Tensorized Multi-View Subspace Representation Learning

Changqing Zhang · Huazhu Fu · Jing Wang · Qinghua Hu · Xiaochun Cao · Wen Li

Abstract Self-representation based subspace learning has shown its effectiveness in many applications. In this paper, we promote the traditional subspace representation learning by simultaneously taking advantages of multiple views and prior constraint. Accordingly, we establish a novel algorithm termed as Tensorized Multi-View Subspace Representation Learning (TMSRL). To exploit different views, the subspace representation matrices of different views are regarded as a low-rank tensor, which effectively models the high-order correlations of multi-view data. To incorporate prior information, a constraint matrix is devised to guide the subspace representation learning within a unified framework. The subspace representation tensor equipped with a low-rank constraint models elegantly the complementary information among different views, reduces redundancy of subspace representations, and then improves the accuracy of subsequent tasks. We formulate the model with a tensor nuclear norm minimization problem constrained with $\ell_{2,1}$-norm and linear equalities. The minimization problem is efficiently solved by using an Augmented Lagrangian Alternating Direction Minimization (AL-ADM) method. Extensive experimental results on diverse multi-view datasets demonstrate the effectiveness of our algorithm.

Keywords Multi-view representation learning · Subspace clustering · Low-rank tensor · Constraint matrix

1 Introduction

Recently, data collected from various sources or represented by different types of features are available in many real-world applications. For images, different types of features are usually extracted based on color, texture and edge. For web pages, different types of features could be extracted based on text, hyperlinks and possibly existing visual information. These different types of information can be considered as different views describing subjects. Different views describe samples from different perspectives, hence it is beneficial to integrate the information from multiple views for more comprehensive learning [5, 6, 9, 10, 12, 13, 21, 22, 35, 44, 54, 65, 73]. Moreover, real-world data are usually associated with prior such as label information, which, if being utilized, can improve the discriminability of representation. To utilize these two cues, in this work, we focus on advancing representation learning by making use of multiple views and prior constraint within a unified framework.

C. Zhang
the School of Computer Science and Technology, Tianjin University, Tianjin , 300072, China
E-mail: zhangchangqing@tju.edu.cn

H. Fu
Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
E-mail: huazhufu@gmail.com

J. Wang
Graduate School of Information Science and Technology, The University of Tokyo, Japan
E-mail: jing_wang@mist.i.u-tokyo.ac.jp

Q. Hu
the School of Computer Science and Technology, Tianjin University, Tianjin , 300072, China
E-mail: huqingqing@tju.edu.cn

X. Cao
the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China
E-mail: caoxiaochun@iie.ac.cn

W. Li
the Computer Vision Laboratory, ETH, Zurich
E-mail: liwenbnu@gmail.com

Most existing multi-view clustering methods exploit different views with graph-based models. Typically, some early approaches address the "2-view" case [5, 12, 21]. The method in [21] relates two views with a bipartite graph and the final clustering result is obtained by using a standard spectral clustering algorithm. The method in [5] focuses on handling the data with two conditionally independent views based on k-means clustering. To be applicable for the data with three or more views, Linked Matrix Factorization (LMF) [54] fuses the information from multiple graphs, where a common factor is shared by all graphs and a view-specific factor is assigned to each individual graph. Some methods [34, 35] co-regularize or co-train different views to enforce the consistence among multiple views. The common space based models [6, 13] usually focus on learning a common representation by using Canonical Correlation Analysis (CCA) to project multiple views onto a low-dimensional common subspace, and followed by conventional clustering algorithms. Recently, some methods [59, 66] were proposed for multi-graph fusion with rank constraint on its Laplacian matrix, thus the cluster indicators are directly obtained by the global graph without performing any post processing (*e.g.*, k-means clustering). Based on self-representation subspace learning, several multi-view subspace learning methods [9, 68, 70] were proposed, which usually jointly learn multiple subspace representation matrices or one unified subspace representation matrix.

Although great progress has been achieved, there are still two limitations for existing methods: (1) previous approaches usually capture pairwise correlations of different views, ignoring the essentially high-order relationship of multi-view data; (2) for multi-view representation learning, there are usually prior constraints (*e.g.*, must-link constraint or partial label information) which could improve the learned multi-view representation. However, this is not guaranteed in existing multi-view clustering approaches.

To address these issues, we propose a novel method termed Tensorized Multi-View Subspace Representation Learning (TMSRL), which is outlined in Fig. 1. The whole procedure includes the following two aspects. Firstly, the proposed TMSRL regards the subspace representations of all views as a high-order structure, *i.e.*, a 3-order tensor. To model the high-order cross different views, the tensor is enforced to be low-rank to enhance the consistency and reduce the redundancy of these multiple subspace representations. Secondly, to incorporate the prior information thus guide the representation learning, a constraint matrix is introduced into our framework. Therefore, the learned representation could be beneficial from both the complementarity of multiple views and the effective prior constraint. Notably, in our model, the high-order correlation indicates the linear correlation by simultaneously considering all views instead of pairwise manner. The well-known Canonical Correlation Analysis (CCA) and its variants are designed for multi-view representation learning by maximizing the sum of pairwise correlations. Although this is a popular way in multi-view learning, it fails to incorporate higher-order correlations. The main contributions are summarized as follows:

- With integrating together all the subspace representations of different views by low-rank tensor, the proposed TMSRL captures the global structure of all views, and explores the correlations within each view and across multiple views. The proposed algorithm levels up the conventional multi-view learning which can only explore pairwise correlation.
- With a constraint matrix with labels as hard constraint, TMSRL guarantees the data with the same label to have the same subspace representation, which seamlessly utilizes prior in the unified multi-view subspace representation framework and promotes the subsequent tasks. The strategy of incorporating additional prior information is parameter free, which makes the algorithm more applicable for practical applications.
- Extensive experiments on benchmark datasets demonstrate the effectiveness of exploring the high-order correlations among multiple views, and the effectiveness of incorporating constraint as well.

2 Related Work

2.1 Multi-View Learning

Different categorises of multi-view learning algorithms have been proposed and applied in various applications. For example, graph fusion based methods [54, 56] usually construct multiple graphs for multiple views and then fuse them into a common graph. Co-regularization based methods [35, 58] jointly regularize the hypotheses to explore the complementary information. Co-training based methods [34, 75] search for the results that agree across different views. Multiple Kernel Learning (MKL) methods usually combine different kernels by adding them equally [18] or learning the combination weights [27]. It is also noteworthy that there are also models designed for real-world applications, including video face clustering [11], medical diagnosis [67] and recommendation systems [24].

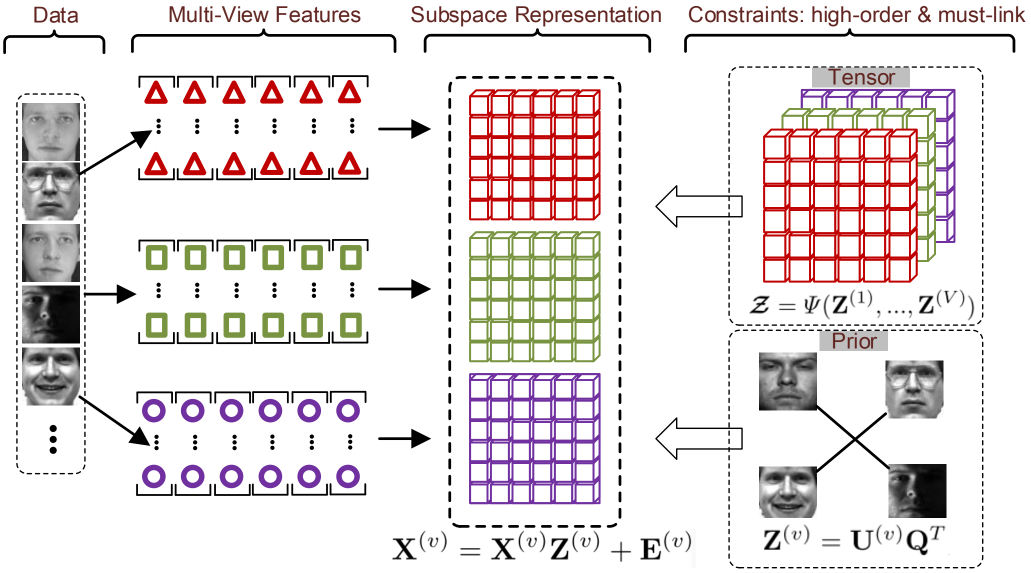

Data
Multi-View Features
Subspace Representation
Constraints: high-order & must-link
Tensor
$\mathcal{Z} = \Psi(\mathbf{Z}^{(1)}, ..., \mathbf{Z}^{(V)})$
Prior
$\mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \mathbf{E}^{(v)}$
$\mathbf{Z}^{(v)} = \mathbf{U}^{(v)}\mathbf{Q}^T$

Figure 1: Overview of Tensorized Multi-View Subspace Representation Learning (TMSRL). Given a collection of data points with multiple views, $\mathbf{X}^{(1)} \cdots \mathbf{X}^{(V)}$, our method integrates all the learned subspace representations, $\mathbf{Z}^{(1)} \cdots \mathbf{Z}^{(V)}$, into a low-rank tensor, $\mathcal{Z}$, under the prior constraints. Then, TMSRL integrates the information from each individual views by exploring high-order correlations and utilizing the prior constraints as well, which jointly promote the multi-view subspace representation.

2.2 Multi-View Representation Learning

For multi-view representation learning, CCA-based algorithms basically maximize the correlation of two different views by mapping the original features into a common space. Generally, CCA can be expressed as an optimization problem over matrix variables as follows

$$(\mathbf{P}^{(1)}, \mathbf{P}^{(2)}) = \underset{\mathbf{P}^{(1)},\mathbf{P}^{(2)}}{\arg\max}\, tr(\mathbf{P}^{(1)^T}\mathbf{X}^{(1)}\mathbf{X}^{(2)^T}\mathbf{P}^{(2)})$$
$$s.t.\ \mathbf{P}^{(v)^T}\mathbf{X}^{(v)}\mathbf{X}^{(v)^T}\mathbf{P}^{(v)} = \mathbf{I}, \quad v = 1, 2.$$

where $\mathbf{X}^{(v)} = [\mathbf{x}_1^{(v)}, \cdots, \mathbf{x}_N^{(v)}] \in \mathbb{R}^{d_v \times n}$ is the feature matrix corresponding to the vth view, with n and d_v being the number of samples and dimensionality for the vth view, respectively. $\mathbf{I}$ is an identity matrix. $\mathbf{P}^{(v)} \in \mathbb{R}^{d_v \times k}$ is the projection matrix for the vth view, and k is the dimensionality of the common space. To address nonlinear correlations, the kernel extension of CCA was proposed. To utilize the neural networks for more general correlations, the Deep CCA [2] jointly learns two deep neural networks (DNN) for different views, and the autoencoder based model [47] aims to obtain a compact representation which can well reconstruct the original input. Similar to CCA based methods, the flexible multi-view dimensionality co-reduction method [69] introduces Hilbert-Schmidt independence criterion (HSIC) to exploit the correlations among different views:

$$(\mathbf{P}^{(1)}, \cdots, \mathbf{P}^{(V)}) =$$
$$\underset{\mathbf{P}_i \in \mathbb{R}^{K^{(v)} \times D^{(v)}}}{\arg\max} \sum_{v=1}^{V} tr(\mathbf{P}^{(v)}\mathbf{X}^{(v)}\mathbf{L}^{(v)}\mathbf{X}^{(v)^T}\mathbf{P}^{(v)^T})$$
$$+ \lambda \sum_{v \neq u} \text{HSIC}(\mathbf{P}^{(v)}\mathbf{X}^{(v)}, \mathbf{P}^{(u)}\mathbf{X}^{(u)}),$$
$$\text{s.t. } \mathbf{P}^{(v)}\mathbf{P}^{(v)^T} = \mathbf{I}, \ \ v = 1, ..., V,$$

where $\mathbf{L}^{(v)}$ is the graph Laplacian for the vth view, and V is the number of views. The hierarchical semi-nonnegative matrix factorization is proposed to obtain the semantics from multi-view data in a layer-wise manner [74]. The common representation of all views is obtained by enforcing the coefficients of different views in the final layer to be the same:

$$(\mathbf{P}_1^{(v)}, \cdots, \mathbf{P}_L^{(v)}, \mathbf{H}) =$$
$$\underset{\mathbf{P}_1^{(v)},\cdots,\mathbf{P}_L^{(v)},\mathbf{H}}{\arg\min} \ = ||\mathbf{X}^{(v)} - \mathbf{P}_1^{(v)}\mathbf{P}_2^{(v)}...\mathbf{P}_L^{(v)}\mathbf{H}||_F^2$$
$$\text{s.t. } \mathbf{H} \succeq 0.$$

where the feature matrix $\mathbf{X}^{(v)}$ is factorized into hierarchical product of matrices $\mathbf{P}_1^{(v)}, \cdots, \mathbf{P}_L^{(v)}$ and latent representation $\mathbf{H}$.

2.3 Subspace Representation Learning

Our work is closely related to subspace clustering [23,30, 41]. Sparse Subspace Clustering(SSC) [23] aims to find a sparse representation matrix whose objective function is:

$$\min ||\mathbf{Z}||_1 \quad s.t. \quad \mathbf{X} = \mathbf{X}\mathbf{Z}, \; diag(\mathbf{Z}) = \mathbf{0}, \tag{1}$$

where $\mathbf{Z}$ is the subspace representation matrix and can be used for subsequent clustering or classification. Low-Rank Representation(LRR) [41] introduces the low rank regularization to subspace clustering by solving the following problem:

$$\min_{\mathbf{Z},\mathbf{E}} ||\mathbf{Z}||_* + \lambda||\mathbf{E}||_{2,1}, \quad s.t. \quad \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \tag{2}$$

where $\mathbf{E}$ corresponds to reconstruction error. Smooth Representation clustering(SMR) [30] underlines the importance of grouping effect to subspace clustering and the corresponding model is:

$$\min_{\mathbf{Z}} \alpha||\mathbf{X} - \mathbf{X}\mathbf{Z}||_F^2 + tr(\mathbf{Z}\widetilde{\mathbf{L}}\mathbf{Z}), \tag{3}$$

where $\widetilde{\mathbf{L}}$ is the graph Laplacian matrix. Although impressive performance has been achieved with these existing methods [23,30,41], they are only applicable for the data with single-view features. Recently, multi-view subspace clustering has achieved impressive performance. Specifically, the methods in [28, 61, 70] formulate multi-view learning as learning a common subspace representation. The dimensionality reduction based methods [6, 13] usually learn a low-dimensional subspace to integrate these multiple views and then obtain the final clustering result by using traditional clustering algorithm. Recently, some multi-view subspace clustering methods are proposed [9, 15, 25, 65, 70] based on the self-representation subspace clustering. Different from these methods learning a common representation [70] or exploring correlations of pairwise views [9, 25], we conduct multi-view subspace clustering with low-rank tensor to explore the high-order correlations across multiple views, and incorporate prior information as well.

2.4 Semi-Supervised Clustering

Generally, clustering is most related to representation learning, and here we mainly review the constrained clustering which could be roughly classified into two groups. The first category [62] usually learns a Mahalanobis distance to minimizes the distance between samples within the same class and maximizes the distance between samples of different classes. However, it may lead to over fitting since constraints are usually scarce. The second category extends the traditional clustering methods, *e.g.*, k-means [4, 57] or Gaussian mixtures [45] to constrained setting. There are also some methods [11, 33] simply replace entries of affinity matrix for must-link pairs with 1 and cannot-link pairs with 0.

For example, Video Face Clustering via Constrained Spares Representation (CS-VFC) [76] utilizes must-link and cannot-link constraints in two steps, *i.e.*, sparse representation and spectral clustering. Its objective function is as follows:

$$\min ||\mathbf{Z}||_1 \; s.t. \; \mathbf{X} = \mathbf{X}\mathbf{Z}, Z_{ji} = 0, (j, i) \in (\mathcal{M} \cup \mathcal{C} \cup \mathcal{I}) \tag{4}$$

where $\mathcal{M}, \mathcal{C}, \mathcal{I}$ are defined as the sets of the must-link, cannot-link constraints and indices corresponding to the elements with value 1 in the identity matrix, respectively. The way of constructing affinity matrix is provided as:

$$\mathbf{W}^{const} = |\mathbf{Z}| + |\mathbf{Z}|^T + \lambda\mathbf{M} + \beta\mathbf{C}, \tag{5}$$

where $\mathbf{M} \in \mathbb{R}^{N\times N}, \mathbf{C} \in \mathbb{R}^{N\times N}$ are the must-link matrix and cannot-link matrix, respectively. Here λ and β are trade-off parameters. Note that, we perform normalization for $\mathbf{Z}$ as $\mathbf{z_i} \leftarrow \mathbf{z_i}/||\mathbf{z_i}||_\infty$ so as to lead the values in affinity matrix to be of the same scale.

3 The Proposed Approach

3.1 Preliminary

In subspace clustering, the subspace representation is usually obtained in self-represented way. The affinity matrix is constructed according to the learned subspace representation. Given the data matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]$ with each column being a D-dimensional sample, where N is the number of samples. To obtain the subspace representation, representative subspace representation learning methods [23, 30, 41] usually share the following formulation

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{E}} \mathcal{L}(\mathbf{X}, \mathbf{X}\mathbf{Z}) + \lambda\Omega(\mathbf{Z}) \\ &s.t. \; \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \end{aligned} \tag{6}$$

where $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N] \in \mathbb{R}^{N\times N}$ is the reconstruction coefficient matrix, whose column $\mathbf{z}_i$ is the learned subspace representation vector corresponding to the sample $\mathbf{x}_i$, and $\mathbf{E} \in \mathbb{R}^{D\times N}$ is the reconstruction error matrix. $\mathcal{L}(\cdot, \cdot)$ denotes the loss function measuring reconstruction error, and $\Omega(\cdot)$ is the regularization term and λ is the trade-off parameter that balances the intensity of the loss and regularization. For clustering task, after obtaining subspace representation matrix $\mathbf{Z}$, an affinity matrix is constructed as $(|\mathbf{Z}| + |\mathbf{Z}^T|)/2$, where $|\cdot|$ denotes the absolute operator, and the spectral clustering algorithm [46] is applied on the affinity matrix for the final clustering result.

However, Eq. (6) could only handle single-view data. To extend the single-view subspace representation learning to multi-view setting, we can rewrite Eq. (6) as:

$$\begin{aligned}&\min_{\mathbf{Z}^{(v)},\mathbf{E}^{(v)}} \sum_{v=1}^{V} \left(\Omega(\mathbf{Z}^{(v)}) + \lambda_v \mathcal{L}(\mathbf{X}^{(v)}, \mathbf{X}^{(v)}\mathbf{Z}^{(v)})\right)\\ &s.t.\ \mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \mathbf{E}^{(v)},\ v = 1, 2, ..., V,\end{aligned} \tag{7}$$

where $\mathbf{X}^{(v)}$, $\mathbf{Z}^{(v)}$, $\mathbf{E}^{(v)}$ denote the data matrix in the v^{th} view, corresponding subspace representation matrix and reconstruction error matrix, respectively. Here λ_v denotes the hyperparameter for the v^{th} view and V is the number of views. Apparently, this naive way deals with each view data independently, which ignores the correlations among different views. Thus, our proposed algorithm aims to capture the high-order correlation among multiple views.

3.2 Multi-View Subspace Representation Learning with Low-rank Tensor

In our work, we propose a multiview subspace clustering method with a low-rank tensor constraint, which aims to learn the subspace representations of distinct views jointly and explore the high-order correlation underlying multiple views. The proposed method regards the subspace representation matrices of all view as a tensor, which is the generalization of matrix concept. The definition of tensor nuclear norm [41, 43, 55, 71, 72] is as follows:

$$\begin{aligned}&||\boldsymbol{\mathcal{Z}}||_* = \sum_{m=1}^{M} \xi_m ||\mathbf{Z}_{(m)}||_*\\ &s.t.\ \ \xi_i > 0,\ \sum_{m=1}^{M} \xi_m = 1,\end{aligned} \tag{8}$$

where ξ_ms are constants and M is the number of modes. The terminology of M-order tensor (or M-mode tensor) is defined as $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_M}$ and the unfold operation along the m^{th} mode on the tensor $\boldsymbol{\mathcal{Z}}$ transforms it into a matrix $\mathbf{Z}_{(m)}$ defined as $\text{unfold}_m(\boldsymbol{\mathcal{Z}}) = \mathbf{Z}_{(m)} \in \mathbb{R}^{I_m \times (I_1 \times \ldots \times I_{m-1} \times I_{m+1} \ldots \times I_M)}$ [20, 37]. The nuclear norm $|| \cdot ||_*$ can well approximate the rank of a matrix, since it is the tightest convex envelop for the rank of a matrix. Essentially the nuclear norm of a tensor is a convex combination of the nuclear norms of all matrices unfolded along each mode. We uses the nuclear norm to enforce the tensor $\boldsymbol{\mathcal{Z}}$ with a low-rank constraint as

$$\begin{aligned}&\min_{\mathbf{Z}^{(v)},\mathbf{E}^{(v)}} ||\mathbf{E}||_{2,1} + \lambda||\boldsymbol{\mathcal{Z}}||_*,\\ &s.t.\ \mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{Z}^{(v)} + \mathbf{E}^{(v)},\ v = 1, 2, ..., V,\\ &\quad\ \boldsymbol{\mathcal{Z}} = \Psi(\mathbf{Z}^{(1)}, ..., \mathbf{Z}^{(V)}),\ \mathbf{E} = [\mathbf{E}^{(1)}; ...; \mathbf{E}^{(V)}],\end{aligned} \tag{9}$$

where $\Psi(\cdot)$ combines the representations of distinct views $\mathbf{Z}^{(v)}$ into a 3-order tensor $\boldsymbol{\mathcal{Z}}$, whose dimensionality is $N \times N \times V$. We concatenate together along the columns of errors with respect to each view in the vertical direction, forming as $\mathbf{E} = [\mathbf{E}^{(1)}; \mathbf{E}^{(2)}; ...; \mathbf{E}^{(V)}]$ and apply $\ell_{2,1}$-norm $||.||_{2,1}$ to encourage $\mathbf{E}$ to be sparse in columns. There is an underlying assumption that corruptions are sample-specific, which means some instances are corrupted. In the manner of integration, the columns of $[\mathbf{E}^{(1)}, \mathbf{E}^{(2)}, ..., \mathbf{E}^{(V)}]$ will be constrained with jointly consistent magnitude values [14]. Note that, to decrease the variation in the magnitude of the error corresponding to different views, we normalize the data matrices in each view to impose the same scale on the error of distinct views. Specifically, we normalize $\mathbf{x}_i$ with $\mathbf{x}_i \leftarrow \mathbf{x}_i/||\mathbf{x}_i||_2$.

3.3 Constrained Multi-View Subspace Representation Learning

Although the subspace representation learning could be improved with multiple views, it is still challenging because there is no label information guiding the learning process. Fortunately, there is usually prior knowledge (*e.g.*, must-link constraint) available which injects discriminative information into the representation learning. The must-link prior information indicates whether samples belong to the same cluster.

To incorporate must-link constraints into the proposed multi-view representation learning model, a constraint matrix [32] is constructed. Suppose there are L samples belonging to C sets where the samples in each set belong to the same class. The rest $N - L$ samples with no constraints are considered to belong to $N - L$ sets. Then, the dataset is partitioned into $N-L+C$ sets, and accordingly, we can construct a constraint matrix $\mathbf{Q} \in \mathbb{R}^{N \times (N-L+C)}$, where $\mathbf{Q}_{i,j} = 1$ if $\mathbf{x}_i$ is in the j^{th} set. To ensure samples in the same set to be clustered into the same cluster, an auxiliary matrix $\mathbf{U}^{(v)} \in \mathbb{R}^{N \times (N-L+C)}$ is designed for each view satisfying $\mathbf{Z}^{(v)} = \mathbf{U}^{(v)}\mathbf{Q}^T$. The objective function in Eq. (9) can be reformulated as

$$\begin{aligned}&\min_{\mathbf{Z}^{(v)},\mathbf{E}^{(v)}} ||\mathbf{E}||_{2,1} + \lambda||\boldsymbol{\mathcal{Z}}||_*\\ &s.t.\ \mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{U}^{(v)}\mathbf{Q}^T + \mathbf{E}^{(v)},\\ &\quad\ \mathbf{E} = [\mathbf{E}^{(1)}; ...; \mathbf{E}^{(V)}],\\ &\quad\ \boldsymbol{\mathcal{Z}} = \Psi(\mathbf{Z}^{(1)}, ..., \mathbf{Z}^{(V)}),\ \mathbf{Z}^{(v)} = \mathbf{U}^{(v)}\mathbf{Q}^T.\end{aligned} \tag{10}$$

Proposition 1 *Under the equation* $\mathbf{Z}^{(v)} = \mathbf{U}^{(v)}\mathbf{Q}^T$*, we have* $||\boldsymbol{\mathcal{Z}}||_* \leq ||\boldsymbol{\mathcal{U}}||_*$*.*

Based on Proposition 1, $||\boldsymbol{\mathcal{U}}||_*$ is an upper bound of $||\boldsymbol{\mathcal{Z}}||_*$. Therefore, for our objective function Eq. (10), we substitute $||\boldsymbol{\mathcal{Z}}||_*$ by $||\boldsymbol{\mathcal{U}}||_*$ to avoid the inverse operation. Accordingly, the optimization problem of Eq. (10) is transformed as

$$\begin{aligned} &\min_{\mathbf{U}^{(v)},\mathbf{E}^{(v)}} ||\mathbf{E}||_{2,1} + \lambda||\boldsymbol{\mathcal{U}}||_* \\ &s.t.\ \mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{U}^{(v)}\mathbf{Q}^T + \mathbf{E}^{(v)}, \\ &\quad \boldsymbol{\mathcal{U}} = \Psi(\mathbf{U}^{(1)}, ..., \mathbf{U}^{(V)}),\ \mathbf{E} = [\mathbf{E}^{(1)}; ...; \mathbf{E}^{(V)}]. \end{aligned} \tag{11}$$

According to the Proposition 1, minimizing $||\mathcal{U}||_*$ can be considered as an approximation of minimizing $||\mathcal{Z}||_*$. This way of approximation is widely used in the field of optimization. Specifically, because $||\mathcal{U}||_*$ is an upper bound of $||\mathcal{Z}||_*$, any constraint $||\mathcal{Z}||_* < a$ can be satisfied by enforcing $||\mathcal{U}||_* < b$ ($b \leq a$ is a sufficient condition). Therefore, in practice use, we can satisfy the strength of the low-rank property for $\mathcal{Z}$ by setting an appropriate value for the hyper-parameter λ.

Model properties. *To summarize, we highlight that the proposed latent partial multi-view representation enjoys the following merits: (1) Our model explores the high-order correlations by simultaneously mining the intra-view and inter-view correlations, which is especially important for the multi-view data. (2) The supervised information is incorporated into the proposed multi-view subspace representation learning model, which could guide the learning process for more accurate result. (3) The proposed algorithm is a flexible framework, where the constraint matrix is constructed automatically according to supervised information and the model will be reduced into unconstrained one if there is no prior information.*

3.4 Optimization

The Augmented Lagrange Multiplier (ALM) is an efficient algorithm for solving optimization problems under equation constraints. The ALM with alternating direction minimizing strategy is an efficient solver for our problem (10). It is necessary to make our objective function separable to adopt this strategy. Thus, we follow [55] to introduce an auxiliary tensor $\boldsymbol{\mathcal{G}}$ consisting of V variables $\mathbf{G}^{(v)}$'s to replace $\boldsymbol{\mathcal{U}}$, and convert it to the following optimization problem as

$$\begin{aligned} &\min_{\mathbf{U}^{(v)},\mathbf{E}^{(v)},\mathbf{G}_m} ||\mathbf{E}||_{2,1} + \lambda||\boldsymbol{\mathcal{G}}||_* \\ &s.t.\ \boldsymbol{\mathcal{U}} = \boldsymbol{\mathcal{G}},\ \mathbf{X}^{(v)} = \mathbf{X}^{(v)}\mathbf{U}^{(v)}\mathbf{Q}^T + \mathbf{E}^{(v)} \\ &\quad \boldsymbol{\mathcal{U}} = \Psi(\mathbf{U}^{(1)}, ..., \mathbf{U}^{(V)}),\ \boldsymbol{\mathcal{G}} = \Psi(\mathbf{G}^{(1)}, ..., \mathbf{G}^{(V)}), \\ &\quad \mathbf{E} = [\mathbf{E}^{(1)}; ...; \mathbf{E}^{(V)}], \end{aligned} \tag{12}$$

where $\boldsymbol{\mathcal{G}}$ is the augmented variable corresponding to $\boldsymbol{\mathcal{U}}$ that makes our problem separable. The first constraint ensures the equivalence between (11) and (12). The second constraint jointly relates the data points of the same cluster, *i.e.*, the same linear subspace, and takes the prior into consideration. The last constraint with $\ell_{2,1}$-norm gives the underlying assumption of error, *i.e.*, sample-specific error. The optimization problem of Eq. (12) can be solved by the AL-ADM method [39], which minimizes the following augmented Lagrangian function:

$$\begin{aligned} &\mathcal{L}_{\mu>0}(\{\mathbf{U}^{(v)}; \mathbf{E}^{(v)}\}_{v=1}^V; \{\mathbf{G}_{(m)}\}_{m=1}^M) = \\ &\quad ||\mathbf{E}||_{2,1} + \sum_{m=1}^M \lambda_m||\mathbf{G}_{(m)}||_* + \Phi(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{U}} - \boldsymbol{\mathcal{G}}) \\ &\quad + \sum_{v=1}^V \Phi(\mathbf{Y}_v^T, \mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{U}^{(v)}\mathbf{Q}^T - \mathbf{E}^{(v)}), \end{aligned} \tag{13}$$

where $\lambda_m = \lambda\xi_m > 0$ encodes the intensity of the low-rank tensor constraint and $\mathbf{G}_{(m)}$ is the m^{th} mode unfolding matrix of $\boldsymbol{\mathcal{G}}$. For convenience, we give the definition $\Phi(\mathbf{Y}, \mathbf{C}) = \frac{\mu}{2}||\mathbf{C}||_F^2 + \langle \mathbf{Y}, \mathbf{C} \rangle$, where $\langle \cdot, \cdot \rangle$ denotes matrix inner product and μ is a positive penalty scalar. The above unconstrained problem can be solved by alternating minimization method corresponding to the variables $\mathbf{E}^{(v)}$, $\mathbf{U}^{(v)}$ and $\mathbf{G}_{(m)}$ and then updating the Lagrange multipliers $\mathbf{Y}_v$ and $\mathbf{W}^{(v)}$ accordingly. In this paper, the AL-ADM strategy is adopted and outlined in Algorithm 1 to optimize our problem by updating each variable for each iteration. The optimization for each subproblem is as follows:

1. $\mathbf{U}^{(v)}$-subproblem: For updating $\mathbf{U}^{(v)}$, we solve the following problem by fixing the other variables:

$$\begin{aligned} \mathbf{U}^{(v)^*} = &\operatorname*{argmin}_{\mathbf{U}^{(v)}} \Phi(\mathbf{W}^{(v)}, \mathbf{U}^{(v)} - \mathbf{G}^{(v)}) \\ &+ \Phi(\mathbf{Y}_v^T, \mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{U}^{(v)}\mathbf{Q}^T - \mathbf{E}^{(v)}). \end{aligned} \tag{14}$$

Taking the derivative with respect to $\mathbf{U}^{(v)}$ and set it to zero, we obtain the following equation:

$$\begin{aligned} &\mathbf{A}\mathbf{U}^{(v)} + \mathbf{U}^{(v)}\mathbf{B} = \mathbf{C} \\ &\text{with } \mathbf{A} = (\mathbf{X}^{(v)^T}\mathbf{X}^{(v)})^{-1},\ \mathbf{B} = \mathbf{Q}^T\mathbf{Q} \\ &\mathbf{C} = (\mathbf{X}^{(v)^T}\mathbf{X}^{(v)})^{-1}\Big(\mathbf{G}^{(v)} - \mathbf{W}^{(v)}/\mu \\ &\quad + \mathbf{X}^{(v)^T}\mathbf{Y}_v\mathbf{Q}/\mu + \mathbf{X}^{(v)^T}\mathbf{X}^{(v)}\mathbf{Q} - \mathbf{X}^{(v)^T}\mathbf{E}^{(v)}\mathbf{Q}\Big). \end{aligned} \tag{15}$$

The above equation is Sylvester equation [3]. We can find a unique solution to solve the problem in (15). The

Algorithm 1: Algorithm of TMSRL

```
Input: Multiple types of feature matrices: $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$,
       ..., $\mathbf{X}^{(V)}$, prior knowledge matrix $\mathbf{Q}$,
       parameters $\lambda_m$'s and the number of clusters $K$
Initialize: $\mathbf{U}^{(1)} = \mathbf{0}, ..., \mathbf{U}^{(V)} = \mathbf{0}$;
$\mathbf{Z}^{(1)} = \mathbf{0}, ..., \mathbf{Z}^{(V)} = \mathbf{0}$; $\mathbf{E}^{(1)} = \mathbf{0}, ..., \mathbf{E}^{(V)} = \mathbf{0}$;
$\mathbf{Y}_1 = \mathbf{0}, ..., \mathbf{Y}_V = \mathbf{0}$; $\mathbf{W}^{(1)} = \mathbf{0}, ..., \mathbf{W}^{(V)} = \mathbf{0}$;
$\mathbf{G}^{(1)} = \mathbf{0}, ..., \mathbf{G}^{(V)} = \mathbf{0}$; $\mu = 10^{-5}$; $\rho = 1.5$; $\varepsilon = 10^{-5}$;
$\max_\mu = 10^{10}$
while not converged do
    for each of V views do
        Update $\mathbf{U}^{(v)}$, $\mathbf{E}^{(v)}$ and $\mathbf{Y}_v$ according to Eq.
        (15), (16) and (17), respectively;
        Compute subspace representation of each view
        by $\mathbf{Z}^{(v)} = \mathbf{U}^{(v)}\mathbf{Q}^T$;
    end
    for each of M modes do
        Update $\mathbf{G}_{(m)}$, $\boldsymbol{\mathcal{W}}$ according to Eq. (18) and
        (19), respectively;
    end
    Update the parameter $\mu$ by $\mu = \min(\rho\mu; \max_\mu)$;
    check the convergence conditions:
    $||\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{U}^{(v)}\mathbf{Q}^T - \mathbf{E}^{(v)}||_\infty < \epsilon$ and
    $||\mathbf{U}^{(v)} - \mathbf{G}^{(v)}||_\infty < \epsilon$;
end
Combine all subspace representations of each view by
$\mathbf{S} = \frac{1}{V}\sum_{v=1}^{V} |\mathbf{Z}^{(v)}| + |\mathbf{Z}^{(v)T}|$;
Apply the spectral clustering/classification algorithm
with $\mathbf{S}$;
Output: Clustering/classification result.
```

classical algorithm for solving the Sylvester equation is the Bartels-Stewart algorithm [3].

2. E-subproblem: The reconstruction error matrix $\mathbf{E}$ is optimized by:

$$\begin{aligned}\mathbf{E}^* = &\operatorname*{argmin}_{\mathbf{E}} ||\mathbf{E}||_{2,1} \\ &+ \sum_{k=1}^{V} \Phi(\mathbf{Y}_v^T, \mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{U}^{(v)}\mathbf{Q}^T - \mathbf{E}^{(v)}) \\ = &\operatorname*{argmin}_{\mathbf{E}} \frac{1}{\mu}||\mathbf{E}||_{2,1} + \frac{1}{2}||\mathbf{E} - \mathbf{F}||_F^2,\end{aligned} \tag{16}$$

where $\mathbf{F}$ is formed by vertically concatenating the matrices $\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{U}^{(v)}\mathbf{Q}^T + \mathbf{Y}^{(v)}/\mu$ together along column. This subproblem can be efficiently solved by Lemma 3.2 in [41].

3. $\mathbf{Y}_v$-subproblem: The multiplier $\mathbf{Y}_v$ is updated by:

$$\mathbf{Y}_v^* = \mathbf{Y}_v + (\mathbf{X}^{(v)} - \mathbf{X}^{(v)}\mathbf{U}^{(v)}\mathbf{Q}^T - \mathbf{E}^{(v)}). \tag{17}$$

Intuitively, the multiplier is updated proportionally to the violation of the equality constraint.

4. $\boldsymbol{\mathcal{G}}$-subproblem: $\mathbf{G}_{(m)}$ is updated by:

$$\begin{aligned}\mathbf{G}_{(m)}^* &= \operatorname*{argmin}_{\mathbf{G}_{(m)}} \lambda_m ||\mathbf{G}_{(m)}||_* + \Phi(\mathbf{W}_{(m)}, \mathbf{U}_{(m)} - \mathbf{G}_{(m)}) \\ &= \frac{\lambda_m}{\mu}||\mathbf{G}_{(m)}||_* + \frac{1}{2}||\mathbf{G}_{(m)} - (\mathbf{U}_{(m)} + \mathbf{W}_{(m)}/\mu)||_F^2.\end{aligned} \tag{18}$$

Specifically, there are three unfolding ways for a three-mode tensor in our model. $\mathbf{G}_{(m)}$ is a matrix coresponding to the mth mode unfolding of $\boldsymbol{\mathcal{G}}$. We can update $\mathbf{G}_{(m)}$ as a matrix by the singular value thresholding operator [7].

5. $\boldsymbol{\mathcal{W}}$-subproblem: Similarly to updating $\mathbf{Y}_v$, the variable $\boldsymbol{\mathcal{W}}$ is updated by:

$$\boldsymbol{\mathcal{W}}^* = \boldsymbol{\mathcal{W}} + \mu(\boldsymbol{\mathcal{U}} - \boldsymbol{\mathcal{G}}). \tag{19}$$

Compared to the penalty method, taking $\mu \to \infty$ is not necessary for the ALM method to solve the original constrained problem. In contrast, owing to the Lagrangian multiplier term, our method has a fast convergence speed since μ can be kept much smaller. Actually any optimization algorithm can be utilized to solve our problem in Eq. (12) and we just provide a general optimization scheme. For example, for large scale problem, LADMPSAP [40] can substitute AL-ADM to achieve a more efficient performance. Furthermore, some methods can also be employed to improve the matrix inversion computation (*e.g.*, [51, 53]) and Singular Value Thresholding (SVT) operators (*e.g.*, [29, 48]).

3.5 Complexity and Convergence

The detail of our method is summarized on Algorithm 1. The optimization process of our models mainly consists of five sub-problems. Firstly, solving the $\mathbf{U}^{(v)}$-subproblem involves matrix inversion and the Sylvester equation, both of which are with the complexity of $O(n^3)$. The complexity of updating $\mathbf{U}^{(v)}$ is $O(dn^2 + n^3)$, where d and n are the dimension of single-view feature and the number of data, respectively. The computations of updating $\mathbf{E}^{(v)}$ and Y_v are matrix multiplication with the complexity of $O(dn^2)$. The complexity of $\mathbf{G}_{(m)}$-subproblem is also $O(n^3)$, since it is with the nuclear norm proximal operator. Overall, the total complexity of our algorithm is $O(dn^2 + n^3)$ for each iteration.

Generally, it is difficult to prove the convergence of our proposed algorithm in theory but convergence properties could be analyzed similarly as those in [39]. For $\mathbf{U}^{(v)}$-subproblem, we can find a unique solution [3]. Lemma 3.2 in [41] gives the optimal solution of $\mathbf{E}^{(v)}$-subproblem and the convergence of $\mathbf{G}_{(m)}$-subproblem is guaranteed in the work [7]. For each subproblem, the convergence is ensured well. Moreover, the empirical evidence on real data suggests that our algorithm has a stable convergence behavior.

4 Experiments

4.1 Experiments Datasets

Fig. 2 presents some example images of four benchmark datasets used in our experiments. These datasets are widely used to perform face and image clustering tasks in recent works [23, 30, 41]. The detailed information of these datasets is described as follows:

•**Yale** [1]**.** The Yale face dataset contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration.

•**Extended YaleB** [2]**.** The Extended YaleB dataset consists of 38 individuals and around 64 near frontal images under different illuminations for each individual. Similarly to the other work [41], we use the images for the first 10 classes, including 640 frontal face images.

•**ORL** [3]**.** There are 10 different images of each of 40 distinct subjects in the ORL face dataset. They took the images at different times, changing the lighting, facial expressions and facial details for some subjects.

•**COIL-20** [4]**.** The Columbia Object Image Library (COIL-20) dataset contains 1440 images of 20 object categories. Each category contains 72 images. All the images are normalized to 32 × 32 pixel arrays with 256 gray levels per pixel.

•**BBCSport** [5]**.** The dataset consists of documents of sports news corresponding to 5 topics, where two different types of features are extracted [63].

•**Football** [6]**.** The dataset is a collection of 248 English Premier League football players and clubs active on Twitter. The disjoint ground truth communities correspond to the 20 clubs in the league.

•**Politicsie** [7]**.** The dataset is a collection of Irish politicians and political organizations assigned to seven disjoint groups according to their affiliation. The two Twitter datasets are associated with 9 different views.

In our experiments, we extract three types of features (i.e., intensity, Local Binary Pattern (LBP) [49] and Gabor [36]) for the image datasets (i.e., Yale, Extended YaleB, ORL, and COIL-20). The intensity feature is the intensity of a single-channel image pixel, *e.g.*, image grayscale. The extracted standard LBP features are with the sampling density size of 8 and the blocking number of 7×8. We extract the Gabor wavelets at four orientations $\theta = \{0^o, 45^o, 90^o, 135^o\}$ with one scale $\lambda = 4$. Accordingly, the dimensionality of intensity feature depends on the size of image and the numbers of dimensions for LBP and Gabor are 3304 and 6750, respectively. For the BBCSport dataset, each document is divided into two segments. And then, standard stemming, stop-word removal and TF-IDF normalization procedures are applied to two segments separately to produce two different views [26]. For the Football and Politicsie datasets from Twitter, the social relationships (networks): 'follows', 'followed by', 'mentions', 'mentioned by', 'retweets' and 'retweeted by' between two users are utilized as six views. Each user belongs to a specific user list with detailed description. Additionally, two views are constructed by the belongs with two kinds of features of user lists, i.e., user-list names and key words of user-list names with textual descriptions. Moreover, the tweet profile vector is constructed with a certain number of each user's most recent tweets to generate the last view. The statistics of the used datasets are shown in Table 1.

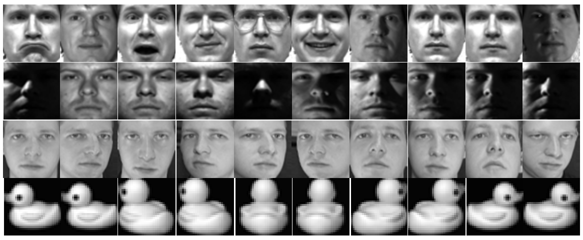


Figure 2: Example images of the four datasets used in this paper (the rows from top to bottom correspond to Yale, Extended YaleB, ORL and COIL-20, respectively).

For fair comparison, we do not use the must-link information for all the compared methods and ours in the unsupervised experiment. Specifically, the subspace representation matrices are learned according to Eq. 9, and then they are combined into an affinity graph. Moreover, in the constrained multi-view representation learning experiment, we introduce the must-link information for all compared methods.

Table 1: Statistics of the used datasets.

dataset	#instance	#view	#class	domain
Yale	165	3	15	image
Extended YaleB	640	3	10	image
ORL	400	3	40	image
COIL-20	1440	3	20	image
BBCSport	737	2	5	text
Football	248	9	5	social media
Politicsie	348	9	7	social media

[1] http://cvc.yale.edu/projects/yalefaces/yalefaces.html
[2] http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html
[3] http://www.uk.research.att.com/facedatabase.html
[4] http://www.cs.columbia.edu/CAVE/software/softlib/
[5] http://mlg.ucd.ie/datasets/
[6] http://mlg.ucd.ie/aggregation/
[7] http://mlg.ucd.ie/aggregation/

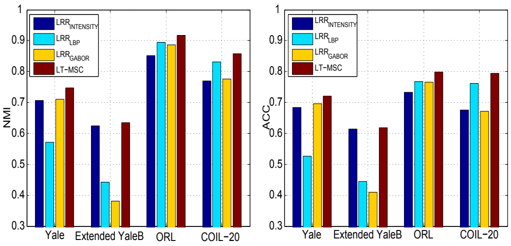


Figure 3: Comparison between LRR with features of each single view and our TMSRL with multiple views.

4.2 Experiments of Unsupervised Multi-View Representation Learning

We first compare our method with other multi-view clustering methods since the subspace representation is usually used for clustering task. For comprehensive evaluation, there are 10 compared methods in our experiments, including 3 single-view and 7 multiview ones. Specifically, these methods are as follows:

• $\mathbf{SPC_{best}}$. This is the standard spectral clustering algorithm [46] employing the most informative view.

•$\mathbf{LRR_{best}}$ [41]. This is the low-rank constraint subspace clustering algorithm with the best performed single view.

•**RTC** [8]. The method utilizes tensor to represent images and it is robust to the outliers.

•$\mathbf{FeatConcate_{PCA}}$. The method first concatenates together all views and then employs PCA to reduce the number of dimensions to 300.

•**PCA+LRR**. The method concatenates all views and employs PCA to reduce the feature dimension to 300, on which LRR is applied.

•**Co-Reg SPC** [35]. The method co-regularizes the clustering hypotheses to enforce corresponding samples to have the same cluster membership.

•**Co-Training SPC** [34]. The method uses the co-training manner within the spectral clustering framework.

•**Min-Disagreement** [21]. The idea of "minimizing-disagreement" is realized based on a bipartite graph.

•**ConvexReg SPC** [17]. The method learns a common representation for all views.

•**RMSC** [64]. The method seeks a cross-view shared low-rank transition probability matrix for clustering.

•**MSSC** [1]. The method exploits the complementarity by using a common representation across different modalities.

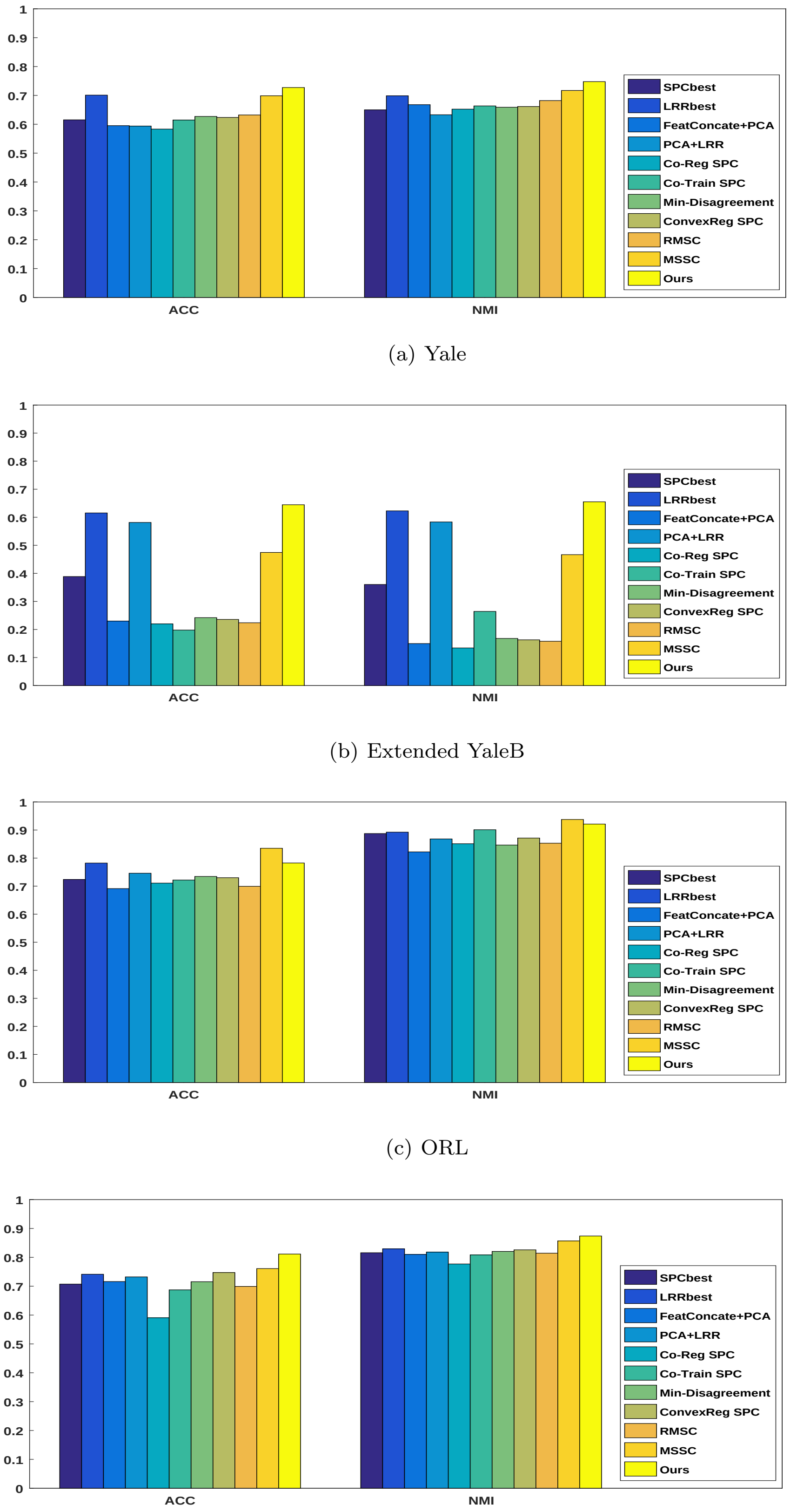
The above comparison methods are conducted by running 30 times and reporting the average performance and standard deviation. We utilize two commonly used metrics to evaluate the clustering quality: Normalized Mutual Information (NMI) and Accuracy (ACC), which have been widely used for perform a clustering evaluation [16, 38]. For instance, the compared methods, Co-Train SPC [34] and LRR [41], also utilize the same metrics for evaluating. Specifically, Co-Train SPC uses NMI and LRR uses accuracy (ACC) for evaluating clustering task. ACC and NMI favors different properties in the clustering, and a higher value indicates a better clustering performance for both of them.

In our experiments, we adopt the inner product kernel to compute the graph similarity. For the parameters of our approach on all the four datasets, we simply set the M parameters with equal value, *i.e.*, $\lambda_1 = .. = \lambda_M = \lambda$, and accordingly only one parameter λ needs to tune. For all the compared methods, we try our best to tune the parameters for the best performance.

In Fig. 3, we compare our model with LRR using each single view. It is observed that LRR using the best single view achieves promising performance, while the performances with different views vary significantly. For example, LBP is the best view on ORL and COIL-20, but there is a serious degeneration on Extended YaleB. Therefore, it is not reasonable to choose the same view for different datasets. On the contrary, our method directly uses all views and achieves competitive performance, while other multi-view clustering methods can not produce promising results. This demonstrates that our model can effectively integrate information from multiple views.

Fig. 4 shows the detailed clustering results of different methods in terms of accuracy and NMI. Our method basically outperforms all the baselines on four benchmark datasets. Specifically for Yale dataset, it is worth noting that RMSC, the most competitive multiview clustering model, obtains a relatively promising performance, but LRR even achieves a better result provided with the best feature. TMSRL outperforms LRR with approximately 3.6% and 5.6% in terms of ACC and NMI, respectively. In addition, as shown in Fig. 4, concatenating all views and reducing dimensionality with PCA ($\mathrm{FeatConcate_{PCA}}$) is not promising as expected since its performance is not always superior to the result of the best single view. Besides, our method also performs better than two state-of-the-art multi-view clustering methods [17, 64]. The experiments results on ORL and COIL-20 in the last two rows of Fig. 4 verify the effectiveness of our approach.

Note that, most comparison methods have relatively unpromising performances on Extended YaleB except the self-representation based subspace methods (*e.g.*, LRR), as shown in the second row of Fig. 4. This is mainly due to the large variation of illumination. For

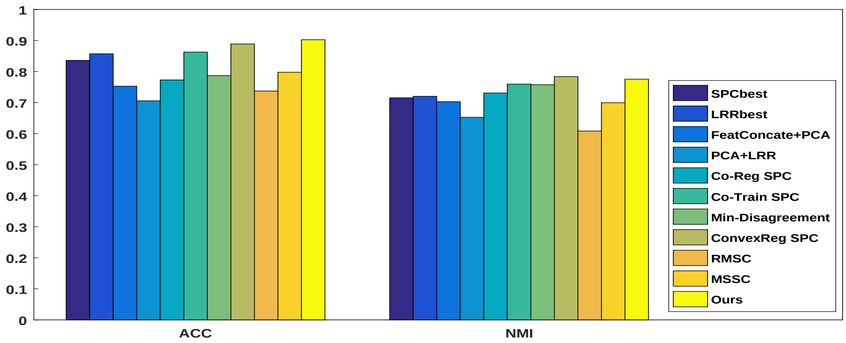


(a) Yale

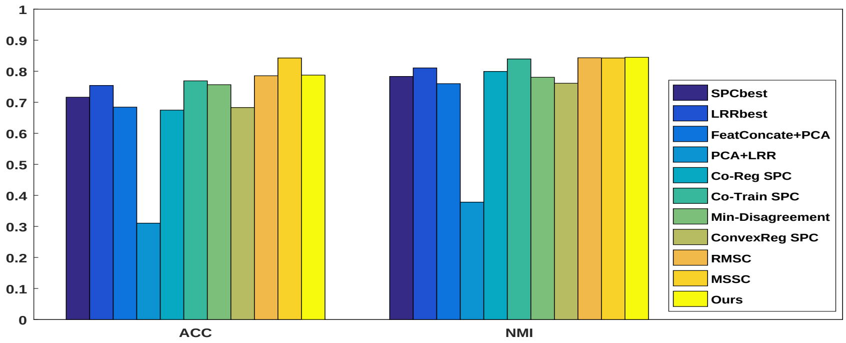
(b) Extended YaleB

(c) ORL

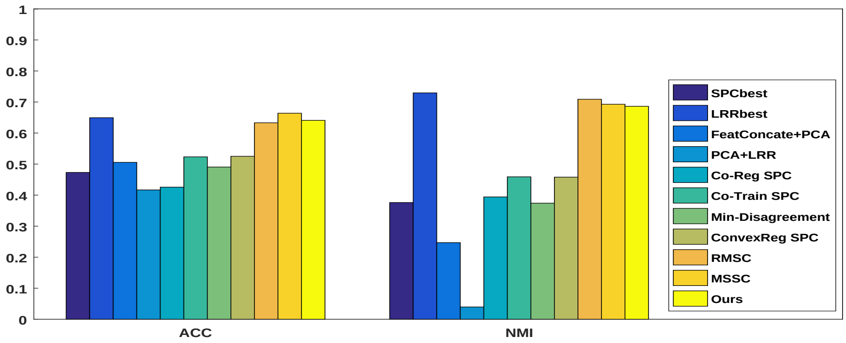
(d) COIL-20



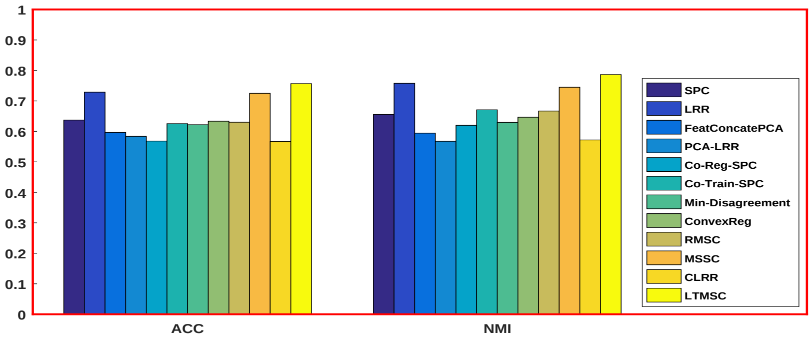
(e) bbcsport



(f) football

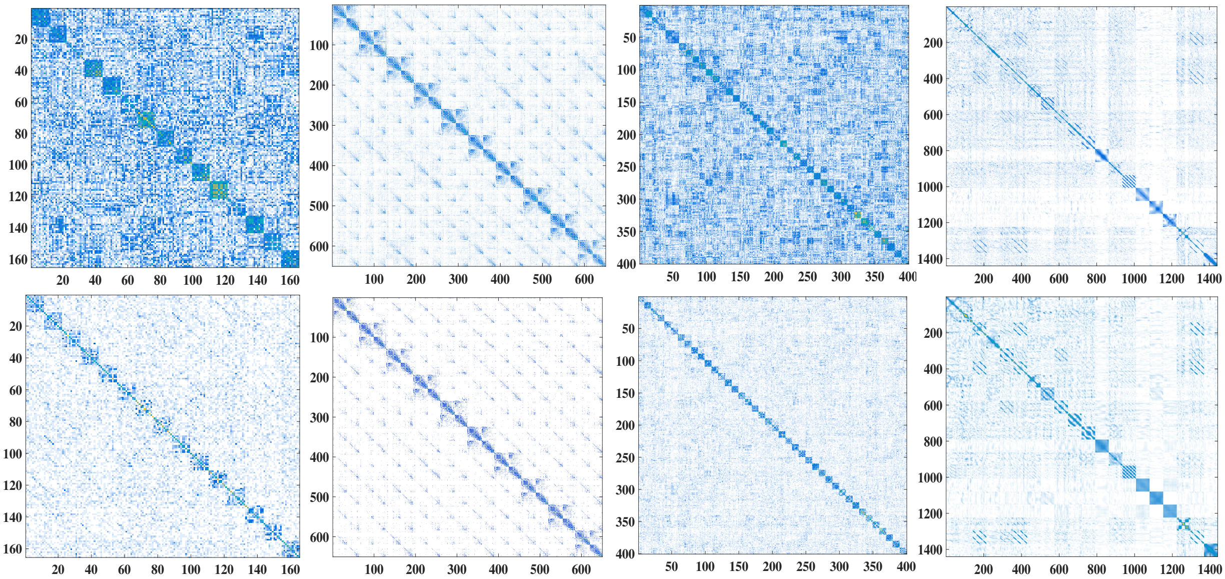


(g) politicsie



(h) averaged performance on all datasets

Figure 4: Results (mean ± standard deviation) in terms of accuracy and NMI.



(a) Yale (b) Extended YaleB (c) ORL (d) COIL-20

Figure 5: Affinity matrices of using naive manner with LRR (top row) as in Eq. 7 and TMSRL (bottom row).

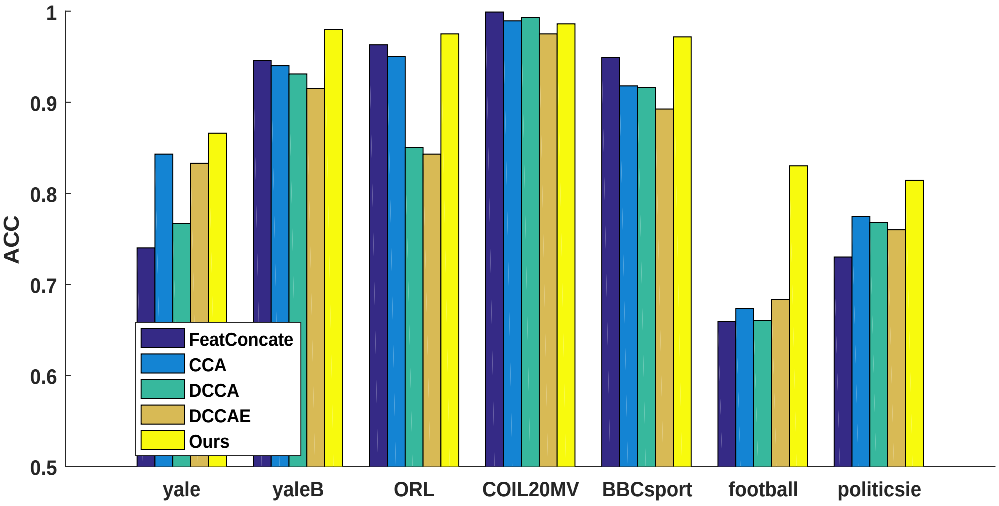
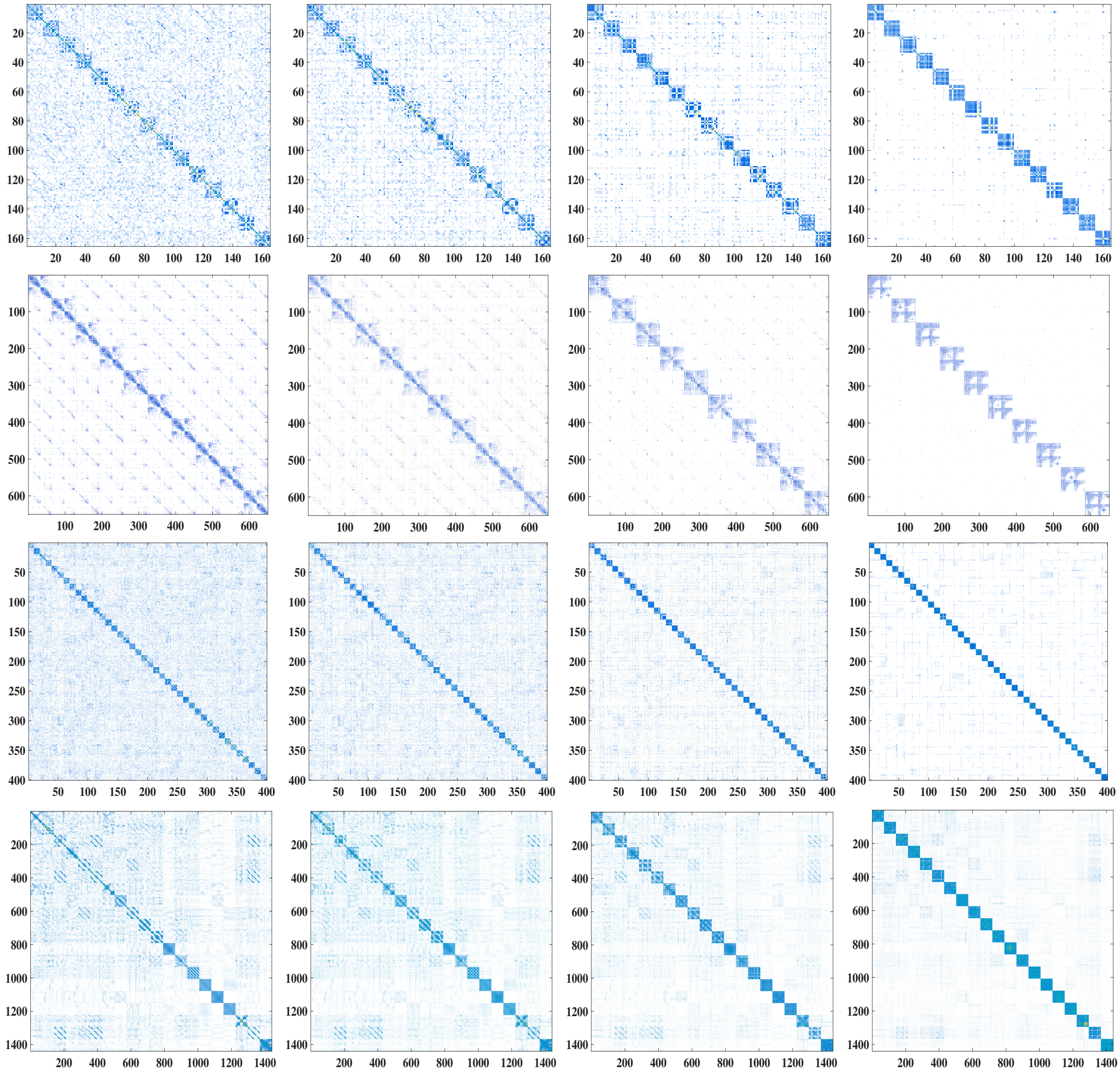


Figure 6: Classification comparison in terms of accuracy.

instance, benefiting from the self-representation manner, the subspace clustering methods are robust with respect to the intensity feature, while the traditional distance-based methods are dramatically degraded. We can see that LRR shows the best performance among the baselines (*e.g.*, Co-Training SPC, Min-Disagreement and ConvexReg SPC). The clustering results of our model are much better than *PCA+LRR* with the help of the high order low-rank tensor constraint. Besides, we find that our method gains such a significant improvement over LRR on Extend YaleB while it is not as much as that of other datasets. This is mainly because the LBP and Gabor features basically not as effective as the intensity features, which degrades the clustering results of ours.

Fig. 5 shows the visualizations for affinity matrices of our method and LRR which independently learns multiple affinity matrices and then adds them. According to the ground-truth clusters, we visualize these affinity matrices. Compared to LRR, our algorithm reveals the underlying clustering structures more clearly. The results of visualizations further verifies that our method can



sup_ratio = 0 **sup_ratio = 0.3** **sup_ratio = 0.6** **sup_ratio = 0.9**

Figure 7: Visualization of affinity matrices on Yale, Extended YaleB, ORL and COIL-20 under different ratios of constraints.

well explore the high-order correlation across multiple views.

We also compared our proposed methods with FeatConcate, CCA, Deep Canonical Correlation Analysis (DCCA) [2] and Deep Canonically Correlated AutoEncoders (DCCAE) [60] on classification task. As shown in Fig. 6, the performance of our proposed method is rather competitive, where ours performs as the best on three out of four datasets. It is observed that FeatConcate performs competitive when the quality of each view is promising. Specifically, according to Fig. 2 and Fig. 6, we can find that the performance with each single view of COL20MV is generally good, which indicates high quality of each view. The possible reason is that simple methods may also work well when each view is enough for promising performance. We can also find that FeatConcate performs rather unpromising on the relatively difficult datasets, *i.e.*, yale, football and politicsie.

4.3 Experiments for Constrained Multi-View Representation Learning

We compare our method with 5 semi-supervised clustering methods under different ratios of supervised infor-

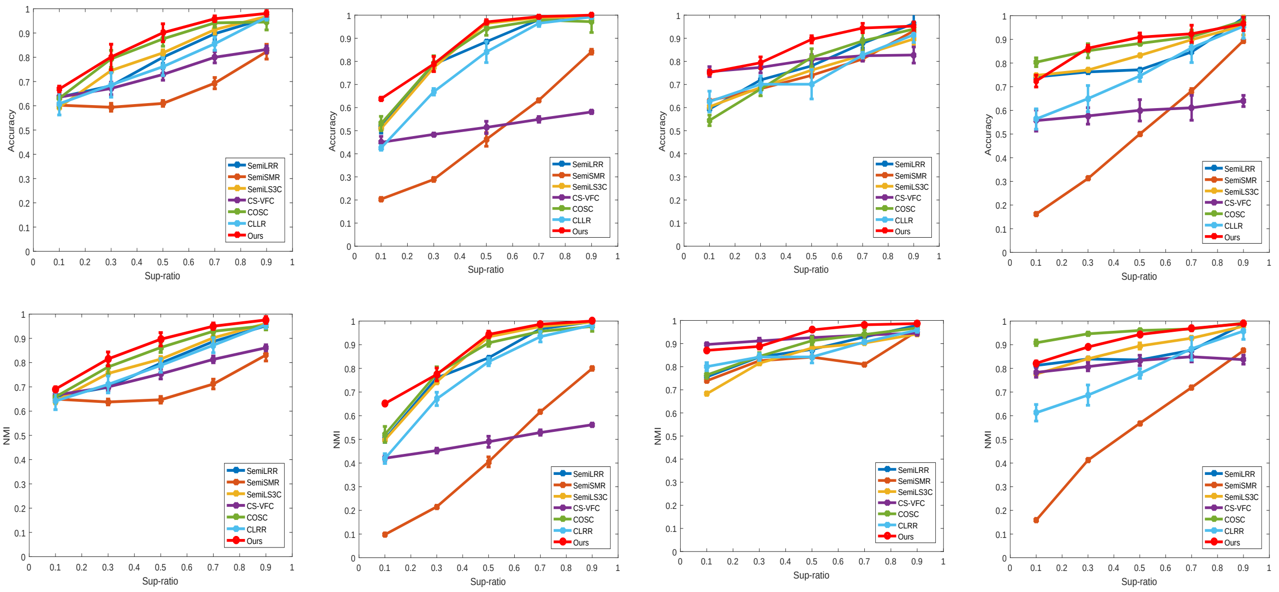
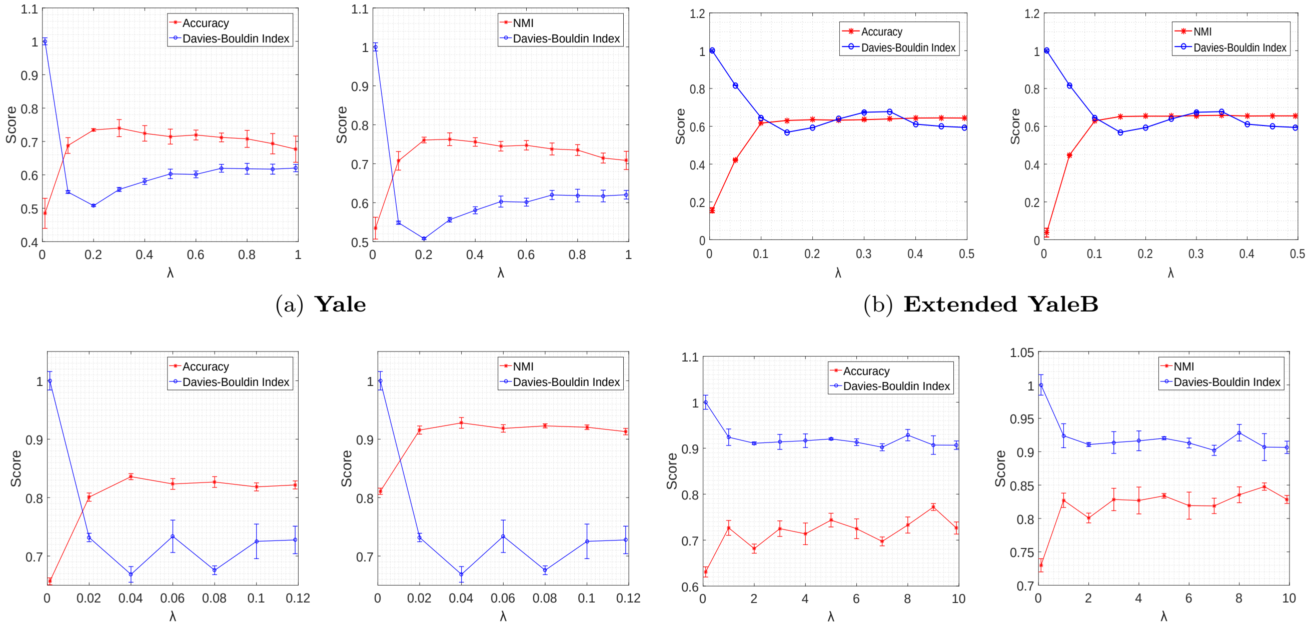


Figure 8: Clustering performance under different ratios of constraints on Yale, Extended YaleB, ORL and COIL-20.



(a) **Yale**

(b) **Extended YaleB**

(c) **ORL**

(d) **COIL20**

Figure 9: Parameter tuning in terms of internal (i.e., Davies-Bouldin Index) and external (ACC and NMI) metrics on four benchmark datasets. Since we set $\lambda_1 = ... = \lambda_M = \lambda$ in our experiment, we only should tune the parameter λ.

mation. We introduce the semi-supervised manner for 3 subspace clustering methods by introducing must-link constraint [42], which serves as the prior information to modify the affinity matrix by setting $S_{i,j} = 1$ if and only if x_i and x_j are of must-link. Specifically, these comparison approaches include 3 slightly modified subspace clustering methods and 2 constrained clustering methods:

- **SemiLRR** [41]. The method concatenates all views and employs PCA to reduce the feature dimension to 1000. SemiLRR modifies the learned affinity matrices by LRR with the must-link constraint.

•**SemiSMR** [31]. Smooth Representation clustering (SMR) introduces the enforced grouping effect conditions a representation based subspace clustering model. SemiSMR modifies the learned affinity matrices by SMR with the must-link constraint.

•**SemiLS3C** [50]. Latent Space Sparse Subspace clustering (LS3C) learns the projection of data and finds the sparse coefficients in the low-dimensional latent space. SemiLS3C modifies the learned affinity matrices by LS3C with the must-link constraint.

•**CS-VFC** [76]. Video Face Clustering via Constrained Sparse Representation (CS-VFC) utilizes the must-link and cannot-link constraints in the video face clustering task on the two stages of sparse representation and spectral clustering.

•**COSC** [52]. Constrained 1-Spectral Clustering (COSC) presents a generalization of the popular spectral clustering technique which integrates must-link and cannot-link constraints.

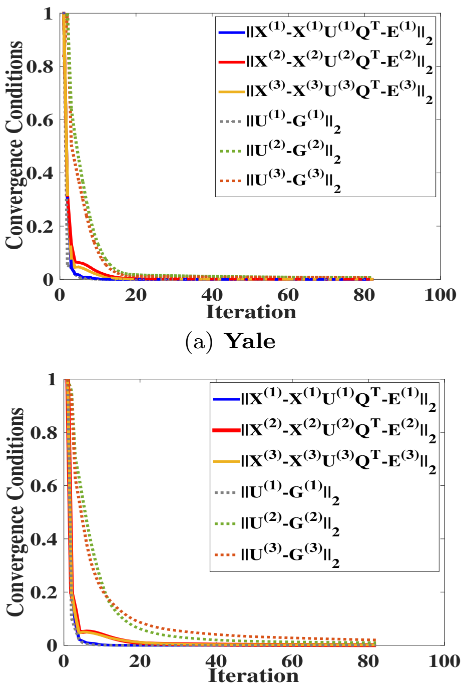
•**CLRR** [32]. The method ensures that data sharing a must-link constraint or same label have the same coordinates in the new representation.

Fig. 7 shows the affinity matrices on 4 datasets. The visualization shows the block-diagonal structure clearly which makes the results of subspace clustering more accurately. Moreover, with the increase of the supervision ratios (denoted by sup_ratio), the block-diagonal structure of the affinity matrices becomes much clearer.

Fig. 8 shows the comparison among diverse methods with respect to must-link constraints under different ratios of supervised information in terms of clustering accuracy and NMI. Obviously, it is observed that the clustering performances becomes better with the increase of supervised information. Note that, COSC achieves a promising performances and defeats these subspace clustering methods, while our algorithm gains a competitive results. We also note that the improvements of SemiSMR are not such significant as others with the increase of supervision ratios on ORL and COIL-20, while our competitive results further validates the effectiveness of our method.

Table 2: Computation cost on COIL-20.

Method	SPC	LRR	RMSC	MSSC	LT-MSC
Time (s)	3.84	276.19	389.78	1833	452.25



(a) **Yale**

(b) **Extended YaleB**

Figure 10: Convergence experiment on Yale and Extended YaleB.

4.4 Parameter Tuning, Convergence & Computational Cost

The experiments of parameter tuning are shown in Fig. 9. In our model, there is only one parameter λ to be tuned. We tune it on 4 benchmark datasets. Overall, the performances with regularization $\lambda > 0$ are better than $\lambda = 0$ on all 4 datasets, which demonstrates the effectiveness of the low-rank tensor constraint. Moreover, we can find an reasonable parameter interval for each dataset to achieve a relatively promising performance. However, different dataset has a distinctly different parameter interval. For instance, our method on ORL performs well with a small λ, which indicates a slight constraint is sufficient to cluster the ORL data. While for Extended YaleB, a much larger λ is needed.

Generally, it is difficult to select a value for the parameter λ in advance for a new dataset because there is no validation set guiding the selection as in supervised task. Even though, we provide a possible way to guide the hyper-parameter selection for clustering. Specifically, since the label information is invalid, we introduce an internal evaluation scheme, where the clustering result is evaluated by using quantities and features inherent to the dataset. We used the Davies-Bouldin index (DBI) [19] as a metric for evaluating clustering algorithms without using label information, where the smaller value in terms of DBI indicates the better clustering result. As shown in the Fig. 9, the promising performances in terms of external metric (e.g., ACC and NMI) are usually consistent with those of internal metric, demonstrating the effectiveness of the proposed parameter selection strategy. Moreover, other internal or combination of multiple internal metrics could be considered in the future.

Fig. 10 shows the convergence experiments on Yale and Extended YaleB. We normalize the values of convergence conditions to $(0, 1)$. The results validate that our algorithm can achieve convergence within a few iterations.

We report the results about computational time of the representative multi-view learning methods on COIL-20, as shown in Table 2. All the methods are tested on a computer with Intel(R) Core(TM) i5-8400 CPU and 8.00GB RAM. Because graph (with the size n × n) is involved for existing subspace clustering methods, it leads to computational cost matrix operations. The time complexities of these subspace-based clustering methods are generally in the same level. We observe that the spectral clustering is much faster because it does not require a number of iterations.

5 Conclusion

We introduced a framework to learn representation for multi-view data by exploiting the complementary information from multiple views. The tensor is introduced to explore high-order correlations of multi-view data, and a constraint matrix is devised to further promote the learned representation. We formulated the problem within a unified optimization framework and proposed an efficient algorithm to obtain the optimal solution. The extensive experimental results validate the effectiveness of the proposed method in exploring high-order correlations and prior information.

References

1. M. Abavisani and V. M. Patel. Multimodal sparse and low-rank subspace clustering. *Information Fusion*, 39:168–177, 2018.
2. G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *ICML*, pages 1247–1255, 2013.
3. R. H. Bartels and G. W. Stewart. Solution of the matrix equation AX + XB = C. *Communications of the ACM*, 15(9):820–826, 1972.
4. S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *ACM SIGKDD*, pages 59–68, 2004.
5. S. Bickel and T. Scheffer. Multi-view clustering. In *ICDM*, 2004.
6. M. B. Blaschko and C. H. Lampert. Correlational spectral clustering. In *CVPR*, 2008.
7. J. F. Cai, Cand, E. J. S, and Z. Shen. *A Singular Value Thresholding Algorithm for Matrix Completion*. Society for Industrial and Applied Mathematics, 2010.
8. X. Cao, X. Wei, Y. Han, Y. Yang, and D. Lin. Robust tensor clustering with non-greedy maximization. In *AAAI*, 2013.
9. X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang. Diversity-induced multi-view subspace clustering. In *CVPR*, 2015.
10. X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh. Constrained multi-view video face clustering. *TIP*, 24(11):4381 – 4393, 2015.
11. X. Cao, C. Zhang, C. Zhou, H. Fu, and H. Foroosh. Constrained multi-view video face clustering. *IEEE Transactions on Image Processing*, 24(11):4381 – 43937, 2015.
12. K. Chaudhuri and S. M. Kakade. Multi-view clustering via canonical correlation analysis. In *ICML*, 2009.
13. K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *ICML*, 2009.
14. B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *ICCV*, 2011.
15. M. Cheng, L. Jing, and M. K. Ng. Tensor-based low-dimensional representation learning for multi-view clustering. *IEEE Transactions on Image Processing*, 28(5):2399–2414, 2018.
16. M. Christopher D., P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*, volume 1. Cambridge university press Cambridge, 2008.
17. M. D. Collins, J. Liu, J. Xu, L. Mukherjee, and V. Singh. Spectral clustering with a convex regularizer on millions of images. In *ECCV*, 2014.
18. C. Cortes, M. Mohri, and A. Rostamizadeh. Learning non-linear combination of kernels. In *NIPS*, 2009.
19. D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.
20. L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(r_1, r_2,..., r_n) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.
21. V. R. de Sa. Spectral clustering with two views. In *ICML*, 2005.
22. Z. Ding, H. Zhao, and Y. Fu. *Learning representation for multi-view data analysis: models and applications*. Springer, 2018.
23. E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.
24. A. M. Elkahky, Y. Song, and X. He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*, pages 278–288. International World Wide Web Conferences Steering Committee, 2015.
25. H. Gao, F. Nie, X. Li, and H. Huang. Multi-view subspace clustering. In *ICCV*, pages 4238–4246, 2015.
26. D. Greene and P. Cunningham. A matrix factorization approach for integrating multiple data views. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 423–438. Springer, 2009.
27. T. Grigorios and L. Aristidis. Kernel-based weighted multi-view clustering. In *ICDM*, 2012.

28. Y. Guo. Convex subspace representation learning from multi-view data. In *AAAI*, 2013.
29. N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 52(2):217–288, 2011.
30. H. Hu, Z. Lin, J. Feng, and J. Zhou. Smooth representation clustering. In *CVPR*, 2014.
31. H. Hu, Z. Lin, J. Feng, and J. Zhou. Smooth representation clustering. In *CVPR*, pages 3834–3841, 2014.
32. W. Jing, W. Xiao, T. Feng, H. L. Chang, and H. Yu. Constrained low-rank representation for robust subspace clustering. *IEEE Transactions on Cybernetics*, PP(99):1–13, 2016.
33. K. Kamvar, S. Sepandar, K. Klein, D. Dan, M. Manning, and C. Christopher. Spectral learning. In *IJCAI*, pages 561–566, 2003.
34. A. Kumar and H. Daumé III. A co-training approach for multi-view spectral clustering. In *ICML*, 2011.
35. A. Kumar, P. Rai, and H. Daumé III. Co-regularized multi-view spectral clustering. In *NIPS*, 2011.
36. M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.
37. L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
38. H. Lawrence and A. Phipps. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.
39. Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1010.0789*, 2010.
40. Z. Lin, R. Liu, and H. Li. Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning. *Machine Learning*, 99(2):287–325, 2015.
41. G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):171–184, 2013.
42. H. Liu, Z. Wu, D. Cai, and T. S. Huang. Constrained nonnegative matrix factorization for image representation. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1299–1311, 2012.
43. J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2013.
44. X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao. Late fusion incomplete multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
45. Z. Lu and T. K. Leen. Penalized probabilistic clustering. *Neural Computation*, 19(6):1528–1567, 2007.
46. A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.
47. J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, pages 689–696, 2011.
48. T.-H. Oh, Y. Matsushita, Y.-W. Tai, and I. S. Kweon. Fast randomized singular value thresholding for nuclear norm minimization. In *CVPR*, 2015.
49. T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.
50. V. M. Patel and H. V. Nguyen. Latent space sparse subspace clustering. In *ICCV*, pages 225–232, 2014.
51. E. Quintana. A note on parallel matrix inversion. *SIAM Journal on Scientific Computing*, 22(5):1762–1771, 2001.
52. S. S. Rangapuram and M. Hein. Constrained 1-spectral clustering. *AISTATS*, 2012.
53. F. Soleymani. A fast convergent iterative solver for approximate inverse of matrices. *Numerical Linear Algebra with Applications*, 21(3):439–452, 2013.
54. W. Tang, Z. Lu, and I. S. Dhillon. Clustering with multiple graphs. In *ICDM*, 2009.
55. R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010.
56. H. Tong, J. He, M. Li, C. Zhang, and W.-Y. Ma. Graph based multi-modality learning. In *ACM MM*, pages 862–871. ACM, 2005.
57. K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.
58. H. Wang, C. Weng, and J. Yuan. Multi-feature spectral clustering with minimax optimization. In *CVPR*, 2014.
59. H. Wang, Y. Yang, and B. Liu. Gmc: graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2019.
60. W. Wang, R. Arora, K. Livescu, and J. Bilmes. On deep multi-view representation learning. pages 1083–1092, 2015.
61. M. White, X. Zhang, D. Schuurmans, and Y.-l. Yu. Convex multi-view subspace learning. In *NIPS*, 2012.
62. B. Wu, Y. Zhang, B. Hu, and Q. Ji. Constrained clustering and its application to face clustering in videos. In *CVPR*, pages 3507–3514, 2013.
63. R. Xia, Y. Pan, L. Du, and J. Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, pages 2149–2155, 2014.
64. R. Xia, Y. Pan, L. Du, and J. Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, 2014.
65. Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, and Y. Qu. On unifying multi-view self-representations for clustering by tensor multi-rank minimization. *International Journal of Computer Vision*, 126(11):1157–1179, 2018.
66. K. Zhan, C. Zhang, J. Guan, and J. Wang. Graph learning for multiview clustering. *IEEE transactions on cybernetics*, 48(10):2887–2895, 2017.
67. C. Zhang, E. Adeli, T. Zhou, X. Chen, and D. Shen. Multi-layer multi-view classification for alzheimers disease diagnosis. In *AAAI*, 2018.
68. C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu. Generalized latent multi-view subspace clustering. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
69. C. Zhang, H. Fu, Q. Hu, P. Zhu, and X. Cao. Flexible multi-view dimensionality co-reduction. *IEEE Transactions on Image Processing*, 26(2):648–659, 2017.
70. C. Zhang, Q. Hu, H. Fu, P. Zhu, and X. Cao. Latent multi-view subspace clustering. In *CVPR*, pages 4333–4341, 2017.
71. T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja. Low-rank sparse coding for image classification. In *ICCV*, 2013.

72. T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem. Robust visual tracking via consistent low-rank sparse learning. *International Journal of Computer Vision*, 111(2):171–190, 2014.
73. Z. Zhang, L. Liu, F. Shen, H. T. Shen, and L. Shao. Binary multi-view clustering. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1774–1782, 2018.
74. H. Zhao, Z. Ding, and Y. Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927, 2017.
75. X. Zhao, N. Evans, and J.-L. Dugelay. A subspace co-training framework for multi-view clustering. *Pattern Recognition Letters*, 41:73–82, 2014.
76. C. Zhou, C. Zhang, X. Li, G. Shi, and X. Cao. Video face clustering via constrained sparse representation. In *ICME*.