Unified Graph and Low-Rank Tensor Learning for Multi-View Clustering

Jianlong Wu,[1,2,3*] **Xingxu Xie,**[3*] **Liqiang Nie,**[1] **Zhouchen Lin,**[3,4†] **Hongbin Zha**[3]
[1]School of Computer Science and Technology, Shandong University
[2]Zhejiang Laboratory
[3]Key Laboratory of Machine Perception (MOE), School of EECS, Peking University
[4]Samsung Research China – Beijing (SRC-B)
jlwu1992@sdu.edu.cn, {xyxie, zlin}@pku.edu.cn, nieliqiang@gmail.com, zha@cis.pku.edu.cn

Abstract

Multi-view clustering aims to take advantage of multiple views information to improve the performance of clustering. Many existing methods compute the affinity matrix by low-rank representation (LRR) and pairwise investigate the relationship between views. However, LRR suffers from the high computational cost in self-representation optimization. Besides, compared with pairwise views, tensor form of all views' representation is more suitable for capturing the high-order correlations among all views. Towards these two issues, in this paper, we propose the unified graph and low-rank tensor learning (UGLTL) for multi-view clustering. Specifically, on the one hand, we learn the view-specific affinity matrix based on projected graph learning. On the other hand, we reorganize the affinity matrices into tensor form and learn its intrinsic tensor based on low-rank tensor approximation. Finally, we unify these two terms together and jointly learn the optimal projection matrices, affinity matrices and intrinsic low-rank tensor. We also propose an efficient algorithm to iteratively optimize the proposed model. To evaluate the performance of the proposed method, we conduct extensive experiments on multiple benchmarks across different scenarios and sizes. Compared with the state-of-the-art approaches, our method achieves much better performance.

Introduction

Along with the arrival of information age, it is easy to get a large number of multimedia data from the Internet and social media. However, the label information is often absent. While it costs much time and money to label the data, we can rely on clustering techniques (Ng, Jordan, and Weiss 2002; Liu et al. 2013; Wu et al. 2019) to investigate the correlations among data. There are many classic clustering methods, such as the k-means, spectral clustering, and subspace clustering (Liu et al. 2013; 2015). These traditional approaches achieve very good performance, but they mainly focus on single view input. In practice, the data is often collected in multiple poses and sources, such as image, text, or video. Even for a specific sample, we can also extract various kinds of features to represent it in different aspect. Under this circumstance, each pose, modality, or type of feature can be regarded as a specific view. Multi-view clustering is proposed to make full use of multi-view information to improve the clustering performance.

The construction of affinity matrix is a key step for clustering. In general, based on the affinity matrices of all views, multi-view clustering hopes to learn an intrinsic matrix, which can well capture both the consistent and complementary information among all views. With this comprehensive representation, we can further improve the performance of clustering. There are already many multi-view clustering methods. Many recent works explore the correlations among views based on the subspace clustering with self-representation. For example, (Cao et al. 2015) utilizes the Hilbert Schmidt Independence Criterion (HSIC) as a diversity term, and (Wang et al. 2017) introduces a position-aware exclusivity term to explore the complementarity. Instead of investigating correlations between pairwise views, (Xie et al. 2018) stacks the subspace representation matrices of all different views into a tensor and extract the integrated representation based on tensor low-rank decomposition. Although these subspace clustering based multi-view methods obtain good results, the computational complexity of self-representation optimization is $\mathcal{O}(n^3)$, which is very high and limits their extension to large dataset. Towards this issue, (Wu, Lin, and Zha 2019) proposes the robust tensor principle component analysis based on fixed affinity matrices, which are computed by the normalized Gaussian kernel of Markov chain based spectral clustering. But the separation of affinity matrix computing and comprehensive representation learning makes the solution sub-optimal for clustering.

In view of the above existing limitations, in this paper, we propose a novel unified graph and low-rank tensor learning (UGLTL) method for multi-view clustering. Specifically, to construct the affinity matrices, we learn view-specific projection matrix so that we can accurately compute similarity between samples according to their distance in the projected subspace. The computation cost is much lower than the self-representation based methods. Second, by stacking the multi-view affinity matrices into a tensor, we learn the low-rank tensor by the tensor Singular Value Decomposi-

*Equal contributions
†Corresponding author

tion (t-SVD) based tensor nuclear norm, which can thoroughly explore the high-order relationships among all views. Finally, we combine these two terms into a unified model to jointly learn the optimal affinity matrices and intrinsic low-rank tensor for clustering.

We summarize our main contributions as follows:

(a) We propose a novel unified method to jointly learn optimal affinity matrices in the projected subspace as well as its intrinsic low-rank tensor for multi-view clustering. With the t-SVD based tensor low-rank constraint, our method is effective to learn the comprehensive information among different views for clustering.

(b) We propose an efficient algorithm to alternately solve the proposed problem. Compared with those self-representation based methods, the computational complexity of our method is much lower.

(c) We conduct extensive experiments on multiple challenging datasets to evaluate the performance of our method. Compared with the state-of-the-art approaches, our method achieves significant improvement.

Related Work

According to the way to compute the affinity matrix, existing multi-view methods can be mainly divided into two categories, including the graph based models and the self-representation based subspace clustering methods.

The graph based methods are derived from the classic spectral clustering (Ng, Jordan, and Weiss 2002). The early multi-view methods mainly focus on the 2-view case. (Kumar and Daumé 2011) searches for the clusterings that agree across the views by a co-training approach. (Kumar, Rai, and Daumé 2011) explores the complementary information across views based on a co-regularization method. Then, RMSC (Xia et al. 2014) aims to recover a shared low-rank representation from multiple graphs. ETLMSC (Wu, Lin, and Zha 2019) learns the essential low-rank tensor based on the affinity tensor. Instead of using the Gaussian kernel to compute the similarity, MLAN (Nie, Cai, and Li 2017) and CLR (Nie et al. 2016) try to learn the weights of multiple graphs based on the Euclidian distance between samples.

Due to the popularity of SSC (Elhamifar and Vidal 2013) and LRR (Liu et al. 2013; Liu, Liu, and Li 2016; Liu and Zhang 2019), many recent multi-view learning methods (Cao et al. 2015; Zhang et al. 2015; Xie et al. 2018; Zhang et al. 2018) learn the affinity matrices based on self-representation. (Zhang et al. 2017) jointly learns the underlying latent representation and the low-rank decomposition. To learn the complementary information across multiple views, (Cao et al. 2015) and (Wang et al. 2017) utilize the Hilbert Schmidt Independence Criterion (HSIC) based diversity term and the position-aware exclusivity term, respectively. Tensor form (Lu et al. 2016; Zhou and Feng 2017; Kong, Xie, and Lin 2018) has been proved to be very effective in exploring the comprehensive information among multiple views. For example, LTMSC (Zhang et al. 2015) first extends the LRR into multi-view subspace clustering with generalized tensor nuclear norm, and then (Zhang et al. 2018) combines it with neural networks for further extension. (Xie et al. 2018) adopts the t-SVD based tensor nuclear norm for constraint. (Xie et al. 2019) extends the SSC into a differentiable form and proposes a new optimization strategy.

Notations and Preliminaries

To help understand the definition of tensor nuclear norm, we briefly introduce some notions and related definitions (Kilmer et al. 2013).

For a 3-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, vector along the i-th mode is called the mode-i fiber. $\mathbf{A}_{(i)}$ denotes the matricization of $\boldsymbol{\mathcal{A}}$ along the i-th mode, which can be constructed by arranging the mode-i fibers to be the columns of the resulting matrix. By transposing each frontal slice and then reversing the order of transposed frontal slices 2 through n_3, we get the transpose $\boldsymbol{\mathcal{A}}^T \in \mathbb{R}^{n_2 \times n_1 \times n_3}$. $\boldsymbol{\mathcal{A}}_f = \text{fft}(\boldsymbol{\mathcal{A}}, [\,], 3)$ denotes the fast Fourier transformation (FFT) of a tensor $\boldsymbol{\mathcal{A}}$ along the 3rd dimension, and its inverse operation is $\boldsymbol{\mathcal{A}} = \text{ifft}(\boldsymbol{\mathcal{A}}_f, [\,], 3)$. The block vectorizing and its inverse operation of $\boldsymbol{\mathcal{A}}$ are $\text{bvec}(\boldsymbol{\mathcal{A}}) = [\mathbf{A}^{(1)}; \mathbf{A}^{(2)}; \cdots; \mathbf{A}^{(n_3)}] \in \mathbb{R}^{n_1 n_3 \times n_2}$ and $\text{fold}(\text{bvec}(\boldsymbol{\mathcal{A}})) = \boldsymbol{\mathcal{A}}$, respectively. The block circulant matrix $\text{bcirc}(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ is defined by:

$$\text{bcirc}(\boldsymbol{\mathcal{A}}) := \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(n_3)} & \cdots & \mathbf{A}^{(2)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \cdots & \mathbf{A}^{(3)} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{A}^{(n_3)} & \mathbf{A}^{(n_3-1)} & \cdots & \mathbf{A}^{(1)} \end{bmatrix}.$$

Below are some related definitions.

Definition 1 (**t-product**). *Let $\boldsymbol{\mathcal{A}}$ be $n_1 \times n_2 \times n_3$, and $\boldsymbol{\mathcal{B}}$ be $n_2 \times n_4 \times n_3$. The t-product $\boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}}$ is the $n_1 \times n_4 \times n_3$ tensor*

$$\boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}} = \text{fold}(\text{bcirc}(\boldsymbol{\mathcal{A}})\text{bvec}(\boldsymbol{\mathcal{B}})). \tag{1}$$

Definition 2 (**f-diagonal tensor**). *A tensor is called f-diagonal if each of its frontal slices is diagonal matrix.*

Definition 3 (**Identity tensor**). *For the identity tensor $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{n \times n \times n_3}$, its first frontal slice is the identity matrix with size $n \times n$, and all other frontal slices are zero.*

Definition 4 (**Orthogonal tensor**). *A tensor $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{n \times n \times n_3}$ is orthogonal if it satisfies*

$$\boldsymbol{\mathcal{Q}}^{\mathrm{T}} * \boldsymbol{\mathcal{Q}} = \boldsymbol{\mathcal{Q}} * \boldsymbol{\mathcal{Q}}^{\mathrm{T}} = \boldsymbol{\mathcal{I}}. \tag{2}$$

Definition 5 (**t-SVD**). *For a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, it can be factorized by t-SVD as*

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^T, \tag{3}$$

where $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal, and $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is f-diagonal.

Definition 6 (**t-SVD based tensor nuclear norm**). *The t-SVD based tensor nuclear norm $\|\boldsymbol{\mathcal{A}}\|_{\circledast}$ of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined by the sum of singular values of all the frontal slices of $\boldsymbol{\mathcal{A}}_f$:*

$$\|\boldsymbol{\mathcal{A}}\|_{\circledast} = \sum_{k=1}^{n_3} \|\boldsymbol{\mathcal{A}}_f^{(k)}\|_* = \sum_{i=1}^{\min(n_1,n_2)} \sum_{k=1}^{n_3} |\boldsymbol{\mathcal{S}}_f^{(k)}(i,i)|, \tag{4}$$

where $\boldsymbol{\mathcal{S}}_f^{(k)}$ is computed by the SVD $\boldsymbol{\mathcal{A}}_f^{(k)} = \boldsymbol{\mathcal{U}}_f^{(k)} \boldsymbol{\mathcal{S}}_f^{(k)} \boldsymbol{\mathcal{V}}_f^{(k)T}$ of frontal slices of $\boldsymbol{\mathcal{A}}_f$.

Unified Graph and Low-rank Tensor Learning

Model Formulation

Let $\mathbf{X}^v = [\mathbf{x}_1^v, \cdots, \mathbf{x}_N^v] \in \mathbb{R}^{d^v \times N}$ denote the data matrix of the v-th view ($v = 1, \cdots, V$), where d^v is the dimension of feature vectors in the v-th view, N is the number of samples, and V is the number of views. For multi-view clustering, we first need to construct the view-specific affinity matrix. Even though we can simply compute the similarity by Gaussian kernel just like what the standard spectral clustering does, we hope this process could be jointly optimized with the later multi-view learning to get the optimal solution. So we assign similarity for data samples according to their distance based on graph learning (Nie, Wang, and Huang 2014).

The basic model of graph learning to learn the similarity $s_{i,j}^v$ can be formulated as follows:

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{v=1}^{V} \sum_{i,j=1}^{N} \|\mathbf{x}_i^v - \mathbf{x}_j^v\|_2^2 s_{ij}^v + \frac{\gamma}{2}(s_{ij}^v)^2, \\ &\text{s.t. } \forall j \text{ and } v,\ (\mathbf{s}_j^v)^T \mathbf{1} = 1,\ \mathbf{s}_j^v \geq \mathbf{0}, \end{aligned} \tag{5}$$

where γ is a balance parameter, $\mathbf{s}_j^v \in \mathbb{R}^{N \times 1}$ is a column vector with the i-th element as s_{ij}^v, $\mathbf{1} \in \mathbb{R}^{N \times 1}$ is a column vector with all elements as 1, and $\mathbf{0} \in \mathbb{R}^{N \times 1}$ is a column vector with all elements as 0. The quadric term is used for regularization to avoid trivial solution. In general, if the distance between two samples is small, then a large similarity s_{ij}^v will be assigned.

In practice, the feature dimension d^v might be very high. The Euclidian distance in the original feature space might not be suitable. It is challenging to deal with high-dimensional data. For this issue, we can project the feature into a lower dimensional subspace and then learn the affinity matrices. Then the graph learning problem is transformed to:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{S}} \sum_{v=1}^{V} \sum_{i,j=1}^{N} \|\mathbf{W}^{vT}\mathbf{x}_i^v - \mathbf{W}^{vT}\mathbf{x}_j^v\|_2^2 s_{ij}^v + \frac{\gamma}{2}(s_{ij}^v)^2, \\ &\text{s.t. } \forall j \text{ and } v,\ (\mathbf{s}_j^v)^T \mathbf{1} = 1,\ \mathbf{s}_j^v \geq \mathbf{0},\ \mathbf{W}^{vT}\mathbf{X}^v\mathbf{X}^{vT}\mathbf{W}^v = \mathbf{I}, \end{aligned} \tag{6}$$

where $\mathbf{I}$ is the identical matrix, $\mathbf{W}^v \in \mathbb{R}^{d^v \times M}$ is the view-specific projection matrix, and M is the dimension of the projected subspace. Similar to canonical component analysis, orthogonal subspace constraint is adopt to make the feature embedding statistically uncorrelated in the projected subspace and constrain learning of view-specific projection matrix $\mathbf{W}^v$.

Based on the view-specific affinity matrix, we hope to learn an intrinsic representation which can capture both consistent and complementary information among multiple views. Many existing methods try to learn a shared representation (Nie, Cai, and Li 2017) or investigate pair-wise correlation (Wang et al. 2017), which results in the loss of comprehensiveness and optimality in the representation. As a comparison, tensor form (Kilmer et al. 2013) is much more suitable to explore the high-order correlations among multiple views, so we stack the affinity matrices of all views into a tensor and rotate it into $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{N \times V \times N}$, which can better investigate the correlations and largely reduce the computational complexity.

As multi-view features are extracted from the same objects, different $\mathbf{S}^v$ should contain some similar information. Another fact is that the number of clusters is always much smaller than the sample number. In this case, the affinity tensor $\boldsymbol{\mathcal{S}}$ should be low-rank. For the tensor rank, since the CANDECOMP/PARAFAC (CP) (Carroll and Chang 1970; Harshman 1970) rank is generally NP-hard to compute and the Sum of Nuclear Norms (SNN) (Huang et al. 2014) for Tucker (Tucker 1966) decomposition is not a tight convex relaxation of the Tucker rank, so we adopt t-SVD (Kilmer et al. 2013) based tensor nuclear norm, which has been proven to be the tightest convex relaxation (Zhang et al. 2014) to ℓ_1-norm of the tensor multi-rank, to constrain the intrinsic low-rank tensor. Considering the influence of noise on $\boldsymbol{\mathcal{S}}$, we learn the low-rank tensor $\boldsymbol{\mathcal{Z}}$ to approximate original affinity tensor by:

$$\min_{\boldsymbol{\mathcal{Z}}} \|\boldsymbol{\mathcal{Z}}\|_{\circledast} + \frac{\alpha}{2}\|\boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{Z}}\|_F^2, \tag{7}$$

where α is a constant to control the influence of noises, and the tensor $\|\cdot\|_F$-norm is defined by $\|\boldsymbol{\mathcal{A}}\|_F = \sqrt{\sum_{ijk} |\boldsymbol{\mathcal{A}}_{ijk}|^2}$, which is used to penalize the noise term.

The separation of learning $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{Z}}$ will make the solution sub-optimal. So we combine the affinity matrices learning in Eq. (6) and low-rank tensor learning in Eq. (7) to jointly optimize them. We also add a constraint on each similarity matrix $\mathbf{S}^\mathbf{v}$ to make the learned graph symmetric, which means that the similarity between two samples should be same ($S_{ij}^v = S_{ji}^v$). Then the final objective function of UGLTL can be formulated as:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{S}},\boldsymbol{\mathcal{Z}},\mathbf{W}} &\sum_{v=1}^{V} \sum_{i,j=1}^{N} \left(\|\mathbf{W}^{vT}\mathbf{x}_i^v - \mathbf{W}^{vT}\mathbf{x}_j^v\|_2^2 s_{ij}^v + \frac{\gamma}{2}(s_{ij}^v)^2 \right) \\ &+ \frac{\alpha}{2}\|\boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{Z}}\|_F^2 + \beta\|\boldsymbol{\mathcal{Z}}\|_{\circledast}, \\ \text{s.t. } &\forall j \text{ and } v,\ (\mathbf{s}_j^v)^T \mathbf{1} = 1,\ \mathbf{s}_j^v \geq \mathbf{0},\ \mathbf{S}^v = \mathbf{S}^{vT}, \\ &\mathbf{W}^{vT}\mathbf{X}^v\mathbf{X}^{vT}\mathbf{W}^v = \mathbf{I},\ \boldsymbol{\mathcal{S}} = \Phi(\mathbf{S}^1, \cdots, \mathbf{S}^V), \end{aligned} \tag{8}$$

where α, β and γ are balance parameters, function $\Phi(\cdot)$ merges affinity matrices of various views into a 3-order tensor and then rotates along the z-axis.

Optimization

In this subsection, we propose an efficient algorithm to solve the problem in Eq. (8) in an alternate way. It's obvious that the problem is not jointly convex to $\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{Z}}$, and $\mathbf{W}$, but it is convex to each variable while other variables are fixed. We alternately optimize each variable with other variables fixed as follows.

$\boldsymbol{\mathcal{Z}}$-subproblem: When the tensor $\boldsymbol{\mathcal{S}}$ and projection matrices $\mathbf{W}$ are fixed, we update the tensor $\boldsymbol{\mathcal{Z}}$ by solving:

$$\boldsymbol{\mathcal{Z}}^* = \operatorname*{argmin}_{\boldsymbol{\mathcal{Z}}} \frac{\alpha}{2}\|\boldsymbol{\mathcal{S}} - \boldsymbol{\mathcal{Z}}\|_F^2 + \beta\|\boldsymbol{\mathcal{Z}}\|_{\circledast}. \tag{9}$$

The optimal solution can be computed by the tensor tubal-shrinkage operator (Hu et al. 2016):

$$\boldsymbol{\mathcal{Z}}^* = \mathcal{C}_{n_3\tau}(\boldsymbol{\mathcal{S}}) = \boldsymbol{\mathcal{U}} * \mathcal{C}_{n_3\tau}(\boldsymbol{\mathcal{O}}) * \boldsymbol{\mathcal{V}}^{\mathrm{T}}, \tag{10}$$

where $\tau = \beta/\alpha$, $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{O}} * \boldsymbol{\mathcal{V}}^{\mathrm{T}}$ denotes the tensor SVD decomposition, and $\mathcal{C}_{n_3\tau}(\boldsymbol{\mathcal{O}}) = \boldsymbol{\mathcal{O}} * \boldsymbol{\mathcal{J}}$, herein, $\boldsymbol{\mathcal{J}}$ is an $n_1 \times n_2 \times n_3$ f-diagonal tensor whose diagonal element in the Fourier domain is $\boldsymbol{\mathcal{J}}_f(i,i,j) = (1 - \frac{n_3\tau}{\boldsymbol{\mathcal{O}}_f^{(j)}(i,i)})_+$.

$\boldsymbol{\mathcal{S}}$-subproblem: To optimize $\boldsymbol{\mathcal{S}}$, we fix the tensor $\boldsymbol{\mathcal{Z}}$ and projection matrices $\mathbf{W}$. We first solve the problem without symmetric constraint. The Lagrangian function to optimize $\boldsymbol{\mathcal{S}}$ can be reformulated as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\mathcal{S}}) = &\sum_{v=1}^{V}\sum_{i,j=1}^{N}\left(\|\mathbf{W}^{vT}\mathbf{x}_i^v - \mathbf{W}^{vT}\mathbf{x}_j^v\|_2^2 s_{ij}^v + \frac{\gamma}{2}(s_{ij}^v)^2\right)\\ &+\frac{\alpha}{2}\|\boldsymbol{\mathcal{S}}-\boldsymbol{\mathcal{Z}}\|_F^2+\sum_{v=1}^{V}\sum_{j=1}^{N}\left(\eta_j^v\left((\mathbf{s}_j^v)^T\mathbf{1}-1\right)-(\lambda_j^v)^T\mathbf{s}_j^v\right),\end{aligned} \tag{11}$$

where η_j^v and λ_j^v are Lagrangian multipliers. η_j^v is a non-negative constant, and the column vector $\lambda_j^v \geq \mathbf{0}$. Then each vector of the tensor $\boldsymbol{\mathcal{S}}$ can be updated by solving the following subproblem to get the close-form solution:

$$\begin{aligned}\mathbf{s}_j^{v*} &= \operatorname*{argmin}_{\mathbf{s}_j^v}\sum_{i=1}^{N}\left(\|\mathbf{W}^{vT}\mathbf{x}_i^v - \mathbf{W}^{vT}\mathbf{x}_j^v\|_2^2 s_{ij}^v + \frac{\gamma}{2}(s_{ij}^v)^2\right)\\ &+\frac{\alpha}{2}\|\mathbf{s}_j^v - \mathbf{z}_j^v\|_2^2 - \eta_j^v\left((\mathbf{s}_j^v)^T\mathbf{1}-1\right) - (\lambda_j^v)^T\mathbf{s}_j^v, \qquad (12)\\ &= \operatorname*{argmin}_{\mathbf{s}_j^v}\ \frac{\gamma+\alpha}{2}\|\mathbf{s}_j^v + \mathbf{g}_j^v\|_2^2 - \eta_j^v\left((\mathbf{s}_j^v)^T\mathbf{1}-1\right)-(\lambda_j^v)^T\mathbf{s}_j^v,\end{aligned}$$

where $g_{ij}^v = \frac{\|\mathbf{W}^{vT}\mathbf{x}_i^v - \mathbf{W}^{vT}\mathbf{x}_j^v\|_2^2 - \alpha\mathbf{z}_{ij}^v}{\gamma+\alpha}$. Based on the KKT conditions, we have:

$$(\gamma+\alpha)(s_{ij}^{v*} + g_{ij}^v) - \eta_j^{v*} - \lambda_{ij}^{v*} = 0,\ \forall i,j; \tag{13}$$

$$\mathbf{s}_{ij}^{v*}\lambda_{ij}^{v*} = 0,\ \ s_{ij}^{v*} \geq 0,\ \ \lambda_{ij}^{v*} \geq 0,\ \forall i,j. \tag{14}$$

According to Eq. (13) and the constraint $(\mathbf{s}_j^v)^T\mathbf{1} = 1$, we get:

$$\eta_j^{v*} = \frac{(\gamma+\alpha)(1 + (\mathbf{g}_j^v)^T\mathbf{1}) - (\lambda_j^v)^T\mathbf{1}}{N}. \tag{15}$$

Then according to Eq. (13) and the complementary slackness condition in Eq. (14), we have:

$$s_{ij}^{v*} = \max\{0, \frac{1 + (\mathbf{g}_j^v)^T\mathbf{1}}{N} - \frac{(\lambda_j^{v*})^T\mathbf{1}}{N(\gamma+\alpha)} - g_{ij}^v\}. \tag{16}$$

Based on s_{ij}^{v*}, according to the Proposition 7 in (Lu et al. 2018), the symmetric constraint can be satisfied by:

$$\tilde{s}_{ij}^{v*} = \frac{1}{2}(s_{ij}^{v*} + s_{ji}^{v*}). \tag{17}$$

W-subproblem: When the similarity tensor $\boldsymbol{\mathcal{S}}$ and the essential tensor $\boldsymbol{\mathcal{Z}}$ are fixed, the problem for solving $\mathbf{W}$ is:

$$\begin{aligned}\min_{\mathbf{W}}&\sum_{v=1}^{V}\sum_{i,j=1}^{N}\left(\|\mathbf{W}^{vT}\mathbf{x}_i^v - \mathbf{W}^{vT}\mathbf{x}_j^v\|_2^2 s_{ij}^v\right),\\ &\text{s.t. } \forall v,\ \mathbf{W}^{vT}\mathbf{X}^v\mathbf{X}^{vT}\mathbf{W}^v = \mathbf{I},\end{aligned} \tag{18}$$

Algorithm 1 Alternating Minimization Method for UGLTL

Input: Multi-view data matrix $\mathbf{X}^v$, parameters α, β, γ.
Initialize s_{ij}^v, $\mathbf{W}^v$ by the Gaussian kernel and identical matrix, respectively.
1: **while** not converged **do**
2: Compute the close-form solution for $\boldsymbol{\mathcal{Z}}$ by Eq. (10);
3: Update the final similarity s_{ij}^v by Eqs. (16) and (17);
4: Update the feature embedding $\mathbf{Y}^v$ by solving Eq. (20);
5: **end while**
Based on $\boldsymbol{\mathcal{Z}}$, compute the affinity matrix by:
$\mathbf{A} = \frac{1}{V}\sum_{v=1}^{V}\left(|\mathbf{Z}^{(v)}| + |\mathbf{Z}^{(v)^T}|\right)$;
Apply the spectral clustering to $\mathbf{A}$ to get final result;
Output: Clustering result.

which is equivalent to solving the following V subproblems:

$$\begin{aligned}&\min_{\mathbf{W}^v}\mathrm{Tr}(\mathbf{W}^{vT}\mathbf{X}^v\mathbf{L}^v\mathbf{X}^{vT}\mathbf{W}^v),\\ &\text{s.t. } \mathbf{W}^{vT}\mathbf{X}^v\mathbf{X}^{vT}\mathbf{W}^v = \mathbf{I}, \forall\, v = 1, 2, \cdots, V,\end{aligned} \tag{19}$$

where $\mathbf{L}^v = \mathbf{D}^v - \mathbf{S}^v$ is the Laplacian matrix for the v-th view, and $\mathbf{D}^v$ is a diagonal matrix with $D_{ii}^v = \sum_j s_{ij}^v$. Denote $\mathbf{Y}^v = \mathbf{W}^{vT}\mathbf{X}^v$, then the above problem in Eq. (19) can be reformulated as:

$$\min_{\mathbf{Y^v}}\mathrm{Tr}\left(\mathbf{Y}^v\mathbf{L}^v\mathbf{Y}^{vT}\right), \text{s.t.}\mathbf{Y}^v\mathbf{Y}^{vT} = \mathbf{I}, \forall\, v = 1, \cdots, V, \tag{20}$$

which is an eigenvalue decomposition problem. The optimal solution of $\mathbf{Y}^v$ can be formed by the k eigenvectors of $\mathbf{L}^v$ corresponding to the k smallest eigenvalues. After we get the optimal $\mathbf{Y}^v$, we do not need to compute the projection matrix $\mathbf{W}^v$ any longer, since we can directly use the projected features $\mathbf{y}_j^v$ to replace $\mathbf{W}^{vT}\mathbf{x}_j^v$ in the later optimization.

After we get the optimal low-rank tensor $\boldsymbol{\mathcal{Z}}$, we adopt the standard way for multi-view clustering to compute the affinity matrix as $\mathbf{A} = \frac{1}{V}\sum_{v=1}^{V}\left(|\mathbf{Z}^{(v)}| + |\mathbf{Z}^{(v)^T}|\right)$, based on which we apply the spectral clustering to compute the final clustering result. The overall process is summarized in Algorithm 1.

Convergence and Complexity

We solve the problem by alternating minimization. Although the problem in Eq. (8) is non-convex and non-smooth, the subproblem is strongly convex with other variables fixed. We can exactly obtain the closed-form solution for each subproblem. For the convergence, we provide a proof sketch here. We first denote $\mathbf{Y} = \mathbf{W}^T\mathbf{X}$ and lift the non-linear constraints in Eq. (8) by adding two indicator functions on the objective $\mathcal{I}(\mathbf{Y}) = \mathcal{I}\{\mathbf{Y}^v\mathbf{Y}^{vT} = \mathbf{I}, \forall v\}$ and $\mathcal{I}(\boldsymbol{\mathcal{S}}) = \mathcal{I}\{\mathbf{s}_j^v \geq \mathbf{0}, \forall v, j\}$. Then Eq. (8) has the form $\min_{\boldsymbol{\mathcal{Z}},\boldsymbol{\mathcal{S}},\mathbf{Y}} f(\boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{S}}, \mathbf{Y}) + \beta\|\boldsymbol{\mathcal{Z}}\|_{\circledast} + \mathcal{I}(\mathbf{Y}) + \mathcal{I}(\boldsymbol{\mathcal{S}})$, s.t. $\mathcal{A}(\boldsymbol{\mathcal{S}}) = 0$, where f is Lipschitz differentiable, and $\mathcal{A}(\cdot)$ is a linear operator. Consider the Lagrangian function $L(\boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{S}}, \mathbf{Y}, \mu)$ of the above problem, where μ is the multiplier. Denote $(\boldsymbol{\mathcal{Z}}^+, \boldsymbol{\mathcal{S}}^+, \mathbf{Y}^+)$ as the variables at

Table 1: Experimental results on the COIL-20 and the UCI-Digit datasets.

Datasets	COIL-20						UCI-Digits					
Methods	NMI	ACC	AR	F-score	Precision	Recall	NMI	ACC	AR	F-score	Precision	Recall
SPC$_{best}$	0.806	0.672	0.619	0.640	0.596	0.692	0.642	0.731	0.545	0.591	0.582	0.601
LRR$_{best}$	0.829	0.761	0.720	0.734	0.717	0.751	0.768	0.871	0.736	0.763	0.759	0.767
PCAN$_{best}$	0.872	0.772	0.620	0.643	0.530	0.816	0.897	0.949	0.887	0.899	0.895	0.902
Co-reg	0.774	0.659	0.592	0.613	0.590	0.640	0.804	0.780	0.755	0.780	0.764	0.798
RMSC	0.800	0.685	0.637	0.656	0.620	0.698	0.822	0.915	0.789	0.811	0.797	0.826
DiMSC	0.846	0.778	0.732	0.745	0.739	0.751	0.772	0.703	0.652	0.695	0.673	0.718
LTMSC	0.860	0.804	0.748	0.760	0.741	0.479	0.775	0.803	0.725	0.753	0.739	0.767
ECMSC	0.942	0.782	0.781	0.794	0.695	0.925	0.780	0.718	0.672	0.707	0.660	0.760
t-SVD-MSC	0.884	0.830	0.786	0.800	0.785	0.808	0.932	0.955	0.924	0.932	0.930	0.934
ETLMSC	0.947	0.877	0.862	0.869	0.830	0.914	0.977	0.958	0.953	0.958	0.940	0.980
UGLTL	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

the $(k+1)$-th iteration and omit the superscript for the k-th iteration. Based on the KKT condition, the strong-convexity of problems (9) and (12), and the optimality of $\mathbf{Y}^+$, we can have $L(\boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{S}}, \mathbf{Y}, \mu) - L(\boldsymbol{\mathcal{Z}}^+, \boldsymbol{\mathcal{S}}^+, \mathbf{Y}^+, \mu^+) \geq C(\|\boldsymbol{\mathcal{Z}}^+ - \boldsymbol{\mathcal{Z}}\|_F^2 + \|\boldsymbol{\mathcal{S}}^+ - \boldsymbol{\mathcal{S}}\|_F^2)$. The coercivity helps to lower bound $L(\boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{S}}, \mathbf{Y}, \mu)$, and then we can have $(\|\boldsymbol{\mathcal{Z}}^+ - \boldsymbol{\mathcal{Z}}\|_F^2 + \|\boldsymbol{\mathcal{S}}^+ - \boldsymbol{\mathcal{S}}\|_F^2) \to 0$ and get the boundness of $\{(\boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{S}}, \mathbf{Y}, \mu)\}$ which implies the existence of accumulation point $(\boldsymbol{\mathcal{Z}}^*, \boldsymbol{\mathcal{S}}^*, \mathbf{Y}^*, \mu^*)$. On the other hand, $(\|\boldsymbol{\mathcal{Z}}^+ - \boldsymbol{\mathcal{Z}}\|_F^2 + \|\boldsymbol{\mathcal{S}}^+ - \boldsymbol{\mathcal{S}}\|_F^2)$ also bounds the generalized subgradient of $L(\boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{S}}, \mathbf{Y}, \mu)$, hence $0 \in \partial L(\boldsymbol{\mathcal{Z}}^*, \boldsymbol{\mathcal{S}}^*, \mathbf{Y}^*, \mu^*)$, which indicates the convergence. Note that $\|\boldsymbol{\mathcal{Z}}\|_{\circledast}, \mathcal{I}(\mathbf{Y})$ and $\mathcal{I}(\boldsymbol{\mathcal{S}})$ locally have Lipschitz continuous (sub)gradient, which essentially ensures the convergence rather than the convexity and separability, please see (Wang, Yin, and Zeng 2019) for more details.

For the complexity, it takes $\mathcal{O}(VN^2\log(N))$ to perform FFT and inverse FFT on a $N \times V \times N$ tensor along the third dimension. To update $\boldsymbol{\mathcal{Z}}$, we also need to compute the SVD of each frontal slice with size $N \times V$ in the Fourier domain, which takes $\mathcal{O}(V^2N^2)$ for the whole tensor. So it takes $\mathcal{O}(V^2N^2 + VN^2\log(N))$ in total to update $\boldsymbol{\mathcal{Z}}$. To update each vector $\mathbf{s}_i^v$, it needs $\mathcal{O}(N)$ by Eq. (16) as we only need to compute $(\mathbf{g}_i^v)^T\mathbf{1}$ and $(\lambda_i^v)^T\mathbf{1}$ once. So we need $\mathcal{O}(VN^2)$ to update the tensor $\boldsymbol{\mathcal{S}}$. For the optimization of subspace embedding, we only need to compute the M smallest eigenvalues and their corresponding eigenvectors of each $\mathbf{L}^v$, which costs $\mathcal{O}(VMN^2)$ in total. In general, the number of views V is smaller than M. Denote K as the number of iterations, the total complexity to optimize UGLTL in Algorithm 1 is $\mathcal{O}(KVN^2(M+\log(N)))$, which is relatively efficient.

Experiments

Experimental Settings

Datasets We adopt six challenging image datasets, which cover various sizes and applications, including the COIL-20[1], UCI-Digits (Asuncion and Newman 2007), Scene-15 (Li and Pietro 2005), Notting-Hill (Zhang et al. 2009), MITIndoor-67 (Quattoni and Torralba 2009), and Caltech-101 (Li, Rob, and Pietro 2007) datasets. In Table 2, we summarize the statistic information of these datasets. For all these datasets, same to (Xia et al. 2014) and (Xie et al. 2018), we extract three different kinds of features as three views. For details and multi-view features of these datasets, please refer to (Xia et al. 2014) and (Xie et al. 2018). We need to mention that for the Caltech-101 dataset, we use all $8,677$ instances of 101 categories to test the performance, which is very challenging.

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

Table 2: Statistics of different datasets

Dataset	Images	Objective	Clusters
COIL-20	1440	Generic object	20
UCI-Digits	2000	Digit	10
Scene-15	4485	Scene	15
Notting-Hill	4660	Video Face	5
MITIndoor-67	5360	Scene	67
Caltech-101	8677	Generic object	101

Compared Methods We compare our approach with the following state-of-the-art methods: the standard spectral clustering (Ng, Jordan, and Weiss 2002) on the best view (SPC$_{best}$), the low-rank representation (Liu et al. 2013) on the best view (LRR$_{best}$), projected graph learning with adaptive neighbors (Nie, Wang, and Huang 2014) on the best view (PCAN$_{best}$), RMSC (Xia et al. 2014), MLAN (Nie, Cai, and Li 2017), DiMSC (Cao et al. 2015), LTMSC (Zhang et al. 2015), ECMSC (Wang et al. 2017), t-SVD-MSC (Xie et al. 2018), and ETLMSC (Wu, Lin, and Zha 2019). For above methods, only the first three are single view based methods, and others focus on multi-view learning.

Evaluation Metrics We adopt all six commonly used metrics including normalized mutual information (NMI), accuracy (ACC), adjusted rand index (AR), F-score, precision, and recall to comprehensively evaluate the performance of clustering. For details of these metrics, please refer to (Xie et al. 2018). These six metrics favour different properties of a clustering task. For all metrics, the higher value indicates

Table 3: Experimental results on the Scene-15 and the Notting-Hill datasets.

Datasets	Scene-15						Notting-Hill					
Methods	NMI	ACC	AR	F-score	Precision	Recall	NMI	ACC	AR	F-score	Precision	Recall
SPC_{best}	0.421	0.437	0.270	0.321	0.314	0.329	0.723	0.816	0.712	0.775	0.780	0.776
LRR_{best}	0.426	0.445	0.272	0.324	0.316	0.333	0.579	0.794	0.558	0.653	0.672	0.636
$PCAN_{best}$	0.545	0.527	0.264	0.336	0.238	0.575	0.100	0.355	0.010	0.364	0.228	0.902
Co-reg	0.470	0.503	0.334	0.380	0.382	0.378	0.703	0.805	0.686	0.754	0.766	0.743
RMSC	0.564	0.507	0.394	0.437	0.425	0.450	0.585	0.807	0.496	0.603	0.621	0.586
DiMSC	0.269	0.300	0.117	0.181	0.173	0.190	0.799	0.837	0.787	0.834	0.822	0.827
LTMSC	0.571	0.574	0.424	0.465	0.452	0.479	0.779	0.868	0.777	0.825	0.830	0.814
ECMSC	0.463	0.457	0.303	0.357	0.318	0.408	0.817	0.767	0.679	0.764	0.637	0.954
t-SVD-MSC	0.858	0.812	0.771	0.788	0.743	0.839	0.900	**0.957**	0.900	0.922	0.937	0.907
ETLMSC	0.902	0.878	0.851	0.862	0.848	0.877	0.911	0.951	0.898	**0.924**	**0.940**	0.908
UGLTL	**0.960**	**0.976**	**0.952**	**0.955**	**0.961**	**0.950**	**0.921**	0.950	**0.903**	**0.924**	0.939	**0.910**

Table 4: Experimental results on the MITIndoor-67 and the Caltech-101 datasets.

Datasets	MITIndoor-67						Caltech-101					
Methods	NMI	ACC	AR	F-score	Precision	Recall	NMI	ACC	AR	F-score	Precision	Recall
SPC_{best}	0.559	0.443	0.304	0.315	0.294	0.340	0.723	0.484	0.319	0.340	0.597	0.235
LRR_{best}	0.226	0.120	0.031	0.045	0.044	0.047	0.728	0.510	0.304	0.339	0.627	0.231
$PCAN_{best}$	0.184	0.081	0.002	0.030	0.016	0.420	0.806	0.585	0.296	0.322	0.471	0.245
Co-reg	0.270	0.149	0.054	0.067	0.066	0.070	0.824	0.582	0.401	0.412	0.661	0.301
RMSC	0.342	0.232	0.110	0.123	0.121	0.125	0.573	0.346	0.246	0.258	0.457	0.182
DiMSC	0.383	0.246	0.128	0.141	0.138	0.144	0.589	0.351	0.226	0.253	0.362	0.191
LTMSC	0.226	0.120	0.031	0.045	0.044	0.047	0.788	0.559	0.393	0.403	0.670	0.288
ECMSC	0.590	0.469	0.323	0.333	0.314	0.355	0.662	0.419	0.312	0.326	0.465	0.251
t-SVD-MSC	0.750	0.684	0.555	0.562	0.543	0.582	0.858	0.607	0.430	0.440	0.742	0.323
ETLMSC	0.899	0.775	0.729	0.733	0.709	0.758	0.899	0.639	0.456	0.465	0.825	0.324
UGLTL	**0.979**	**0.948**	**0.940**	**0.940**	**0.930**	**0.951**	**0.902**	**0.669**	**0.504**	**0.513**	**0.960**	**0.365**

the better performance.

All experiments are implemented in Matlab on a desktop with 3.4GHz CPU and 32G RAM.

Experimental Results and Analysis

Performance Comparison In Tables 1-4, we present the experimental results on these six datasets. The bold values denote the best performance. All results are measured by the average of 20 runs. The standard deviations on all datasets and under all metrics are smaller than 0.1, so we do not show it due to page limit. To better compare the performance of different methods, we divide all methods into four subclasses in the table, including single view methods, spectral clustering methods, subspace learning methods, and tensor based methods. Most results are directly copied from (Xie et al. 2018), while other results are achieved based on their shared code with parameter adjustment.

It is obvious that our proposed UGLTL achieves the best performance on nearly all datasets under all metrics. There is a clear advance over the ETLMSC and t-SVD-MSC, which achieve the second and third best results, respectively. Compared with ETLMSC, take ACC for example, our method gains significant improvement around 12.3%, 4.2%, 9.8%, 17.3%, 3.0% on the COIL-20, UCI-Digit, Scene-15, MITIndoor-67, and Caltech-101 datasets, respectively. Especially on COIL-20 and UCI-Digits, our method can accurately cluster all instances. Even on the challenging Caltech-101 dataset with 101 clusters, UGLTL works very well. On the Notting-Hill dataset of video face, these three tensor based methods achieve comparable results. According to (Arpit, Nwogu, and Govindaraju 2014), facial images have a subspace structure, so subspace learning based methods is more suitable for this task. While t-SVD-MSC is based on subspace learning, the performance of our method is still comparable to that achieved by t-SVD-MSC.

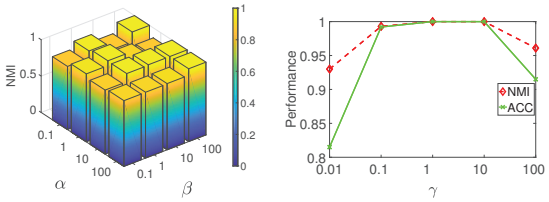
For different subclasses of methods, we can see that the tensor based methods achieve much better results than all other methods on all datasets, which can verify the effectiveness of tensor low-rank minimization in exploring high-order correlations among multiple views. In general cases, multi-view learning methods achieves better performance than the single view methods. However, with the incorporation of deep features on the MITIndoor-67 and the Caltech-101 datasets, some multi-view methods, such as RMSC, DiMSC, and ECMSC, show worse results than the best single view methods, which can be attributed to their less representation ability and being easier to be affected by the degenerated view. The improvement of our method over ETLMSC also shows the importance of learning similarity and multi-view embedding jointly.

Table 5: NMI comparison of three methods on these datasets.

Methods\Datasets	COIL-20	UCI-Digits	Scene-15	Notting-Hill	MITIndoor-67	Caltech-101
MLAN	0.937	0.934	0.471	0.679	0.442	0.813
UGLTL(no projection)	**1.000**	0.989	0.942	0.905	0.941	0.878
UGLTL	**1.000**	**1.000**	**0.960**	**0.921**	**0.979**	**0.902**

Table 6: Computational complexity and running time on the COIL-20 dataset of different methods. K, V, N are the number of iterations, views, and samples, respectively. M is the dimension of projected features in the subspace.

Methods	RMSC	DiMSC	LTMSC	ECMSC	t-SVD-MSC	ETLMSC	UGLTL (ours)
Complexity	$\mathcal{O}(KN^3)$	$\mathcal{O}(KVN^3)$	$\mathcal{O}(KVN^3)$	$\mathcal{O}(KVN^3)$	$\mathcal{O}(VN^3+KVN^2\log(N))$	$\mathcal{O}(KVN^2\log(N))$	$\mathcal{O}(KVN^2(M+\log(N)))$
Time(seconds)	74.8s	1075.1s	396.0s	954.2s	103.4s	19.6s	**16.5s**



(a) α and β (b) γ

Figure 1: Parameters tuning with respect to α, β, and λ on the COIL-20 dataset. (a) Fix $\gamma = 1$, tune the α and β; (b) Fix $\alpha = 10$ and $\beta = 50$, tune γ. All the horizontal axises are in log scale. (Best view in color)

Influence of Low-rank Tensor and Projection To evaluate the effectiveness of low-rank tensor decomposition and subspace projection, we compare UGLTL with MLAN and UGLTL without projection. Please note that MLAN simply learns a low-rank matrix shared by all different views. For simplicity, we only present the NMI results on all six datasets in Table 5. We can easily observe that both UGLTL and its no projection version achieve 5% higher NMI performance than MLAN on all these datasets, especially on the difficult Scene-15 and MITIndoor-67 datasets, which shows the superiority of low-rank tensor decomposition. On the other hand, there is an average 2% improvement by incorporating the subspace projection, because this projection operation can benefit the learning of optimal affinity matrices and computation efficiency.

Complexity Comparison In Table 6, we present the computational complexity and running time of the state-of-the-art methods on the COIL-20 dataset. Our method has the shortest processing time among all related approaches on this dataset. The dimension M of projected features in the subspace is very small and we set it to $M = 8$ in our experiments. So the complexity of UGLTL is in the same order as ETLMSC and much lower than other methods. Since our method converges very fast and K is a very small value in our algorithm, so it has lower running time than ETLMSC. Compared with another tensor based method, our algorithm can finish within 20 seconds, while t-SVD-MSC needs more than 100 seconds.

Parameters Setting and Sensitivity Analysis The parameters α, β, and γ are fine-tuned by searching the grid of $\{0.01, 0.1, 1, 10, 100\}$. In Figure 1, we present the experimental results on the COIL-20 dataset with respect to different parameters. As there are 3 parameters in our model, we first fix $\gamma = 1$ to tune α and β. According to Figure 1a, we can see that when both α and $\tau = \frac{\beta}{\alpha}$ range in $[1, 10]$, the result is very stable. Then we show the results of different γ by fixing $\alpha = 10$ and $\beta = 50$ in Figure 1b. It is obvious that our algorithm is insensitive to parameter γ, especially when it ranges in $[0.1, 10]$, both its NMI and ACC are all nearly 1. In summary, when these parameters range in a relatively large interval, our algorithm is insensitive. Besides, parameters of our algorithm are insensitive to different datasets. On five datasets, we use the same set of parameters to achieve the performance presented in Tables 1-4.

Convergence Analysis The parameter $\tau = \frac{\beta}{\alpha}$ plays an important role in controlling the contribution of low-rank tensor minimization, which has a serious impact on the iteration number K. With a proper τ, our objective value converges very fast. In experiments, we set $\tau = 5$ on all datasets, and it needs around 2 to 5 iterations on these datasets until convergence.

Conclusions

In this paper, we propose a novel unified graph and low-rank tensor learning for multi-view clustering. View-specific affinity matrix is learned based on projected graph learning. By reorganizing the affinity matrices into tensor, we explore the high-order correlations among views via tensor low-rank approximation. Finally, we unify these two terms together and jointly learn the optimal projection matrices, affinity matrices and low-rank tensor. We also propose an efficient algorithm to optimize the proposed model in an alternating way. We conduct extensive experiments on six challenging datasets to evaluate the performance. Our approach achieves significant improvement over all state-of-the-art methods on nearly all datasets and under six different metrics.

Acknowledgment

J. Wu is supported by the Fundamental Research Funds of Shandong University and SenseTime Research Fund for Young Scholars. L. Nie is supported by the National Natural

Science Foundation (NSF) of China (grant no.s 61772310, 61702300, 61702302, 61802231, and U1836216), the Project of Thousand Youth Talents 2016, the Shandong Provincial Natural Science and Foundation (grant no.s ZR2019JQ23 and ZR2019QF001), and the Future Talents Research Funds of Shandong University (grant no. 2018WLJH 63). Z. Lin is supported by the NSF of China (grant no.s 61625301 and 61731018), Major Scientific Research Project of Zhejiang Lab (grant no.s 2019KB0AC01 and 2019KB0AB02), and Beijing Academy of Artificial Intelligence. H. Zha is supported by the National Key Research and Development Program of China (grant no. 2017YFB1002601) and NSF of China (grant no.s 61632003 and 61771026).

References

Arpit, D.; Nwogu, I.; and Govindaraju, V. 2014. Dimensionality reduction with subspace structure preservation. In *NeurIPS*, 712–720.

Asuncion, A., and Newman, D. 2007. UCI machine learning repository.

Cao, X.; Zhang, C.; Fu, H.; Liu, S.; and Zhang, H. 2015. Diversity-induced multi-view subspace clustering. In *IEEE CVPR*, 586–594.

Carroll, J. D., and Chang, J.-J. 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika* 35(3):283–319.

Elhamifar, E., and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE TPAMI* 35(11):2765–2781.

Harshman, R. 1970. Foundations of the parafac procedure: Models and conditions for an explanatory multimodal factor analysis.

Hu, W.; Tao, D.; Zhang, W.; Xie, Y.; and Yang, Y. 2016. The twist tensor nuclear norm for video completion. *IEEE TNNLS* 28(12):2961–2973.

Huang, B.; Mu, C.; Goldfarb, D.; and Wright, J. 2014. Provable low-rank tensor recovery. *Optimization-Online* 4252:2.

Kilmer, M. E.; Braman, K.; Hao, N.; and Hoover, R. C. 2013. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications* 34(1):148–172.

Kong, H.; Xie, X.; and Lin, Z. 2018. t-schatten-p norm for low-rank tensor recovery. *IEEE JSTSP* 12(6):1405–1419.

Kumar, A., and Daumé, H. 2011. A co-training approach for multi-view spectral clustering. In *ICML*, 393–400.

Kumar, A.; Rai, P.; and Daumé, H. 2011. Co-regularized multi-view spectral clustering. In *NeurIPS*, 1413–1421.

Li, F.-F., and Pietro, P. 2005. A bayesian hierarchical model for learning natural scene categories. In *IEEE CVPR*, 524–531.

Li, F.-F.; Rob, F.; and Pietro, P. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVIU* 106(1):59–70.

Liu, G., and Zhang, W. 2019. Recovery of future data via convolution nuclear norm minimization. *arXiv preprint arXiv:1909.03889*.

Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; and Ma, Y. 2013. Robust recovery of subspace structures by low-rank representation. *IEEE TPAMI* 35(1):171–184.

Liu, G.; Xu, H.; Tang, J.; Liu, Q.; and Yan, S. 2015. A deterministic analysis for LRR. *IEEE TPAMI* 38(3):417–430.

Liu, G.; Liu, Q.; and Li, P. 2016. Blessing of dimensionality: Recovering mixture data via dictionary pursuit. *IEEE TPAMI* 39(1):47–60.

Lu, C.; Feng, J.; Chen, Y.; Liu, W.; Lin, Z.; and Yan, S. 2016. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In *IEEE CVPR*, 5249–5257.

Lu, C.; Feng, J.; Lin, Z.; Mei, T.; and Yan, S. 2018. Subspace clustering by block diagonal representation. *IEEE TPAMI* 41(2):487–501.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *NeurIPS*, 849–856.

Nie, F.; Wang, X.; Jordan, M. I.; and Huang, H. 2016. The constrained laplacian rank algorithm for graph-based clustering. In *AAAI*, 1969–1976.

Nie, F.; Cai, G.; and Li, X. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *AAAI*, 2408–2414.

Nie, F.; Wang, X.; and Huang, H. 2014. Clustering and projected clustering with adaptive neighbors. In *ACM KDD*, 977–986.

Quattoni, A., and Torralba, A. 2009. Recognizing indoor scenes. In *IEEE CVPR*, 413–420.

Tucker, L. R. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31(3):279–311.

Wang, X.; Guo, X.; Lei, Z.; Zhang, C.; and Li, S. Z. 2017. Exclusivity-consistency regularized multi-view subspace clustering. In *IEEE CVPR*, 923–931.

Wang, Y.; Yin, W.; and Zeng, J. 2019. Global convergence of admm in nonconvex nonsmooth optimization. *Journal of Scientific Computing* 78(1):29–63.

Wu, J.; Long, K.; Wang, F.; Qian, C.; Li, C.; Lin, Z.; and Zha, H. 2019. Deep comprehensive correlation mining for image clustering. In *IEEE ICCV*, 8150–8159.

Wu, J.; Lin, Z.; and Zha, H. 2019. Essential tensor learning for multi-view spectral clustering. *IEEE TIP* 28(12):5910–5922.

Xia, R.; Pan, Y.; Du, L.; and Yin, J. 2014. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, 2149–2155.

Xie, Y.; Tao, D.; Zhang, W.; Liu, Y.; Zhang, L.; and Qu, Y. 2018. On unifying multi-view self-representations for clustering by tensor multi-rank minimization. *IJCV* 126(11):1157–1179.

Xie, X.; Wu, J.; Liu, G.; Zhong, Z.; and Lin, Z. 2019. Differentiable linearized admm. In *ICML*, 6902–6911.

Zhang, Y.-F.; Xu, C.; Lu, H.; and Huang, Y.-M. 2009. Character identification in feature-length films using global face-name matching. *IEEE TMM* 11(7):1276–1288.

Zhang, Z.; Ely, G.; Aeron, S.; Hao, N.; and Kilmer, M. 2014. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *IEEE CVPR*, 3842–3849.

Zhang, C.; Fu, H.; Liu, S.; Liu, G.; and Cao, X. 2015. Low-rank tensor constrained multiview subspace clustering. In *IEEE ICCV*, 1582–1590.

Zhang, C.; Hu, Q.; Fu, H.; Zhu, P.; and Cao, X. 2017. Latent multi-view subspace clustering. In *IEEE CVPR*, 4279–4287.

Zhang, C.; Fu, H.; Hu, Q.; Cao, X.; Xie, Y.; Tao, D.; and Xu, D. 2018. Generalized latent multi-view subspace clustering. *IEEE TPAMI*.

Zhou, P., and Feng, J. 2017. Outlier-robust tensor PCA. In *IEEE CVPR*, 3938–3946.