li, Auto-weighted multi-view learning for image clustering and semi-supervised classification, IEEE Trans. Image Process. 27 (3) (2018) 1501–1511.

[49] L. Gui, L.-P. Morency, Learning and transferring deep convnet representations with group-sparse factorization, in: Proc. IEEE Int. Conf. Comput. Vis., vol. 3, 2015.

[50] H. Schütze, C.D. Manning, P. Raghavan, Introduction to Information Retrieval, vol. 39, Cambridge Univ. Press, 2008.

[51] P. Zhou, C. Lu, Z. Lin, C. Zhang, Tensor factorization for low-rank tensor completion, IEEE Trans. Image Process. 27 (3) (2017) 1152–1163.

Yongyong Chen is now pursuing his Ph.D. degree in the department of computer and information science, University of Macau, Macau, China. His research interests include image processing, data mining and computer vision.

Xiaolin Xiao received the Ph.D. degree in the department of computer and information science from University of Macau in 2019. She is now a Postdoc Fellow in the Department of Computer and Information Science, University of Macau, Macau, China.

Yicong Zhou received his M.S. and Ph.D. degrees from Tufts University, Massachusetts, USA. He is currently an Associate Professor and Director of the Vision and Image Processing Laboratory in the Department of Computer and Information Science at University of Macau, Macau, China.