2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) | 978-1-6654-6819-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/BIBM55620.2022.9995487

Accurate Multi-view Clustering by Exploiting Within-view High-order Affinities through Tensor Self-representation

Haiyan Wang[1], Jiazhou Chen[2], Bin Zhang[2], Hongmin Cai[2,*]
[1]*College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China*
[2]*School of Computer Science and Engineering, South China University of Technology, Guangzhou, China*
[*]Corresponding author: hmcai@scut.edu.cn

***Abstract*—Multi-view clustering divides data into their underlying partitions by exploiting multiple views information. Popular approaches leverage cross-view information by self-expressive tensor learning and then learn a low-rank or sparse essential representation tensor for capturing the global structure of multi-view data. However, this process may encounter instability due to the lack of protection for local within-view structures. To overcome this problem, this paper proposes a unified *L*ow-rank and *H*yper*G*raph Laplacian regularized *T*ensor learning (LHGT) method for multi-view clustering, which aims to integrate within-view high-order affinities in self-expressive tensor learning for capturing inherent clustering structure. LHGT effectively extracts global cross-view and local within-view high-order statistics. An effective optimization procedure is tailored for the proposed model. Experimental results on six real-world datasets illustrate the efficacy of LHGT, where a clear advance over nine state-of-the-art approaches.**

***Index Terms*—Multi-view clustering, low-rank, hypergraph-Laplacian, self-expressive tensor learning, subspace representation.**

I. INTRODUCTION

With advances in technology, an object can be readily obtained to describe from distinct sources. It remains a challenge for multi-view clustering [1]–[6] to learn an intact and informative representation by analyzing all the views.

Considerable efforts for multi-view clustering generally fall into two streams [7]: matrix-based and tensor-based representation learning. The matrix-oriented method organizes the data in matrix form, which analyzes the observation based on matrix operations. For example, Cao et al. [8] utilized the Hilbert-Schmidt independence criterion to enforce subspace representation for diversity-induced multiview clustering. Zhang et al. [9] introduced a latent multi-view subspace clustering method by concurrently seeking the underlying representation and data reconstruction. Tan et al. [10] proposed a multi-view clustering framework with sparse and low-rank structure by constructing two new data matrix decomposition models into a unified optimization model. Yang et al. [11] introduced multiplicative decomposition and variable splitting schemes for multi-view subspace clustering. Zhang et al. [12] achieved clustering by focusing on cross-modality feature-matching rather than sample-wise affinity. Although the before-mentioned algorithms have been proved their efficiency, an obvious limitation of these algorithms is that they fail to catch the global high-order information [13], [14] among multiple views. In this regard, recent advances have shifted the focus to tensor-based clustering algorithms [13]–[16]. For example, Zhang et al. [13] proposed a low-rank tensor constrained multiview subspace clustering, which employed the tensor to explore the high-order correlations. Cheng et al. [15] developed a low-dimensional representation learning method for multi-view clustering based on self-expressive tensor learning and low-dimensional representation learning. Yin et al. [14] introduced a novel subspace clustering method over the tensor data based on low-rank self-expressive tensor learning and the Markov chain based spectral clustering. Besides, Chen et al. [16] proposed a nonlinear multi-view clustering method which aims to jointly learn the kernel representation tensor and affinity matrix. In contrast, they build up high-order statistics in tensor space that are conducive to clustering. However, these methods only emphasize the across-view global consensus but ignore the within-view local geometrical structure during the self-expressive tensor learning.

Recently, our preliminary work [17] has taken maintaining within-view local structures into account in self-expressive tensor learning. For convenience, we term this method tensor-based low-rank and graph regularized representation learning (TLGRL). However, TLGRL focuses on the pairwise (sample-to-sample) local geometrical structure and such local geometrical structure is not updated iteratively, resulting in the deterioration of clustering performance in some applications owning to outliers or noise. To mitigate this issue, we introduce an improved version in this paper, by introducing a hyper-Laplacian regularization term to conserve the high-order local structures embedded in self-expressive tensor space, which shows more stable and better performance than TLGRL.

Inspired by the recent success of hypergraph induced hyper-Laplacian regularization [18] in multilinear multi-view self-representation model, we propose a novel essential self-expressive tensor learning model, namely LHGT, for clustering

This work was supported in part by the National Natural Science Foundation of China (U21A20520, 62172112, 62202176), the Key-Area Research and Development of Guangdong Province (2022A0505050014, 2020B1111190001), the Key-Area Research and Development Program of Guangzhou City (202206030009), and the National Key Research and Development Program of China (2022YFE0112200).

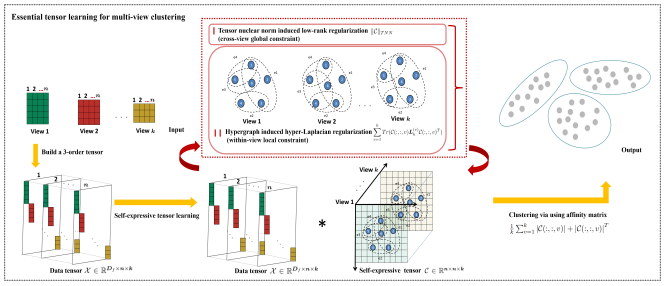


Fig. 1: Illustration of our proposed **LHGT**. For multi-view data $\{\boldsymbol{X}^{(v)} \in \mathbb{R}^{d_v \times n}\}$ $(v = 1, 2, \cdots, k)$ with k views, where d_v is the dimension of feature, n is the sample size. By diagonally arranging view-specific features and $twist$ manipulation, each sample can be covert into a $D_f \times 1 \times k$ tensor ($D_f = \sum_{v=1}^{k} d_v$). Thus, we can easily use a third-order tenor $\mathcal{X} \in \mathbb{R}^{D_f \times n \times k}$ to represent multi-view data. Our algorithm encourages a low-rank hyper-Laplacian regularized self-expressive tensor $\mathcal{C} \in \mathbb{R}^{n \times n \times k}$ which helps to preserve the most informative structure in multiple views.

tasks. Beyond sample-wise pairwise affinities, the hypergraph takes the connections among multiple samples (samples-to-samples) into consideration, thereby catching high-order relationship of the data locality [18]. The backbone of our model is a tensor-based representation learning, which incorporates the tensor low-rank regularization and hyper-Laplacian regularization to simultaneously capture both global across-view and local within-view high-order correlations for boosting the performance of clustering. **Fig. 1** shows the pipeline of the proposed LHGT method.

The highlights of this paper lies in three aspects:

- Based on TLGRL, we propose a novel essential self-expressive tensor learning method (LHGT), which incorporates global across-view and local within-view high-order correlations for multi-view clustering. The former is driven by tensor low-rank regularization, while the latter is driven by hyper-Laplacian regularization. LHGT shows more stable and better performance than our early work TLGRL.
- LHGT explores the effectiveness of sample-wise high-order affinities in improving tensor self-representation learning. These high-order affinities will be effectively and iteratively updated along with the optimization process.
- Numerical results on six benchmark datasets illustrate the superior performance of our proposed algorithm.

II. Low-rank and HyperGraph Laplacian Regularized Tensor Learning (LHGT)

A. The proposed LHGT

The proposed method is motivated by the self-expressive tensor learning model (TLGRL) [17] and the hyper-Laplacian regularization [18]. Even with promising performance, the TLGRL [17] relies on a pairwise relationship and is vulnerable to noise corruptions. In this section, we propose an improved version, by integrating sample-wise high-order affinities in ameliorating the performance of tensor self-representation, for multi-view clustering. Specifically, we proposed a novel essential tensor learning method (LHGT). The method modeled low-rank and hyper-Laplacian regularized self-expressive tensor learning into a united optimization problem. Our LHGT considers the original multi-view data as a $D_f \times n \times k$ third-order tenor $\mathcal{X}$, as shown in **Fig. 1**. Thus, we formulate a low-rank representation learning model [17] as follows:

$$\min_{\mathcal{C}} \frac{1}{2}\|\mathcal{X} - \mathcal{X} * \mathcal{C}\|_F^2 \ + \ \theta\|\mathcal{C}\|_{TNN}, \tag{1}$$

where $\mathcal{C} \in \mathbb{R}^{n \times n \times k}$ represents the self-expressive tensor, each frontal slice $\{\mathcal{C}(:,:,v)\}$ $(v = 1, 2, \cdots, k)$ in $\mathcal{C}$ denotes the self-representation matrix of individual view. $\|\cdot\|_{TNN}$ represents a tensor nuclear norm, which encourages a low-rank representation tensor. θ is an trade-off parameter.

Inspired by the hypergraph induced hyper-Laplacian regularization in [18] , one may employ a view-specific **hyper-Laplacian regularization** (i.e., local invariance constraint) to capure the high-order local geometrical structure. Specifically, given a hypergraphy $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$, $\boldsymbol{V}$ and $\boldsymbol{E}$ are the vertice and the hyperedge sets respectively. $\boldsymbol{W}$ denotes the weight matrix containing the weights of hyperedges. The hyper-Laplacian matrix [18] is defined as follows:

$$\boldsymbol{L}_h = \boldsymbol{D_V} - \boldsymbol{H}\boldsymbol{W}\boldsymbol{D_E}^{-1}\boldsymbol{H}^T, \tag{2}$$

where $\boldsymbol{D_V}$ and $\boldsymbol{D_E}$ represent diagonal matrices corresponding to the vertex and hyperedge degrees, respectively. The incidence matrix $\boldsymbol{H}$ whose elements denote the relationship between $\boldsymbol{V}$ and $\boldsymbol{E}$. We should emphasize that hypergraph is a graph that each edge can contain more than two vertices (i.e., high-order correlation) [19], while each edge of a simple graph contains only two vertices (i.e., pairwise correlation). Because of the existence of high-order relationships in the real world, we consider using hyper-Laplacian regularization to characterize and maintain local high-order affinities. To incorporate (1) and (2) together, the proposed model can be formulated as follows:

$$\begin{aligned} \min_{\mathcal{C}} \ & \frac{1}{2}\|\mathcal{X} - \mathcal{X} * \mathcal{C}\|_F^2 \ + \ \theta\|\mathcal{C}\|_{TNN} \\ & + \alpha \sum_{v=1}^{k} Tr(\mathcal{C}(:,:,v)\boldsymbol{L}_h^{(v)}\mathcal{C}(:,:,v)^T), \end{aligned} \tag{3}$$

where $\boldsymbol{L}_h^{(v)}$ denotes the hyper-Laplacian matrix of v-th view built on individual self-representation matrix $\mathcal{C}(:,:,v)$. The first term encourages minimum fitting error. The next two terms penalize the tensor representation hoping to impose global and local constraints on the estimated subspaces. The former is driven by tensor nuclear norm induced low-rank constraint, while the latter is driven hyper-Laplacian regularization. They are balanced by parameters θ and α.

The proposed LHGT combines global across-view and local within-view high-order correlations. Thus, the optimal projection matrices and the self-expressive tensor corresponding to multi-view data can be effectively caught.

B. Optimization of LHGT

Due to the inseparable property of variable $\mathcal{C}$ in (3), we adopt the variable-splitting technique and introduce two auxiliary tensor variables $\mathcal{Z}$ and $\mathcal{B}$. Then, the proposed optimization problem (3) becomes

$$\begin{aligned}\min_{\mathcal{Z},\mathcal{B},\mathcal{C}} \ & \frac{1}{2}\|\mathcal{X}-\mathcal{X}*\mathcal{C}\|_F^2 \ + \ \theta\|\mathcal{Z}\|_{TNN} \\ & + \ \alpha\sum_{v=1}^{k} Tr(\mathcal{B}(:,:,v)\boldsymbol{L}_h^{(v)}\mathcal{B}(:,:,v)^T) \\ s.t. \ & \mathcal{Z}=\mathcal{C}, \mathcal{B}=\mathcal{C}.\end{aligned} \tag{4}$$

Using the ALM methodology [20], the corresponding augmented Lagrangian function of (4) is obtained by

$$\begin{aligned}\min_{\mathcal{Z},\mathcal{B},\mathcal{C}} \ & \frac{1}{2}\|\mathcal{X}-\mathcal{X}*\mathcal{C}\|_F^2 \ + \ \theta\|\mathcal{Z}\|_{TNN} \\ & + \ \alpha\sum_{v=1}^{k} Tr(\mathcal{B}(:,:,v)\boldsymbol{L}_h^{(v)}\mathcal{B}(:,:,v)^T) \\ & + \ \frac{\mu}{2}(\|\mathcal{Z}-\mathcal{C}+\frac{1}{\mu}\mathcal{G}_1\|_F^2+\|\mathcal{B}-\mathcal{C}+\frac{1}{\mu}\mathcal{G}_2\|_F^2),\end{aligned} \tag{5}$$

where $\mu>0$ is a penalty parameter, and $\{\mathcal{G}_i\}_{i=1}^2$ are Lagrange multipliers.

Let $\mathcal{D}_1=\mathcal{C}-(1/\mu)\mathcal{G}_2$, the problem (6) can be transformed into the following form:

$$\min_{\mathcal{B}} \ \alpha\sum_{v=1}^{k} Tr(\mathcal{B}(:,:,v)\boldsymbol{L}_h^{(v)}\mathcal{B}(:,:,v)^T) \ + \ \frac{\mu}{2}\|\mathcal{B}-\mathcal{D}_1\|_F^2. \tag{6}$$

Let $\mathcal{B}^{(v)}=\mathcal{B}(:,:,v)$, and according to [14], variable $\mathcal{B}$ can be optimized by solving for each of its frontal slices:

$$\min_{\mathcal{B}^{(v)}} \ \alpha Tr(\mathcal{B}^{(v)}\boldsymbol{L}_h^{(v)}\mathcal{B}^{(v)^T}) \ + \ \frac{\mu}{2}\|\mathcal{B}^{(v)}-\mathcal{C}^{(v)}+(1/\mu)\mathcal{G}_2^{(v)}\|_F^2. \tag{7}$$

Taking derivation w.r.t $\mathcal{B}^{(v)}$ and letting it be zero, we have the following close-form solution:

$$vec(\mathcal{B}^{(v)}) \ = \ \boldsymbol{M}_1^{-1}\boldsymbol{M}_2, \tag{8}$$

where $\boldsymbol{M}_1=\Big(\boldsymbol{I}\otimes((\mu/2)\boldsymbol{I})+(\alpha\boldsymbol{L}_h^{(v)})^T\otimes\boldsymbol{I}\Big)$, and $\boldsymbol{M}_2=vec((\mu/2)\mathcal{D}_1^{(v)})$. The $vec(\cdot)$ denotes the vec-operator of a matrix, which stacks the columns of a matrix and converts them into a vector. $\boldsymbol{I}$ denotes an identity matrix.

$\mathcal{Z}$ sub-problem:

$$\min_{\mathcal{Z}} \ \theta\|\mathcal{Z}\|_{TNN} \ + \ \frac{\mu}{2}\|\mathcal{Z}-\mathcal{C}+\frac{1}{\mu}\mathcal{G}_1\|_F^2 \tag{9}$$

Let $\|\mathcal{Z}\|_{TNN}=\ (1/k)\sum_{v=1}^{k}\|\hat{\mathcal{Z}}(:,:,v)\|_*$, and according to the Parseval's theorem [14], [21] (the quadratic sum of a function is equal to the quadratic sum of its Fourier transform), we have

$$\min_{\mathcal{Z}} \frac{\theta}{k}\sum_{v=1}^{k}\|\hat{\mathcal{Z}}(:,:,v)\|_* \ + \ \frac{\mu}{2k}\|\hat{\mathcal{Z}}-\hat{\mathcal{C}}+\frac{1}{\mu}\hat{\mathcal{G}}_1\|_F^2. \tag{10}$$

Similarly, variable $\hat{\mathcal{Z}}$ can be efficiently slice-by-slice optimized from the frontal side:

$$\min_{\hat{\mathcal{Z}}^{(v)}} \ \theta\|\hat{\mathcal{Z}}^{(v)}\|_* \ + \ \frac{\mu}{2}\|\hat{\mathcal{Z}}^{(v)}-\hat{\mathcal{C}}^{(v)} \ + \ \frac{\hat{\mathcal{G}}_1^{(v)}}{\mu}\|_F^2, \tag{11}$$

The minimizer solution of problem [22] is:

$$\hat{\mathcal{Z}}^{(v)}=\mathcal{P}_{\frac{\theta}{\mu}}\left(\hat{\mathcal{C}}^{(v)}-\frac{\hat{\mathcal{G}}_1^{(v)}}{\mu}\right), \tag{12}$$

where for any matrix $\boldsymbol{Y}$ and its singular value decomposition $\boldsymbol{Y}=\boldsymbol{U}\Sigma\boldsymbol{V}^T$, the singular value thresholding (SVT) operator is provided as

$$\mathcal{P}_\tau(\boldsymbol{Y})=\boldsymbol{U}\mathcal{S}_\tau(\Sigma)\boldsymbol{V}^T, \quad \mathcal{S}_\tau(\Sigma)=\max(\Sigma-\tau,0) \tag{13}$$

where $\mathcal{S}_\tau(\Sigma)$ denotes the element-wise soft thresholding operator. Once we obtained $\hat{\mathcal{Z}}^{(v)}$ and $\hat{\mathcal{Z}}$, the $\mathcal{Z}^{\iota+1}$ can be sloved by using inverse FFT transform, $\mathcal{Z}^{\iota+1}=ifft(\hat{\mathcal{Z}}^{\iota+1},[\,],3)$.

$\mathcal{C}$ sub-problem:

$$\begin{aligned}\min_{\mathcal{C}} \ & \frac{1}{2}\|\mathcal{X}-\mathcal{X}*\mathcal{C}\|_F^2 \\ & + \ \frac{\mu}{2}(\|\mathcal{Z}-\mathcal{C}+\frac{1}{\mu}\mathcal{G}_1\|_F^2+\|\mathcal{B}-\mathcal{C}+\frac{1}{\mu}\mathcal{G}_2\|_F^2).\end{aligned} \tag{14}$$

Let $\mathcal{D}_2=\mathcal{Z}+\frac{1}{\mu}\mathcal{G}_1$ and $\mathcal{D}_3=\mathcal{B}+\frac{1}{\mu}\mathcal{G}_2$, the minimization of problem (14) can be rewritten as:

$$\min_{\hat{\mathcal{C}}} \ \frac{1}{2}\|\hat{\mathcal{X}}-\hat{\mathcal{X}}\odot\hat{\mathcal{C}}\|_F^2 \ + \ \frac{\mu}{2}\|\hat{\mathcal{C}}-\hat{\mathcal{D}}_2\|_F^2 \ + \ \frac{\mu}{2}\|\hat{\mathcal{C}}-\hat{\mathcal{D}}_3\|_F^2, \tag{15}$$

where $\odot$ denotes point-wise multiplication. Similarly, we can optimize each frontal slice of $\hat{\mathcal{C}}$ individually by solving the following minimization problem:

$$\begin{aligned}\min_{\hat{\mathcal{C}}^{(v)}} \ & \frac{1}{2}\|\hat{\mathcal{X}}^{(v)}-\hat{\mathcal{X}}^{(v)}\hat{\mathcal{C}}^{(v)}\|_F^2 \\ & + \ \frac{\mu}{2}\|\hat{\mathcal{C}}^{(v)}-\hat{\mathcal{D}}_2^{(v)}\|_F^2 \ + \ \frac{\mu}{2}\|\hat{\mathcal{C}}^{(v)}-\hat{\mathcal{D}}_3^{(v)}\|_F^2.\end{aligned} \tag{16}$$

Computing the derivative of Eq.(16) w.r.t $\hat{\mathcal{C}}^{(v)}$ and letting it be zero, we have the following close-form solution:

$$\hat{\mathcal{C}}^{(v)}= \ \boldsymbol{M}_3^{-1}\boldsymbol{M}_4, \tag{17}$$

where $\boldsymbol{M}_3 \ = \ \Big((\hat{\mathcal{X}}^{(v)})^T\hat{\mathcal{X}}^{(v)} \ + \ 2\mu\boldsymbol{I}\Big)$, and $\boldsymbol{M}_4 \ = \ (\hat{\mathcal{X}}^{(v)})^T\hat{\mathcal{X}}^{(v)} \ + \ \mu\hat{\mathcal{D}}_2^{(v)} \ + \ \mu\hat{\mathcal{D}}_3^{(v)}$. After obtaining $\hat{\mathcal{C}}^{(v)}$ and $\hat{\mathcal{C}}$, it is easy to compute $\mathcal{C}^{\iota+1}$ via inverse FFT.

Update $\mathcal{G}_1$ and $\mathcal{G}_2$ using

$$\begin{aligned}\mathcal{G}_1 &= \mathcal{G}_1 \ + \ \mu(\mathcal{Z}-\mathcal{C}) \\ \mathcal{G}_2 &= \mathcal{G}_2 \ + \ \mu(\mathcal{B}-\mathcal{C}) \\ \mu &= \min(\mu_{\max},\rho\mu).\end{aligned} \tag{18}$$

The complete procedure for solving $\mathcal{C}$ and the steps of our method are summarized in **Algorithm 1**, and the stopping condition is set by $\max(\|\mathcal{Z}^{\iota+1}-\mathcal{C}^{\iota+1}\|_F, \|\mathcal{B}^{\iota+1}-\mathcal{C}^{\iota+1}\|_F, \|\mathcal{Z}^{\iota+1}-\mathcal{Z}^{\iota}\|_F, \|\mathcal{B}^{\iota+1}-\mathcal{B}^{\iota}\|_F, \|\mathcal{C}^{\iota+1}-\mathcal{C}^{\iota}\|_F) \leq \epsilon$.

Algorithm 1 LHGT

Input: Data matrix of all views $\{\boldsymbol{X}^{(v)} \in \mathbb{R}^{d_v \times n}\}$ $(v = 1, 2, \cdots, k)$, number of clusters k, regularization parameters θ, α.

Organize $\{\boldsymbol{X}^{(v)} \in \mathbb{R}^{d_v \times n}\}$ into tensorial data $\mathcal{X} \in \mathbb{R}^{D_f \times n \times k}$ $(D_f = \sum_{v=1}^{k} d_v)$.

Initialization: $\mathcal{C}^0 = \mathcal{B}^0 = \mathcal{Z}^0 = \mathcal{I}$, $\mathcal{G}_1^0 = \mathcal{G}_2^0 = \mathcal{I}$, $\rho = 1.9$, $\epsilon = 10^{-3}$, $\mu_{max} = 10^6$, μ.

while stopping condition is not satisfied $(\iota = 0, 1, \cdots)$

do

  Compute $\boldsymbol{L}_h^{(v)}$ from $\mathcal{B}(:,:,v)$ by using (2)

  Update $\mathcal{B}^{\iota+1}$ by solving (6)

  Update $\mathcal{Z}^{\iota+1}$ by solving (9)

  Update $\mathcal{C}^{\iota+1}$ by solving (14)

  Update $\mathcal{G}_1$, $\mathcal{G}_2$, and μ by solving (18)

  Check the stopping condition

end while

Compute affinity matrix $\boldsymbol{A}^* = \frac{1}{k}\sum_{v=1}^{k} |\mathcal{C}(:,:,v)| + |\mathcal{C}(:,:,v)|^T$.

Apply spectral clustering on $\boldsymbol{A}^*$ to obtain the final clustering labels.

Output: Clustering labels.

C. Complexity Analyses

The computational time of our proposed model mainly taken by updating the $\{L_h^{(v)}\}_{v=1}^k$, $\mathcal{B}$, $\mathcal{Z}$, and $\mathcal{C}$. As mentioned in **Algorithm 1**, $\mathcal{C}$ is optimized via iterative process. In particular, the complexity of constructing hyper-Laplacian $\{L_h^{(v)}\}_{v=1}^k$ is $\mathcal{O}(kn^2log(n))$ [18], where k is the number of views, and n is the number of multi-view samples. For computing $\mathcal{B}$, the main complexity is to perform matrix inversion, which is $\mathcal{O}(n^3)$. Calculating $\mathcal{Z}$ will spend $\mathcal{O}(krn^2)$ [14], in which r is the lowest rank for $\mathcal{Z}$. Computing $\mathcal{C}$ involves a matrix inversion, it will take $\mathcal{O}(n^3)$. Hence, the total computational cost of updating $\mathcal{C}$ with ι iterations is $\mathcal{O}(\iota(n^2(klog(n)+kr+2n))$. In addition, spectral clustering spends $\mathcal{O}(n^3)$. Overall, the total complexity of LHGT is $\mathcal{O}(\iota(n^2(klog(n)+kr+2n))+\mathcal{O}(n^3)$.

III. Experiments and Results

A. Datasets

We evaluate the practicability of LHGT on two news store datasets: BBCSport [23], 3-sources [17]; two face datasets: Yale [8], Notting-Hill [24]; one gene expression datasets: Prokaryotic [25], and one generic object dataset: COIL-20 [26]. The statistics of these datasets are listed in **Table I**.

B. Competitors and evaluation metrics

We compare the proposed method with nine state-of-the-art competitors include five matrix-oriented clustering algorithms (i.e., Co-Reg [27], Co-Pair [27], DiMSC [8], LMSC [9], and SM^{2}SC [11]) and four tensor-based clustering algorithms (i.e., LT-MSC [13], SCMV-3DT [14], ETLMSC [26], and TLGRL [17]). Four well-known criteria are exploited to evaluate the clustering performance, including: accuracy (ACC), normalized mutual information (NMI), F-score, and adjusted rand index (ARI).

TABLE I: Datasets uesd in our experiments

Dataset	Type	♯Instances	♯Clusters
BBCSport	News store	282	5
3-sources	News store	169	6
Yale	Face	165	15
Notting-Hill	Face	550	5
Prokaryotic	Gene Expression	551	4
COIL-20	Generic Object	1440	20

TABLE II: Mean clustering results on three real-world datasets.

Method	**BBCSport** ($\alpha = 2$, $\theta = 1.6$)			
	ACC	NMI	F-score	ARI
Co-Reg [27]	0.497±0.018	0.250±0.002	0.415±0.014	0.160±0.014
Co-Pair [27]	0.491±0.010	0.235±0.002	0.404±0.005	0.131±0.004
DiMSC [8]	0.733±0.004	0.639±0.005	0.667±0.007	0.564±0.009
LMSC [9]	0.798±0.010	0.653±0.015	0.725±0.002	0.637±0.003
SM^{2}SC [11]	0.833±0.000	0.790±0.000	0.827±0.000	0.769±0.000
LT-MSC [13]	0.715±0.008	0.592±0.004	0.632±0.002	0.504±0.001
SCMV-3DT [14]	0.749±0.008	0.573±0.016	0.646±0.010	0.526±0.012
ETLMSC [26]	0.764±0.002	**0.847±0.005**	0.829±0.003	0.777±0.005
TLGRL [17]	0.869±0.000	0.753±0.000	**0.845±0.000**	0.790±0.000
LHGT	**0.890±0.001**	0.757±0.001	0.841±0.002	**0.790±0.002**

Method	**3-sources** ($\alpha = 0.04$, $\theta = 1.2$)			
	ACC	NMI	F-score	ARI
Co-Reg [27]	0.612±0.015	0.582±0.011	0.567±0.014	0.453±0.018
Co-Pair [27]	0.552±0.007	0.518±0.005	0.473±0.004	0.323±0.006
DiMSC [8]	0.802±0.006	0.746±0.010	0.754±0.005	0.678±0.007
LMSC [9]	0.702±0.022	0.668±0.020	0.639±0.020	0.546±0.026
SM^{2}SC [11]	0.763±0.000	0.595±0.000	0.624±0.000	0.528±0.000
LT-MSC [13]	0.784±0.004	0.707±0.013	0.740±0.008	0.660±0.011
SCMV-3DT [14]	0.684±0.013	0.602±0.006	0.654±0.010	0.560±0.013
ETLMSC [26]	0.570±0.010	0.534±0.009	0.499±0.010	0.359±0.013
TLGRL [17]	0.799±0.000	0.742±0.000	0.741±0.000	0.656±0.000
LHGT	**0.873±0.004**	**0.767±0.007**	**0.824±0.002**	**0.774±0.003**

Method	**Yale** ($\alpha = 1$, $\theta = 0.2$)			
	ACC	NMI	F-score	ARI
Co-Reg [27]	0.619±0.000	0.663±0.000	0.480±0.000	0.444±0.000
Co-Pair [27]	0.622±0.000	0.661±0.000	0.486±0.000	0.451±0.000
DiMSC [8]	0.721±0.000	0.736±0.000	0.593±0.000	0.566±0.000
LMSC [9]	0.703±0.022	0.715±0.020	0.537±0.020	0.505±0.026
SM^{2}SC [11]	0.735±0.005	0.772±0.005	0.622±0.004	0.596±0.000
LT-MSC [13]	0.741±0.002	0.765±0.008	0.598±0.006	0.570±0.004
SCMV-3DT [14]	0.606±0.024	0.643±0.014	0.443±0.019	0.405±0.021
ETLMSC [26]	0.633±0.001	0.680±0.000	0.510±0.001	0.475±0.001
TLGRL [17]	0.642±0.000	0.672±0.000	0.511±0.000	0.479±0.000
LHGT	**0.746±0.004**	**0.780±0.004**	**0.635±0.010**	**0.611±0.010**

C. Results and Discussion

Table II shows the detailed experimental results on BBCSport, 3-sources, and Yale datasets. **Table III** shows the detailed clustering results on Notting-Hill, Prokaryotic, and COIL-20 datasets. We recorded the mean and variance of all methods over 20 independent trials. The value in bold indicates optimal performance.

From **Table II** and **Table III**, we come up with the following conclusions. Firstly, the proposed LHGT achieves competitive results on six benchmark datasets. LHGT consistently outperforms all the competitors on the BBCSport, 3-sources, Yale, and Prokaryotic datasets. Taking experiments on 3-sources as an example, we improved around 7.1%, 2.1%, 7.0%, and 9.6% in terms of ACC, NMI, F-score, and ARI when comparing with the suboptimal result, respectively. Meanwhile, LHGT yields the second-best performance on the rest datasets. These results well manifest the effectiveness of

TABLE III: Mean clustering results on three real-world datasets.

Method	Notting-Hill ($\alpha = 2.5$, $\theta = 1.5$)			
	ACC	NMI	F-score	ARI
Co-Reg [27]	0.808±0.000	0.759±0.000	0.783±0.000	0.723±0.000
Co-Pair [27]	0.794±0.005	0.750±0.007	0.777±0.006	0.715±0.008
DiMSC [8]	0.824±0.019	0.802±0.017	0.819±0.021	0.769±0.026
LMSC [9]	0.817±0.054	0.696±0.052	0.701±0.087	0.615±0.112
SM^{2}SC [11]	0.875±0.000	0.763±0.000	0.807±0.000	0.755±0.000
LT-MSC [13]	0.873±0.000	0.767±0.000	0.798±0.000	0.740±0.000
SCMV-3DT [14]	0.814±0.001	0.655±0.001	0.712±0.001	0.623±0.002
ETLMSC [26]	0.865±0.032	0.855±0.017	0.840±0.025	0.795±0.032
TLGRL [17]	**0.960±0.000**	**0.903±0.000**	**0.941±0.000**	**0.925±0.000**
LHGT	0.919±0.001	0.853±0.002	0.883±0.001	0.851±0.002

Method	Prokaryotic ($\alpha = 0.3$, $\theta = 0.75$)			
	ACC	NMI	F-score	ARI
Co-Reg [27]	0.565±0.008	0.267±0.004	0.459±0.007	0.197±0.008
Co-Pair [27]	0.561±0.014	0.338±0.008	0.475±0.012	0.208±0.018
DiMSC [8]	0.381±0.000	0.028±0.000	0.345±0.000	0.036±0.000
LMSC [9]	0.690±0.000	0.400±0.000	0.612±0.000	0.298±0.000
SM^{2}SC [11]	0.566±0.003	0.040±0.000	0.555±0.000	0.098±0.000
LT-MSC [13]	0.412±0.000	0.118±0.000	0.388±0.000	0.023±0.000
SCMV-3DT [14]	0.612±0.003	0.236±0.005	0.611±0.004	0.267±0.005
ETLMSC [26]	0.566±0.000	0.264±0.000	0.459±0.001	0.204±0.006
TLGRL [17]	0.719±0.000	**0.413±0.000**	0.641±0.000	0.327±0.000
LHGT	**0.733±0.001**	0.394±0.001	**0.656±0.001**	**0.383±0.001**

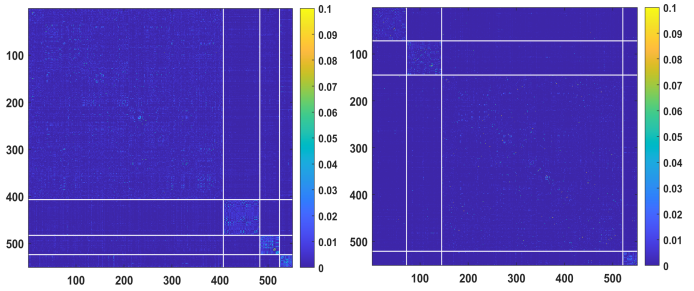
Method	COIL-20 ($\alpha = 7$, $\theta = 4$)			
	ACC	NMI	F-score	ARI
Co-Reg [27]	0.726±0.011	0.823±0.004	0.696±0.009	0.680±0.009
Co-Pair [27]	0.665±0.007	0.783±0.003	0.628±0.006	0.608±0.006
DiMSC [8]	0.784±0.001	0.849±0.002	0.739±0.002	0.725±0.002
LMSC [9]	0.757±0.027	0.854±0.026	0.715±0.051	0.699±0.054
SM^{2}SC [11]	0.814±0.000	0.925±0.000	0.771±0.000	0.758±0.000
LT-MSC [13]	0.724±0.032	0.821±0.001	0.683±0.011	0.666±0.012
SCMV-3DT [14]	0.701±0.028	0.810±0.009	0.654±0.029	0.635±0.003
ETLMSC [26]	**0.873±0.002**	**0.963±0.003**	**0.887±0.002**	**0.881±0.002**
TLGRL [17]	0.811±0.000	0.864±0.000	0.781±0.000	0.770±0.000
LHGT	0.817±0.004	0.873±0.002	0.792±0.006	0.781±0.006

our method. Secondly, compared with the TLGRL method, the proposed LHGT has a significant improvement in clustering results for the most part. For example, LHGT achieves an increasing ratios of 2.1%, 7.4%, 10.4%, −4.1%, 1.4%, and 0.6% in terms of ACC when comparing with TLGRL on the BBCSport, 3-sources, Yale, Nottinghill, Prokaryotic, and COIL-20 datasets, respectively. Thirdly, LHGT obtains a fairly stable performance on all datasets, as illustrated by **Tables II-III**. LHGT yields the optimal or suboptimal results on six datasets. In contrast, the results of some methods are unstable. For instance, ETLMSC attains relatively satisfactory results on COIL-20. However, it did not obtain promising performances on the remaining datasets. Finally, by checking the results, we notice that LHGT boosts the clustering performances on all datasets. The corresponding results are better than those of other popular approaches. The main reason is that LHGT has ability to sufficiently capture the inherent principle components from multiple collaborative views. The LHGT not only positively exploits the local geometrical structure within-view but also effectively captures high-order correlation across-view, thus possessing a more reliable clustering structure.

To be more intuitive, **Fig. 2** compares the affinity matrices on the Prokaryotic dataset derived by TLGRL and LHGT. It shows that their affinity structures are well consistent with the clustering results recorded in **Table II**. More precisely, the LHGT outperformed TLGRL for the dataset Prokaryotic due to its fairly clear diagonal-block structure.

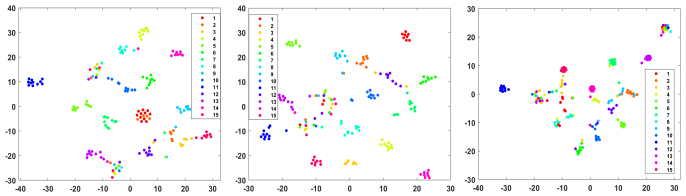
Fig. 3 visualizes the learned latent representation for the dataset Yale derived by various comparative methods with t-distributed stochastic neighbor embedding (t-SNE) [28]. We can see that the LHGT obtained the best separation performance.

On balance, the LHGT has strong competitiveness in the

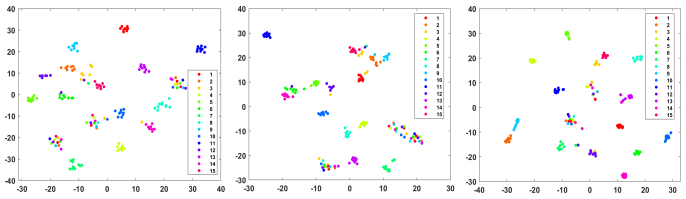


(a) TLGRL (b) LHGT

Fig. 2: Visualization of affinity matrices on the Prokaryotic dateset.



(a) SM^{2}SC (b) LT-MSC (c) SCMV-3DT



(d) ETLMSC (e) TLGRL (f) LHGT

Fig. 3: Visualization of latent representation with t-SNE on the Yale dateset.

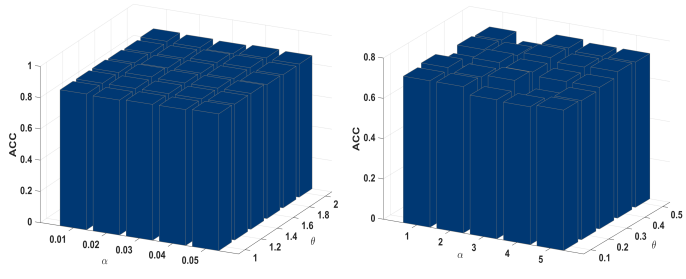
visualization of affinity matrices, the spatial distribution of sample points, and clustering metric. The above experimental results demonstrate the efficacy of LHGT.

D. Parameter Sensitivity Analysis

For the proposed **Algorithm 1**, we empirically set the initial value of μ. Concretely, for 3-sources and Yale, we set $\mu = 10$ and $\mu = 0.2$, respectively. For BBCSport, Notting-Hill, Prokaryotic, and COIL-20, we set $\mu = 2$ throughout all experiments.

In our objective function, there are mainly two parameters, including the parameters α and θ, for balancing the effect of hypergraph Laplacian regularized term and low-rank term. We record the ACC performance of LHGT with respect to the varying hyperparameter α and θ. For space considerations, we take two datasets 3-sources and Yale as examples. The effect of the α and θ on our algorithm for these two datasets are recorded in **Fig. 4**. We test each parameter in turn while keeping another one invariant. One can find that LHGT is insensitive to parameters if the parameters are in a proper

range. According to **Fig. 4 (a)**, we choose $\alpha = 0.04$ and $\theta = 1.2$ for the 3-sources dataset. In view of **Fig. 4 (b)**, we thus select $\alpha = 1$ and $\theta = 0.2$ for the Yale dataset.



(a) Hyperparameter analysis on 3-sources (b) Hyperparameter analysis on Yale

Fig. 4: Parameter sensitivity of the proposed LHGT.

IV. Conclusion

In this article, we present a unified low-rank and hyper-Laplacian regularized tensor learning method (LHGT) for multi-view clustering. The LHGT aims to improve clustering by imposing within-view high-order affinities. LHGT succeeds in catching across-view and within-view high-order statistics to obtain intrinsic components of the data, and therefore the learned latent representation shows better performance. Also, we tailor an effective optimization scheme to solve the proposed optimization problem. The affinity matrix can be directly derived from the self-expressive tensor for clustering. Extensive experiments conducted on six benchmark datasets have showed the effectiveness of the proposed LHGT.

References

[1] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Information Fusion*, vol. 38, pp. 43–54, 2017.

[2] L. Fu, P. Lin, A. V. Vasilakos, and S. Wang, "An overview of recent multi-view clustering," *Neurocomputing*, vol. 402, pp. 148–161, 2020.

[3] X. Xiao, Y. Gong, Z. Hua, and W. Chen, "On reliable multi-view affinity learning for subspace clustering," *IEEE Transactions on Multimedia*, doi:10.1109/TMM.2020.3045259.

[4] S. Wang, Y. Chen, Y. Jin, Y. Cen, Y. Li, and L. Zhang, "Error-robust low-rank tensor approximation for multi-view clustering," *Knowledge-Based Systems*, vol. 215, p. 106745, 2021.

[5] X. Zhang, H. Sun, Z. Liu, Z. Ren, Q. Cui, and Y. Li, "Robust low-rank kernel multi-view subspace clustering based on the schatten p-norm and correntropy," *Information Sciences*, vol. 477, pp. 430–447, 2019.

[6] H. Wang, G. Han, J. Li, B. Zhang, J. Chen, Y. Hu, C. Han, and H. Cai, "Learning task-driving affinity matrix for accurate multi-view clustering through tensor subspace learning," *Information Sciences*, vol. 563, pp. 290–308, 2021.

[7] H. Wang, G. Han, B. Zhang, G. Tao, and H. Cai, "Multi-view learning a decomposable affinity matrix via tensor self-representation on grassmann manifold," *IEEE Transactions on Image Processing*, vol. 30, pp. 8396–8409, 2021.

[8] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 586–594.

[9] C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao, "Latent multi-view subspace clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4279–4287.

[10] J. Tan, Y. Shi, Z. Yang, C. Wen, and L. Lin, "Unsupervised multi-view clustering by squeezing hybrid knowledge from cross view and each view," *IEEE Transactions on Multimedia*, doi:10.1109/TMM.2020.3019683.

[11] Z. Yang, Q. Xu, W. Zhang, X. Cao, and Q. Huang, "Split multiplicative multi-view subspace clustering," *IEEE Transactions on Image Processing*, vol. 28, no. 10, pp. 5147–5160, 2019.

[12] B. Zhang, H. Cai, J. Chen, Y. Hu, J. Huang, W. Rong, W. Weng, Q. Huang, H. Wang, and H. Peng, "Fast and accurate clustering of multiple modality data via feature matching," *IEEE Transactions on Cybernetics*, pp. 1–11., 2020.

[13] C. Zhang, H. Fu, S. Liu, G. Liu, and X. Cao, "Low-rank tensor constrained multiview subspace clustering," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1582–1590.

[14] M. Yin, J. Gao, S. Xie, and Y. Guo, "Multiview subspace clustering via tensorial t-product representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 851–864, 2018.

[15] M. Cheng, L. Jing, and M. K. Ng, "Tensor-based low-dimensional representation learning for multi-view clustering," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2399–2414, 2018.

[16] Y. Chen, X. Xiao, and Y. Zhou, "Jointly learning kernel representation tensor and affinity matrix for multi-view clustering," *IEEE Transactions on Multimedia*, vol. 22, no. 8, pp. 1985–1997, 2019.

[17] H. Wang, G. Han, B. Zhang, Y. Hu, H. Peng, C. Han, and H. Cai, "Tensor-based low-rank and graph regularized representation learning for multi-view clustering," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 821–826.

[18] Y. Xie, W. Zhang, Y. Qu, L. Dai, and D. Tao, "Hyper-laplacian regularized multilinear multiview self-representations for clustering and semisupervised learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 572–586, 2018.

[19] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Advances in Neural Information Processing Systems*, 2007, pp. 1601–1608.

[20] Z. Wen, D. Goldfarb, and W. Yin, "Alternating direction augmented lagrangian methods for semidefinite programming," *Mathematical Programming Computation*, vol. 2, no. 3-4, pp. 203–230, 2010.

[21] G. B. Arfken and H. J. Weber, "Mathematical methods for physicists," 1999.

[22] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[23] W. Weng, W. Zhou, J. Chen, H. Peng, and H. Cai, "Enhancing multi-view clustering through common subspace integration by considering both global similarities and local structures," *Neurocomputing*, vol. 378, pp. 375–386, 2020.

[24] Z. Tao, H. Liu, S. Li, Z. Ding, and Y. Fu, "From ensemble clustering to multi-view clustering," in *in: Twenty-sixth International Joint Conference on Artificial Intelligence (IJCAI)*, 2017, pp. 2843–9.

[25] M. Brbić and I. Kopriva, "Multi-view low-rank sparse subspace clustering," *Pattern Recognition*, vol. 73, pp. 247–258, 2018.

[26] J. Wu, Z. Lin, and H. Zha, "Essential tensor learning for multi-view spectral clustering," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5910–5922, 2019.

[27] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," in *Advances in Neural Information Processing Systems*, 2011, pp. 1413–1421.

[28] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.