

Exsavi: Excavating both sample-wise and view-wise relationships to boost multi-view subspace clustering

Haiyan Wang^a, Guoqiang Han^a, Bin Zhang^a, Guihua Tao^a, Hongmin Cai^{a,b,c,*}

^a School of Computer Science and Engineering, South China University of Technology, GuangZhou 510006, China

^b Guangdong Provincial Key Lab of Computational Intelligence and Cyberspace Information, Guangzhou 510006, China

^c Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan

ARTICLE INFO

Article history:

Received 19 January 2020

Revised 1 July 2020

Accepted 9 July 2020

Available online 25 July 2020

Communicated by Deng Cai

Keywords:

Multi-view learning

Clustering

Representation learning

Tensor subspace analysis

ABSTRACT

Multi-view clustering aims to partition objects based on multiple views through unsupervised learning. To exploit cross-view information, recent advances have shifted the focus from matrix-based to tensor-based subspace learning. Though, both approaches leverage the subspace representation of the data, further higher-order statistics extraction is often overlooked. To tackle the issue, in this paper, a novel multi-view clustering method is proposed to extract both intrinsic sample-wise and view-wise statistics from multi-view data. The multi-view data is represented in tensor form and mapped into a latent tensor subspace to exploit its sample-wise relationship. Through using multi-dimensional sparse coding in its similarity tensor, the view-wise relationship is exploited and evacuated. The two relationships are jointly learned and a latent clustering structure is essentially captured in a data-driven way. We conducted extensive experiments on face, object, digital image, and text datasets and compared with ten state-of-the-art methods. The experimental results demonstrated that the proposed method, named as Exsavi, outperform the baselines in terms of various evaluation metrics.

© 2020 Published by Elsevier B.V.

1. Introduction

Multi-view clustering problem has been spurred into a researching hot spot. It is a challenging issue to learn an informative representation by analyzing all the views simultaneously [1]. In recent years, many excellent clustering algorithms have been proposed [2–9].

Recent advances in multi-view clustering can generally be divided into two major approaches: matrix and tensor oriented. The matrix-oriented method data from each view as a separate matrix, and the subsequent learning is mainly based on matrix theory. Since the multi-view learning problem is non-determinant, various constraints are penalized to achieve pre-defined requirements, such as sparsity [10] and low-rank [11]. Liu et al. [12] used popular non-negative matrix factorization (NMF) to search for a factorization that gives compatible clustering solutions across multiple views. Lu et al. [4] presented a convex sparse multi-view spectral clustering method to jointly learn the affinity matrices from different views. Xia et al. [13] proposed to recover a shared transition probability matrix through Markov chain learning.

* Corresponding author at: School of Computer Science and Engineering, South China University of Technology, GuangZhou 510006, China.

E-mail address: hmc@scut.edu.cn (H. Cai).

However, previous multi-view subspace clustering works address the problem almost by constructing an affinity matrix on each view separately and can not simultaneously explore all possible correlations within views.

The multi-view data can be naturally represented by a tensor. Therefore, recent advances in tensor-based subspace clustering have attracted considerable attention [14,15–17]. Cheng et al. [1] introduced a tensor-based representation learning method for multi-view clustering using a sparsity constraint, which unifies high-dimensional multi-view feature spaces into a low-dimensional shared latent feature space. Zhang et al. [14] presented a low-rank tensor constrained multi-view subspace clustering method, which performs multi-view clustering via integrating the subspace representation matrices of all views into a tensor to discover the structure of data. Yin et al. [15,16] developed a multi-view subspace clustering via tensorial t -product representation, which executes self-expressive tensor learning using sparsity and low-rank constraints. Xie et al. [17] imposed a new type of low-rank tensor constraint on a rotated tensor to capture the low-rank tensor subspace.

However, existing multi-view clustering methods have their limitations. For matrix-oriented methods, the different views are processed independently and thus the possible intrinsic

relationship among different views is mostly overlooked. In comparison, tensor-based clustering methods provide a natural way to exploit multi-view mutual information. However, the view-wise relationship has not been extracted. To address these drawbacks, we propose a novel multi-view clustering method, called Exsavi. The proposed method fully exploits both the sample-wise relationship using subspace learning and view-wise relationship using sparse coding. The overview of our proposed method is shown in Fig. 1.

Our contributions and novelties are summarized as follows:

- We propose a novel multi-view clustering method that seamlessly combines self-expressive tensor learning and multi-dimensional sparse representation.
- The proposed method fully exploits both the sample-wise and view-wise intrinsic relationships for multi-view data. The two relationships are jointly learned and are driven by the clustering data to essentially capture the objective clustering structure.
- We applied the method to four real-world datasets. Extensive experimental results demonstrated its superior performance in comparison with ten baseline methods.

The remainder of this paper is structured as follows. Section 2 briefly overviews the related tensor-based preliminaries for this paper. In Section 3, an original multi-view clustering approach is proposed. In Section 4, we present the experimental results and analysis to demonstrate the superior performance of the proposed method for multi-view clustering. Finally, we provide concluding remarks in Section 5.

2. Notations and related work

This section reviews the preliminaries of self-expressive tensor learning and multi-dimensional sparse representation.

2.1. Notations

Throughout the paper, light lowercase letters (e.g., x) denote scalars, bold lowercase letters (e.g., $\mathbf{x}$) denote vectors, bold uppercase letters (e.g., $\mathbf{X}$) denote matrices, and bold calligraphic letters (e.g., $\mathcal{X}$) denote tensors.

$\mathbf{0}$ denotes the null tensor depending on the context. $\mathbf{I}$ denotes the identity matrix. $\mathbf{I}$ denotes the identity tensor whose first frontal slice is identity matrix and other frontal slices are all zeros. The i th column of matrix $\mathbf{X}$ is denoted as $\mathbf{x}_i$. Besides, $\text{vec}(\mathbf{X})$ is used for the vectorization of a matrix, $\text{vec}(\mathbf{X})$ for the vectorization of a tensor.

Tensors can be viewed as multiway arrays. We use $\mathbf{X}(i, :, :)$, $\mathbf{X}(:, j, :)$, and $\mathbf{X}^{(k)} = \mathbf{X}(:, :, k)$ to denote the i th horizontal slice, j th lateral slice, and k -th frontal slice of $\mathbf{X}$, respectively. Vectors $\mathbf{X}(:, j, k)$, $\mathbf{X}(i, :, k)$, and $\mathbf{X}(i, j, :)$ are called by the 1st dimension, 2nd dimension and 3rd dimension fiber of $\mathbf{X}$, respectively. We use $\hat{\mathbf{X}} = \text{fft}(\mathbf{X}, [], 3)$ to denote the fast Fourier transform (FFT) along the 3rd dimension for $\mathbf{X}$.

Definition 2.1. (Vector ℓ_p and ℓ_0 norm): For a vector $\mathbf{x} \in \mathbb{R}^n$, its ℓ_p norm $\|\mathbf{x}\|_p$ ($p \in [1, +\infty)$) is defined as $\|\mathbf{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$, and its ℓ_0 norm $\|\mathbf{x}\|_0$ is defined as $\|\mathbf{x}\|_0 = \#\{i | x_i \neq 0\}$.

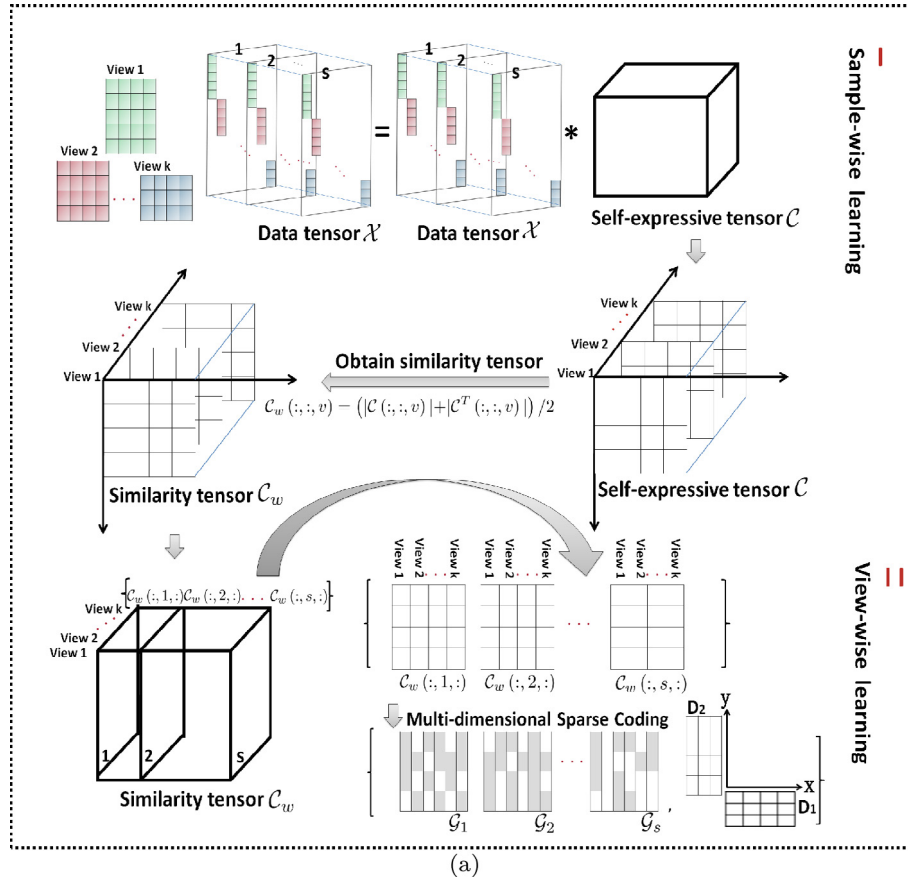


Fig. 1. Flowchart of the proposed method.

Definition 2.2. (Frobenius norm): For a matrix $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$, its Frobenius norm is defined as $\|\mathbf{X}\|_F = (\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} |\mathbf{x}_{ij}|^2)^{1/2}$. The Frobenius norm for a tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined similarly, $\|\mathbf{X}\|_F = (\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} |\mathbf{X}(i, j, k)|^2)^{1/2}$.

Definition 2.3. (F1 norm and FF1 norm): For a tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its F1 norm is defined as $\|\mathbf{X}\|_{F1} = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \|\mathbf{X}(i, j, :)\|_F$, which is equal to the sum of the Frobenius norms of the tubes. Similarly, the tensor FF1 norm can be defined by computing the sum of the Frobenius norms of the horizontal slices, $\|\mathbf{X}\|_{FF1} = \sum_{i=1}^{n_1} \|\mathbf{X}(i, :, :)\|_F$.

Definition 2.4. (Tensor nuclear norm): For a tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its tensor nuclear norm (TNN) is defined as $\|\mathbf{X}\|_{TNN} = (1/n_3) \sum_{k=1}^{n_3} \|\mathbf{X}(:, :, k)\|_*$, where $\|\cdot\|_*$ denotes the matrix nuclear norm, i.e., sum of singular values of the matrix.

Definition 2.5. (t-product): For two third-order tensors $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathbf{C} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, the t-product of the two tensors is denoted by:

$$\mathbf{X} * \mathbf{C} = \text{fold}(\text{bcirc}(\mathbf{X}) \text{bvec}(\mathbf{C})) \in \mathbb{R}^{n_1 \times n_4 \times n_3}, \quad (1)$$

where $\text{bcirc}(\mathbf{X})$ is the block circulant matrix obtained from n_3 frontal slices of tensor $\mathbf{X}$ as

$$\text{bcirc}(\mathbf{X}) = \begin{bmatrix} \mathbf{X}^{(1)} & \mathbf{X}^{(n_3)} & \mathbf{X}^{(n_3-1)} & \dots & \mathbf{X}^{(2)} \\ \mathbf{X}^{(2)} & \mathbf{X}^{(1)} & \mathbf{X}^{(n_3)} & \dots & \mathbf{X}^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{X}^{(n_3)} & \mathbf{X}^{(n_3-1)} & \mathbf{X}^{(n_3-2)} & \dots & \mathbf{X}^{(1)} \end{bmatrix}, \quad (2)$$

$\text{bvec}(\mathbf{C})$ vectorizes the tensor $\mathbf{C}$ along its frontal slices,

$$\text{bvec}(\mathbf{C}) = \begin{bmatrix} \mathbf{C}^{(1)} \\ \mathbf{C}^{(2)} \\ \vdots \\ \mathbf{C}^{(n_3)} \end{bmatrix}, \quad (3)$$

and $\text{fold}(\text{bvec}(\mathbf{C})) = \mathbf{C}$ back to a tensor form.

Definition 2.6. (Kronecker product): For two matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$ and $\mathbf{B} \in \mathbb{R}^{n_3 \times n_4}$, the Kronecker product of $\mathbf{A}$ and $\mathbf{B}$ is defined as the matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{a}_{11}\mathbf{B} & \dots & \mathbf{a}_{1n_2}\mathbf{B} \\ \vdots & \ddots & \vdots \\ \mathbf{a}_{n_11}\mathbf{B} & \dots & \mathbf{a}_{n_1n_2}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{n_1 n_3 \times n_2 n_4}. \quad (4)$$

Definition 2.7. (n-mode product): For a third-order tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ and a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_n}$, the n-mode product of $\mathbf{X} \times_n \mathbf{A} = \mathbf{Y} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N}$ is denoted by

$$\mathbf{Y}_{i_1 \times \dots \times i_{n-1} \times j \times i_{n+1} \times \dots \times i_N} = \sum_{i_n=1}^{I_n} \mathbf{x}_{i_1 i_2 \dots i_n} \mathbf{a}_{j i_n}. \quad (5)$$

There is an equivalent formulation in matrix form [18]:

$$\mathbf{Y} = \mathbf{X} \times_n \mathbf{A} \iff \mathbf{Y}_{(n)} = \mathbf{A} \mathbf{X}_{(n)}. \quad (6)$$

where $\mathbf{X}_{(n)}$ is the n th dimensional unfolding matrix obtained via arranging all the n -th dimensional fibers of $\mathbf{X}$ as the columns of the matrix, and is denoted by $\mathbf{X}_{(n)} = \text{unflod}_n(\mathbf{X}) \in \mathbb{R}^{I_n \times (\prod_{k \neq n} I_k)}$.

2.2. Self-expressive tensor learning

In this paper, we use the third-order tensor to represent multi-view data. Consider multi-view dataset $\{\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times s}, v = 1, 2, \dots, k\}$ from k views, with d_v features and s samples in the v th view. For each sample across different views, it is reformatted into a $D_f \times k$ block-diagonal matrix, whose v th column corresponds to features from the v -th view and $D_f = \sum_{v=1}^k d_v$. Through twist manipulation [19], the multi-view data for the i -th sample can be easily transformed into a $D_f \times 1 \times k$ third-order tensor. Then, we can obtain a tensor $\mathbf{X} \in \mathbb{R}^{D_f \times s \times k}$ for multi-view data via collecting all tensors along the second dimension.

To capture the global structure and high-order correlation hidden in multi-view data, recent works have shown that tensor-based self-representation learning is an effective way [20,15,16]. For example, Kernfeld et al. [20] proposed a tensor-based self-representation clustering method as follows:

$$\min_{\mathbf{C}} \mathcal{L}(\mathbf{C}|\mathbf{X}) = \min_{\mathbf{C}} \|\mathbf{C}\|_{F1} + \lambda_1 \|\mathbf{C}\|_{FF1} + \lambda_2 \|\mathbf{X} - \mathbf{X} * \mathbf{C}\|_F^2, \quad (7)$$

where $\mathbf{C}$ is the sparse representation tensor, the first term $\|\cdot\|_{F1}$ and second term $\|\cdot\|_{FF1}$ are to enforce the sparse constraint on the self-expressive tensor $\mathbf{C}$, so that the relevant important data objects are selected for the reconstruction of each cluster. λ_1 and λ_2 are trade off parameters.

Yin et al. [16] introduced a low-rank constraint on the self-expressive tensor to ensure a decomposable representation using the following model:

$$\min_{\mathbf{C}} \mathcal{L}(\mathbf{C}|\mathbf{X}) = \min_{\mathbf{C}} \lambda_1 \|\mathbf{C}\|_{F1} + \lambda_2 \|\mathbf{C}\|_{TNN} + \frac{1}{2} \|\mathbf{X} - \mathbf{X} * \mathbf{C}\|_F^2 + \frac{\lambda_3}{2} \sum_{1 \leq v, v_2 \leq k, v \neq v_2} \|\mathbf{C}(:, :, v) - \mathbf{C}(:, :, v_2)\|_F^2, \quad (8)$$

where $\mathbf{C} \in \mathbb{R}^{s \times s \times k}$ denotes the representation tensor, and the first term $\|\cdot\|_{F1}$ and second term $\|\cdot\|_{TNN}$ are the tensor sparse and nuclear norm, respectively. The last term makes the consensus clustering by compelling all the view coefficients to be close to each other. λ_1, λ_2 , and λ_3 are the parameters to trade off these terms.

2.3. Multi-dimensional sparse coding

Sparse coding is an effective signal representation technique and acquires numerous promising and significant impact in computation vision [21]. However, classical sparse representation destroys structural information among multi-dimensional signals if modeling the signals directly into a one-dimensional(1-D) vector. To alleviate this problem, a multi-dimensional signal synthesis sparse model (MD-SSM) was developed [22,23].

For an N -order tensor signal $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, the MD-SSM aims to extract sparse core tensor $\mathbf{G} \in \mathbb{R}^{K_1 \times K_2 \times \dots \times K_N}$ satisfying,

$$\mathbf{X} = \mathbf{G} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \times_3 \dots \times_N \mathbf{D}_N, \|\mathbf{G}\|_0 = \tau, \quad (9)$$

where sparsity is attained by a ℓ_0 norm, that is, $\|\mathbf{G}\|_0$, and factor matrix $\mathbf{D}_n \in \mathbb{R}^{I_n \times K_n}$ is the n -th dimensional dictionary.

Let the i th sample be denoted by $\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$. The MD-SSM model aims to extract its sparse tensor representation by minimizing the approximation error:

$$\min_{\{\mathbf{G}_i\}_{i=1}^s, \{\mathbf{D}_n\}_{n=1}^N} \mathbf{L}(\{\mathbf{G}_i\}_{i=1}^s, \{\mathbf{D}_n\}_{n=1}^N | \{\mathbf{X}_i\}_{i=1}^s) \\ = \sum_{i=1}^s \|\mathbf{X}_i - \mathbf{G}_i \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \times_3 \cdots \times_N \mathbf{D}_N\|_F^2 \quad (10)$$

$$\text{s.t. } \|\mathbf{G}_i\|_0 \leq \tau, \quad i = 1, 2, \dots, s$$

$$\|\mathbf{D}_n(:, r)\|_2^2 = 1, \quad n = 1, 2, \dots, N, \quad r = 1, 2, \dots, K_n,$$

where $\{\mathbf{G}_i\}_{i=1}^s$ and $\mathbf{D}_n$ are sparse core tensors (that belong to s samples) and the n th dimensional dictionary, respectively.

3. Excavating both sample-wise and view-wise relationships to boost multi-view subspace clustering

In this section, we present a novel method for multi-view data clustering. The method exploits both the sample-wise relationship using tensor subspace learning and view-wise relationship using multi-dimensional sparse coding for effective clustering and thus is named as Exsavi.

3.1. Mathematical model formulation

Consider multi-view data $\{\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times s}, v = 1, 2, \dots, k\}$ of k views. Through diagonally integrating the features of all views and twist manipulation, it can be represented by third-order tensor $\mathbf{X} \in \mathbb{R}^{D_f \times s \times k}$.

One can employ the self-representation tensor clustering model (8) to obtain self-representation tensor $\mathbf{C} \in \mathbb{R}^{s \times s \times k}$. Then, the clustering task could be accomplished by working on affinity tensor $\frac{\mathbf{C} + \mathbf{C}^T}{2}$. However, such a scheme suffers from a noteworthy shortcoming. A perfect subspace representation cannot guarantee good clustering performance. For example, the trial solution $\mathbf{C} = \mathbf{I}$ fails to produce reliable clustering. Therefore, the clustering task is actually decoupled. Furthermore, it overlooks the intrinsic correlations among different views.

To exploit the view-dependent correlation, we propose learning an affinity core tensor from the self-representation coefficient rather than performing simple averaging $\frac{\mathbf{C} + \mathbf{C}^T}{2}$. We achieve the goal by extracting a sparse representation on each $\{\mathbf{C}_w(:, i, :)\}_{i=1}^s \in \mathbb{R}^{s \times k \times s}$ through coding using uniform dictionaries:

$$\min_{\{\mathbf{G}_i\}_{i=1}^s, \{\mathbf{D}_n\}_{n=1}^N} \mathbf{L}(\{\mathbf{G}_i\}_{i=1}^s, \{\mathbf{D}_n\}_{n=1}^N | \{\mathbf{C}_w(:, i, :)\}_{i=1}^s) \\ = \sum_{i=1}^s \|\mathbf{C}_w(:, i, :) - \mathbf{G}_i \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2\|_F^2 \quad (11)$$

s.t.

$$\|\mathbf{G}_i\|_0 \leq \tau, \quad i = 1, 2, \dots, s$$

$$\|\mathbf{D}_n(:, r)\|_2^2 = 1, \quad n = 1, 2, \quad r = 1, 2, \dots, K_n,$$

where $\mathbf{C}_w(:, i, :)$ measures the similarity between the i th sample and the other samples on all views. The core tensor $\mathbf{G}_i \in \mathbb{R}^{K_1 \times K_2}$ is the sample affinity, which essentially characterizes the clustering structure. Each $\mathbf{G}_i$ can be considered as a new representation of multi-view data in a similarity tensor subspace, which is task-oriented and highly informative in extracting the view-dependent characteristics.

Finally, by combining self-representation and sparse coding, we propose using the following model to achieve multi-view clustering:

$$\min_{\mathbf{C}, \{\mathbf{G}_i\}_{i=1}^s, \{\mathbf{D}_n\}_{n=1}^N} \mathbf{L}(\mathbf{C}, \{\mathbf{G}_i\}_{i=1}^s, \{\mathbf{D}_n\}_{n=1}^N) \\ = \min_{\mathbf{C}, \{\mathbf{G}_i\}_{i=1}^s, \{\mathbf{D}_n\}_{n=1}^N} \mathbf{L}_1(\mathbf{C} | \mathbf{X}) + \mathbf{L}_2(\{\mathbf{G}_i\}_{i=1}^s, \{\mathbf{D}_n\}_{n=1}^N | \mathbf{C}_w), \\ = \min_{\mathbf{C}, \{\mathbf{G}_i\}_{i=1}^s, \{\mathbf{D}_n\}_{n=1}^N} \lambda_1 \|\mathbf{C}\|_{F_1} + \lambda_2 \|\mathbf{C}\|_{TNN} + \frac{1}{2} \|\mathbf{X} - \mathbf{X} * \mathbf{C}\|_F^2 \\ + \frac{\lambda_3}{2} \sum_{1 \leq v, v_2 \leq k, v \neq v_2} \|\mathbf{C}(:, :, v) - \mathbf{C}(:, :, v_2)\|_F^2 \\ + \lambda_4 \sum_{i=1}^n \|\mathbf{C}_w(:, i, :) - \mathbf{G}_i \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2\|_F^2 \quad (12)$$

s.t.

$$\|\mathbf{G}_i\|_0 \leq \tau, \quad i = 1, 2, \dots, s$$

$$\|\mathbf{D}_n(:, r)\|_2^2 = 1, \quad n = 1, 2, \quad r = 1, 2, \dots, K_n$$

$$\mathbf{C}_w(:, :, v) = (|\mathbf{C}(:, :, v)| + |\mathbf{C}^T(:, :, v)|) / 2$$

The first two terms penalize the tensor representation, thereby making it sparse and low-rank. The third term represents the approximation errors. The fourth term encourages consensus representation between across views. The last is higher-order statistics term, which is used to obtain task-dependent tensor affinity tensor $\mathbf{G}_i$ for each sample i . The affinity tensor is encoded by two dictionaries, $\mathbf{D}_1$ and $\mathbf{D}_2$, along the dimension of the views, thus exploiting intrinsic correlations among different views. $\lambda_1, \lambda_2, \lambda_3$, and λ_4 are trade off parameters.

Comparing with the existing tensor-based multi-view representation clustering models [14,15–17], our scheme aims to extract more efficient higher-order correlation statistics for multi-view data rather than only focusing on tensor self-representation.

Our proposed model enjoys the merits of tensor-based low-dimensional representation learning for multi-view clustering (tRLMvC) [1] and multi-view subspace clustering via tensorial t -product representation (SCMV-3DT) [16]. However, there are two crucial distinctions between tRLMvC and our model. One is that the primary aim of tRLMvC is to identify a low-dimensional multi-view data representation. In comparison, our model aims to identify a similarity tensor subspace $\mathbf{C}_w$ for which the clustering task is highly related. Therefore, our approach is a task-driven learning process and thus rendering superior clustering performances. The other crucial distinction is that the factor components of the tensor decomposition used in clustering are different: tRLMvC aims to extract the combination of the first and second dimension factor matrices ($\mathbf{H} = [\mathbf{U}, \mathbf{V}]$) for clustering using Tucker decomposition, while our method aims to extract the core tensors ($\{\mathbf{G}_i\}_{i=1}^s$) for clustering using the MD-SSM model. SCMV-3DT [16] computes the latent transition probability matrix using self-representation tensor $\mathbf{C}$ and uses the standard Markov chain method to separate data. However, in our work, we introduce a novel part, that is, multi-dimensional sparse representation learning on similarity tensor $\mathbf{C}_w$, to obtain the higher-order statistics representation ($\{\mathbf{G}_i\}_{i=1}^s$) of multi-view objects for clustering.

Our proposed method is also distinct from LT-MSC [14], LRSD-RMSC [13]. The former fails to explicitly consider complementary knowledge from complementary views. The latter recovers a shared low-rank transition probability, which serves as an affinity for the standard Markov chain method to obtain the clustering assignment.

Note that we extract two different scales of correlational information compared with existing methods. The full extraction and utilization of the comprehensive correlation on multi-view data enables clustering to achieve superior performance.

3.2. Numerical scheme

We execute the tensor self-representation learning and higher-order representation learning into two close phases. Similar technique has been used in similar problem, such as SCMV-3DT [16], and has been shown to be effective. That is, we divide the proposed model into two steps, the self-expressive tensor $\mathbf{C}$ is learnt, and then multi-dimensional sparse representation is applied on $\mathbf{C}_w$ to get $\{\mathbf{G}_i\}_{i=1}^S$. Finally, the clustering task is accomplished by the application of spectral clustering on $\text{vec}(\{\mathbf{G}_i\}_{i=1}^S)$.

SubProblem 1: Solving self-expressive tensor $\mathbf{C}$. Let λ_4 equals zero, the minimization problem in (12) can be simplified as

$$\begin{aligned} \min_{\mathbf{C}} & \lambda_1 \|\mathbf{C}\|_{F1} + \lambda_2 \|\mathbf{C}\|_{TNN} + \frac{1}{2} \|\mathbf{X} - \mathbf{X} * \mathbf{C}\|_F^2 \\ & + \frac{\lambda_3}{2} \sum_{1 \leq v, v_2 \leq k, v_2 \neq v} \|\mathbf{C}(:, :, v) - \mathbf{C}(:, :, v_2)\|_F^2. \end{aligned} \quad (13)$$

It can be solved by the alternative direction minimization method (ADMM) [24,16]. By introducing two tensor variables $\mathbf{Z}$ and $\mathbf{Y}$, we have

$$\begin{aligned} \min_{\mathbf{C}, \mathbf{Y}, \mathbf{Z}} & \lambda_1 \|\mathbf{Y}\|_{F1} + \lambda_2 \|\mathbf{Z}\|_{TNN} + \frac{1}{2} \|\mathbf{X} - \mathbf{X} * \mathbf{C}\|_F^2 \\ & + \frac{\lambda_3}{2} \sum_{1 \leq v, v_2 \leq k, v_2 \neq v} \|\mathbf{C}(:, :, v) - \mathbf{C}(:, :, v_2)\|_F^2 \end{aligned} \quad (14)$$

Subject to :

$$\begin{aligned} \mathbf{Y} &= \mathbf{C} \\ \mathbf{Z} &= \mathbf{C}. \end{aligned}$$

The augmented Lagrangian equation for (14) is thus given by

$$\begin{aligned} \min_{\mathbf{C}, \mathbf{Y}, \mathbf{Z}} & \lambda_1 \|\mathbf{Y}\|_{F1} + \lambda_2 \|\mathbf{Z}\|_{TNN} + \frac{1}{2} \|\mathbf{X} - \mathbf{X} * \mathbf{C}\|_F^2 \\ & + \frac{\lambda_3}{2} \sum_{1 \leq v, v_2 \leq k, v_2 \neq v} \|\mathbf{C}(:, :, v) - \mathbf{C}(:, :, v_2)\|_F^2 \\ & + \langle \mathbf{L}_1, \mathbf{Y} - \mathbf{C} \rangle + \langle \mathbf{L}_2, \mathbf{Z} - \mathbf{C} \rangle + \frac{\rho}{2} (\|\mathbf{Y} - \mathbf{C}\|_F^2 + \|\mathbf{Z} - \mathbf{C}\|_F^2), \end{aligned} \quad (15)$$

where $\mathbf{L}_1$ and $\mathbf{L}_2$ are Lagrangian multipliers, $\rho > 0$ is a penalty parameter. $\langle \cdot, \cdot \rangle$ denotes inner product. i.e., $\langle \mathbf{X}_1, \mathbf{X}_2 \rangle = \sum_i \text{tr}(\mathbf{X}_1(:, :, i) \mathbf{X}_2(:, :, i))$.

1) **Update $\mathbf{Z}$** by discarding irrelevant variables except one $\mathbf{Z}$, and the Parseval's theorem tells us that the sum of square of a function equals the sum of square of its Fourier transform [16,25], we have the following subproblem

$$\begin{aligned} \min_{\mathbf{Z}} & \lambda_2 \|\mathbf{Z}\|_{TNN} + \langle \mathbf{L}_2, \mathbf{Z} - \mathbf{C} \rangle + \frac{\rho}{2} (\|\mathbf{Z} - \mathbf{C}\|_F^2) \\ & = \min_{\mathbf{Z}} \frac{\lambda_2}{k} \sum_{v=1}^k \|\hat{\mathbf{Z}}(:, :, v)\|_* + \frac{\rho}{2k} \|\hat{\mathbf{Z}} - \hat{\mathbf{C}} + \frac{1}{\rho} \hat{\mathbf{L}}_2\|_F^2. \end{aligned} \quad (16)$$

Let $\hat{\mathbf{Z}}^{(v)} = \hat{\mathbf{Z}}(:, :, v)$, we can solve $\hat{\mathbf{Z}}$ slice-by-slice along the frontal direction. Viewed in Fourier space, the subproblem of learning task (16) can be optimized by

$$\min_{\hat{\mathbf{Z}}^{(v)}} \lambda_2 \|\hat{\mathbf{Z}}^{(v)}\|_* + \frac{\rho}{2} \|\hat{\mathbf{Z}}^{(v)} - \hat{\mathbf{C}}^{(v)} + \frac{1}{\rho} \hat{\mathbf{L}}_2^{(v)}\|_F^2, \quad (17)$$

which has a closed-form solution using singular value thresholding (SVT) algorithm [26]. After obtaining $\hat{\mathbf{Z}}^{(v)}$ and $\hat{\mathbf{Z}}$, it is easy to obtain $\mathbf{Z}^{t+1}$ via inverse FFT, that is, $\mathbf{Z}^{t+1} = \text{ifft}(\hat{\mathbf{Z}}^{t+1}, [], 3)$.

2) **Update $\mathbf{Y}$** using

$$\min_{\mathbf{Y}} \lambda_1 \|\mathbf{Y}\|_{F1} + \langle \mathbf{L}_1, \mathbf{Y} - \mathbf{C} \rangle + \frac{\rho}{2} (\|\mathbf{Y} - \mathbf{C}\|_F^2). \quad (18)$$

Similarly, the variable $\mathbf{Y}$ can be solved fiber-by-fiber along the third dimension, that is,

$$\mathbf{Y}^{t+1}(i, j, :) = \frac{\|\mathbf{A}(i, j, :)\|_F^{-\frac{\lambda_1}{\rho}}}{\|\mathbf{A}(i, j, :)\|_F} \mathbf{A}(i, j, :), \quad (19)$$

where $\mathbf{A} = \mathbf{C} - \frac{1}{\rho} \mathbf{L}_1$.

3) **Update $\mathbf{C}$** using

$$\begin{aligned} \min_{\mathbf{C}} & \frac{1}{2} \|\mathbf{X} - \mathbf{X} * \mathbf{C}\|_F^2 + \langle \mathbf{L}_1, \mathbf{Y} - \mathbf{C} \rangle + \langle \mathbf{L}_2, \mathbf{Z} - \mathbf{C} \rangle \\ & + \frac{\lambda_3}{2} \sum_{1 \leq v, v_2 \leq k, v_2 \neq v} \|\mathbf{C}(:, :, v) - \mathbf{C}(:, :, v_2)\|_F^2 \\ & + \frac{\rho}{2} (\|\mathbf{Y} - \mathbf{C}\|_F^2 + \|\mathbf{Z} - \mathbf{C}\|_F^2). \end{aligned} \quad (20)$$

Similar to the step in solving the subproblem on $\mathbf{Z}$, the problem (20) can be equivalently solved by the help of FFT:

$$\begin{aligned} \min_{\hat{\mathbf{C}}} & \frac{1}{2} \|\hat{\mathbf{X}} - \hat{\mathbf{X}} \odot \hat{\mathbf{C}}\|_F^2 + \frac{\lambda_3}{2} \sum_{1 \leq v, v_2 \leq k, v_2 \neq v} \|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{C}}^{(v_2)}\|_F^2 \\ & + \frac{\rho}{2} (\|\hat{\mathbf{C}} - \hat{\mathbf{P}}_1\|_F^2 + \|\hat{\mathbf{C}} - \hat{\mathbf{P}}_2\|_F^2), \end{aligned} \quad (21)$$

where $\mathbf{P}_1 = \mathbf{Y} + \frac{1}{\rho} \mathbf{L}_1$, $\mathbf{P}_2 = \mathbf{Z} + \frac{1}{\rho} \mathbf{L}_2$, and $\odot$ denotes point-wise multiplication. Then, we $\hat{\mathbf{C}}$ can be optimized slice-by-slice from the frontal side:

$$\begin{aligned} \min_{\hat{\mathbf{C}}^{(v)}} & \frac{1}{2} \|\hat{\mathbf{X}}^{(v)} - \hat{\mathbf{X}}^{(v)} \hat{\mathbf{C}}^{(v)}\|_F^2 + \frac{\lambda_3}{2} \sum_{v_2, v_2 \neq v} \|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{C}}^{(v_2)}\|_F^2 \\ & + \frac{\rho}{2} (\|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{P}}_1^{(v)}\|_F^2 + \|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{P}}_2^{(v)}\|_F^2). \end{aligned} \quad (22)$$

The problem (22) is convex and thus has a global minimizer. For ease of implementation, we introduce a slack variable $\mathbf{Q} = \{\mathbf{Q}_v\}$, $v = 1, 2, \dots, k$ for solving $\hat{\mathbf{C}} = \{\hat{\mathbf{C}}^{(v)}\}$, $v = 1, 2, \dots, k$. Therefore, the problem is reformulated as

$$\begin{aligned} \min_{\hat{\mathbf{C}}^{(v)}, \mathbf{Q}_v} & \frac{1}{2} \|\hat{\mathbf{X}}^{(v)} - \hat{\mathbf{X}}^{(v)} \hat{\mathbf{C}}^{(v)}\|_F^2 + \frac{\lambda_3}{2} \sum_{v_2, v_2 \neq v} \|\mathbf{Q}_v - \hat{\mathbf{C}}^{(v_2)}\|_F^2 \\ & + \frac{\rho}{2} (\|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{P}}_1^{(v)}\|_F^2 + \|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{P}}_2^{(v)}\|_F^2) \\ \text{s.t. } & \mathbf{Q}_v = \hat{\mathbf{C}}^{(v)}, \quad v = 1, 2, \dots, k. \end{aligned} \quad (23)$$

Its augmented Lagrangian formulation is formulated as follows:

$$\begin{aligned} \min_{\hat{\mathbf{C}}^{(v)}, \mathbf{Q}_v} & \frac{1}{2} \|\hat{\mathbf{X}}^{(v)} - \hat{\mathbf{X}}^{(v)} \hat{\mathbf{C}}^{(v)}\|_F^2 + \frac{\lambda_3}{2} \sum_{v_2, v_2 \neq v} \|\mathbf{Q}_v - \hat{\mathbf{C}}^{(v_2)}\|_F^2 \\ & + \frac{\rho}{2} (\|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{P}}_1^{(v)}\|_F^2 + \|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{P}}_2^{(v)}\|_F^2) \\ & + \frac{\gamma}{2} \|\mathbf{Q}_v - \hat{\mathbf{C}}^{(v)} + \frac{1}{\gamma} \mathbf{W}_v\|_F^2, \end{aligned} \quad (24)$$

where $\mathbf{W}_v$ is an error term for measuring the difference between the slack variable $\mathbf{Q}_v$ and the variable $\hat{\mathbf{C}}^{(v)}$, and $\gamma > 0$ is a penalty parameter.

In solving the minimization problem with respect to the variable $\hat{\mathbf{C}}^{(v)}$, there is an error-correcting term $\sum_{v_2, v_2 \neq v} \|\mathbf{Q}_v - \hat{\mathbf{C}}^{(v_2)}\|_F^2$. The term aims to preserve the consensus representation between across views $\hat{\mathbf{C}}^{(v)}$. It can be rewritten as $\sum_{v_2, v_2 \neq v} \|\hat{\mathbf{C}}^{(v)} - \mathbf{Q}_{v_2}\|_F^2$ by the assumption $\mathbf{Q}_v = \hat{\mathbf{C}}^{(v)}$, $v = 1, 2, \dots, k$. Therefore, we iteratively update the two variables in (24) by.

i) Solving the variable $\hat{\mathbf{C}}^{(v)}$, $v = 1, 2, \dots, k$ through

$$\begin{aligned} \min_{\hat{\mathbf{C}}^{(v)}} & \frac{1}{2} \|\hat{\mathbf{X}}^{(v)} - \hat{\mathbf{X}}^{(v)} \hat{\mathbf{C}}^{(v)}\|_F^2 + \frac{1}{2} \lambda_3 \sum_{v_2, v_2 \neq v} \|\hat{\mathbf{C}}^{(v)} - \mathbf{Q}_{v_2}\|_F^2 \\ & + \frac{1}{2} \rho \left(\|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{P}}_1^{(v)}\|_F^2 + \|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{P}}_2^{(v)}\|_F^2 \right) \\ & + \frac{\gamma}{2} \|\mathbf{Q}_v - \hat{\mathbf{C}}^{(v)} + \frac{1}{\gamma} \mathbf{W}_v\|_F^2, \end{aligned} \quad (25)$$

Equivalently,

$$\begin{aligned} \min_{\hat{\mathbf{C}}^{(v)}} & \frac{1}{2} \|\hat{\mathbf{X}}^{(v)} - \hat{\mathbf{X}}^{(v)} \hat{\mathbf{C}}^{(v)}\|_F^2 + \frac{1}{2} \lambda_3 (k-1) \|\hat{\mathbf{C}}^{(v)} - \frac{1}{k-1} \sum_{v_2, v_2 \neq v} \mathbf{Q}_{v_2}\|_F^2 \\ & + \frac{1}{2} \rho \left(\|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{P}}_1^{(v)}\|_F^2 + \|\hat{\mathbf{C}}^{(v)} - \hat{\mathbf{P}}_2^{(v)}\|_F^2 \right) + \frac{\gamma}{2} \|\mathbf{Q}_v - \hat{\mathbf{C}}^{(v)} + \frac{1}{\gamma} \mathbf{W}_v\|_F^2. \end{aligned} \quad (26)$$

Setting the gradient with respect to $\hat{\mathbf{C}}^{(v)}$ to zero, we have

$$\hat{\mathbf{C}}^{(v)} = \mathbf{M}_1^{-1} \mathbf{M}_2, \quad (27)$$

where $\mathbf{M}_1 = \left((\lambda_3(k-1) + \gamma + 2\rho) \mathbf{I} + \left(\hat{\mathbf{X}}^{(v)} \right)^T \hat{\mathbf{X}}^{(v)} \right)$, and

$$\mathbf{M}_2 = \lambda_3 \sum_{v_2, v_2 \neq v} \mathbf{Q}_{v_2} + \gamma \left(\mathbf{Q}_v + \frac{1}{\gamma} \mathbf{W}_v \right) + \left(\hat{\mathbf{X}}^{(v)} \right)^T \hat{\mathbf{X}}^{(v)} + \rho \left(\hat{\mathbf{P}}_1^{(v)} + \hat{\mathbf{P}}_2^{(v)} \right).$$

ii) Solving the variable $\mathbf{Q}_v$, $v = 1, 2, \dots, k$ through

$$\min_{\mathbf{Q}_v} \frac{\lambda_3}{2} \sum_{v_2, v_2 \neq v} \|\mathbf{Q}_v - \hat{\mathbf{C}}^{(v_2)}\|_F^2 + \frac{\gamma}{2} \|\mathbf{Q}_v - \hat{\mathbf{C}}^{(v)} + \frac{1}{\gamma} \mathbf{W}_v\|_F^2. \quad (28)$$

Setting the gradient with respect to $\mathbf{Q}_v$ to zero, we have

$$(\lambda_3(k-1) + \gamma) \mathbf{Q}_v = \lambda_3 \sum_{v_2, v_2 \neq v} \hat{\mathbf{C}}^{(v_2)} + \gamma \hat{\mathbf{C}}^{(v)} - \mathbf{W}_v. \quad (29)$$

iii) Correcting the error term $\mathbf{W}_v$ by

$$\mathbf{W}_v = \mathbf{W}_v + \gamma \left(\mathbf{Q}_v - \hat{\mathbf{C}}^{(v)} \right). \quad (30)$$

iv) **Update $\mathbf{L}_1$ and $\mathbf{L}_2$** using

$$\begin{aligned} \mathbf{L}_1 &= \mathbf{L}_1 + \mathbf{u}(\mathbf{Y} - \mathbf{C}) \\ \mathbf{L}_2 &= \mathbf{L}_2 + \mathbf{u}(\mathbf{Z} - \mathbf{C}) \\ \rho &= \min(\rho_{\max}, \mathbf{u}\rho), \end{aligned} \quad (31)$$

where we update ρ after each iteration by a ratio of $\mathbf{u}$ until they reach a pre-defined maximum value $\rho_{\max}$.

The algorithm for solving $\mathbf{C}$ is summarized in Algorithm 1, and the stopping condition is set by $\max(\|\mathbf{Y}^{t+1} - \mathbf{C}^{t+1}\|_F / \|\mathbf{X}\|_F, \|\mathbf{Z}^{t+1} - \mathbf{C}^{t+1}\|_F / \|\mathbf{X}\|_F, \|\mathbf{Y}^{t+1} - \mathbf{Y}^t\|_F / \|\mathbf{Y}^t\|_F, \|\mathbf{Z}^{t+1} - \mathbf{Z}^t\|_F / \|\mathbf{Z}^t\|_F, \|\mathbf{C}^{t+1} - \mathbf{C}^t\|_F / \|\mathbf{C}^t\|_F) \leq \epsilon$.

Algorithm 1: Solve $\mathbf{C}$ via ADMM on (13)

Input: $\mathcal{X}$, λ_1 , λ_2 , and λ_3 .

Output: $\mathcal{C}^*$.

1: **Initialization:** $\mathcal{C}^0 = \mathcal{Y}^0 = \mathcal{Z}^0 = \mathbf{0}$, $\mathcal{L}_1^0 = \mathcal{L}_2^0 = \mathbf{0}$,

$\rho = \gamma = 0.01$, $\mathbf{u} = 1.9$.

2: **while** the stopping condition is not satisfied ($t = 0, 1, \dots$) **do**

3: Update $\mathbf{Z}^{t+1}$ by solving (16)

4: Update $\mathbf{Y}^{t+1}$ by solving (18)

5: Update $\mathbf{C}^{t+1}$ by solving (20)

6: Update $\mathbf{L}_1$, $\mathbf{L}_2$, and ρ by solving (31)

7: Check the stopping condition

8: **end while**

SubProblem 2: Solving novel higher-order representation $\{\mathbf{G}_i\}_{i=1}^s$. By fixing $\mathbf{C}$ and $\mathbf{C}_w$ to solve variables $\{\mathbf{G}_i\}_{i=1}^s$, the objective equation in (12) can be equivalently transformed as a MD synthesis sparse representation problem:

$$\begin{aligned} \min_{\{\mathbf{G}_i\}_{i=1}^s, \{\mathbf{D}_n\}_{n=1}^2} & \sum_{i=1}^s \|\mathbf{C}_w(:, i, :) - \mathbf{G}_i \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2\|_F^2 \\ \text{s.t.} & \quad \|\mathbf{G}_i\|_0 \leq \tau, \quad i = 1, 2, \dots, s \\ & \quad \|\mathbf{D}_n(:, r)\|_2^2 = 1, \quad n = 1, 2, \quad r = 1, 2, \dots, K_n. \end{aligned} \quad (32)$$

Problem (32) can be minimized by the alternating optimization strategy cycling over tensor sparse coding and dictionary updating .2.1) **Sparse coding.** By providing all dictionaries $\{\mathbf{D}_n\}_{n=1}^2$, the tensor sparse coding $\{\mathbf{G}_i\}_{i=1}^s$ are optimized by

$$\begin{aligned} \min_{\{\mathbf{G}_i\}_{i=1}^s} & \sum_{i=1}^s \|\mathbf{C}_w(:, i, :) - \mathbf{G}_i \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2\|_F^2 \\ \text{s.t.} & \quad \|\mathbf{G}_i\|_0 \leq \tau, \quad i = 1, 2, \dots, s. \end{aligned} \quad (33)$$

Since the variables $\mathbf{G}_i$, $i = 1, 2, \dots, s$ are independent for each sample, the minimization could be further decoupled to solve each $\mathbf{G}_i$ separately:

$$\min_{\mathbf{G}_i} \|\mathbf{C}_w(:, i, :) - \mathbf{G}_i \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2\|_F^2 + \lambda \|\mathbf{G}_i\|_0. \quad (34)$$

It can be rewritten as a 1-D sparse coding problem:

$$\min_{\text{vec}(\mathbf{G}_i)} \|\text{vec}(\mathbf{C}_w(:, i, :)) - \mathbf{D} \text{vec}(\mathbf{G}_i)\|_F^2 + \lambda \|\text{vec}(\mathbf{G}_i)\|_0. \quad (35)$$

To alleviate the computational burden involved in evaluating the ℓ_0 norm, we relax it to solve its convex approximation ℓ_1 norm,

$$\min_{\text{vec}(\mathbf{G}_i)} \|\text{vec}(\mathbf{C}_w(:, i, :)) - \mathbf{D} \text{vec}(\mathbf{G}_i)\|_F^2 + \lambda \|\text{vec}(\mathbf{G}_i)\|_1. \quad (36)$$

It is the classical least absolute shrinkage and selection operator (LASSO) problem. By using a proximal gradient, its solution can be obtained using the iterative soft-thresholding algorithm (ISTA) [27].2.2) **Dictionary updating.** By fixing all core tensors $\{\mathbf{G}_i\}_{i=1}^s$, the dictionaries $\{\mathbf{D}_n\}_{n=1}^2$ can be obtained by solving

$$\begin{aligned} \min_{\{\mathbf{D}_n\}_{n=1}^2} & \sum_{i=1}^s \|\mathbf{C}_w(:, i, :) - \mathbf{G}_i \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2\|_F^2 = \min_{\{\mathbf{D}_n\}_{n=1}^2} \|\mathbf{X}_{(n)} - \mathbf{D}_n \mathbf{A}_{(n)}\|_F^2 \\ \text{s.t.} & \quad \|\mathbf{D}_n(:, r)\|_2^2 = 1, \quad n = 1, 2, \quad r = 1, 2, \dots, K_n. \end{aligned} \quad (37)$$

$$\begin{aligned} \mathbf{X}_{i(n)} &= \text{unfold}_n(\mathbf{C}_w(:, i, :)) \\ \mathbf{G}_{i(n)} &= \text{unfold}_n(\mathcal{G}_i) \\ \mathbf{X}_{(n)} &= [\mathbf{X}_{1(n)}, \mathbf{X}_{2(n)}, \dots, \mathbf{X}_{s(n)}] \\ \mathbf{A}_{i(n)} &= \mathbf{G}_{i(n)} (\mathbf{D}_N \otimes \dots \otimes \mathbf{D}_{n+1} \otimes \mathbf{D}_{n-1} \otimes \dots \otimes \mathbf{D}_1)^T, \text{ where } N = 2. \\ \mathbf{A}_{(n)} &= [\mathbf{A}_{1(n)}, \mathbf{A}_{2(n)}, \dots, \mathbf{A}_{s(n)}]. \end{aligned} \quad (38)$$

Hence, the n -th dimensional dictionary $\mathbf{D}_n$ can be resolved using its Lagrangian dual function. The solution is given by

$$\mathbf{D}_n = \mathbf{X}_{(n)} \mathbf{A}_{(n)}^T \left(\mathbf{A}_{(n)} \mathbf{A}_{(n)}^T + \mathbf{I} \right)^{-1}, \quad (39)$$

where ϵ is a small positive number.

Algorithm 2: Solve $\{\mathbf{G}_i\}_{i=1}^s$ via the MD-SSM Algorithm

Input: Training set $\{\mathbf{C}_w(:, 1, :), \mathbf{C}_w(:, 2, :), \dots, \mathbf{C}_w(:, s, :)\}$ with s samples, parameters τ and t .

Output: $\{\mathbf{G}_i\}_{i=1}^s$ and $\{\mathbf{D}_n\}_{n=1}^2$.

1: Initialize dictionaries $\{\mathbf{D}_n\}_{n=1}^2$ with the normalized data sampled in $\mathbf{X}_{(n)}$, respectively.

2: **repeat**

3: **Sparse coding step:**

4: Fix $\{\mathbf{D}_n\}_{n=1}^2$, compute $\mathbf{D} = \mathbf{D}_2 \otimes \mathbf{D}_1$

5: Update $\{\mathbf{G}_i\}_{i=1}^s$ by solving (34) and obtain $\mathbf{A}_{(n)}$

6: **Dictionary updating step:**

7: Fix $\{\mathbf{G}_i\}_{i=1}^s$, update $\{\mathbf{D}_n\}_{n=1}^2$ by solving (37)

8: **until** convergence

The complete procedure of our method is summarized in Algorithm 3.

Algorithm 3: Method

Input: Data matrix of all views $\{\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times s}\}$ ($v = 1, 2, \dots, k$), number of clusters c , regularization parameters $\lambda_1, \lambda_2, \lambda_3$ and λ_4 .

Input: Clustering labels.

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

2: Solve (13) using Algorithm 1, and obtain the optimal solution $\mathbf{C}^*$.

3: Compute view-specific similarity metrics by

$$\mathbf{C}_w(:, :, v) = (|\mathbf{C}(:, :, v)| + |\mathbf{C}^T(:, :, v)|)/2.$$

4: Collect these similarity metrics to yield a third-order similarity tensor subspace $\mathbf{C}_w^*$.

5: Solve (32) via Algorithm 2, and obtain the higher-order statistics representation $\{\mathbf{G}_i\}_{i=1}^s$.

6: Similar to the work of [28,29], construct the local affinity matrix using $\text{vec}(\{\mathbf{G}_i\}_{i=1}^s)$, and input it into spectral clustering to obtain the final clustering labels.

4. Experiments and results

In this section, we applied Exsavi to four types of real-world datasets to evaluate the clustering performance by comparison with ten recently proposed methods.

All our experiments were performed on a desktop computer with a 4.20 GHz Intel(R) Core(TM) i7 – 7700 K CPU and 16.0 GB of RAM and used MATLAB R2017a (x64).

4.1. Four types of benchmark datasets for performance evaluation

Four types of widely used benchmark datasets were selected to test the effectiveness of our method. The data statistics are summarized in Table 1. Snapshots are provided in Fig. 2.

- **UCI Digits dataset**¹: This dataset contains 2000 instances of handwritten digits (0 – 9), with 200 instances per digit. These instances are represented by five published feature sets: 240 dimensional pixel averages in 2×3 windows (Pix), 216

dimensional profile correlations (Fac), 76 dimensional Fourier coefficients of character shapes (Fou), 47 dimensional Zernike moments (Zer), and 6 dimensional morphological (Mor) features.

- **Caltech-7 dataset**²: This dataset consists of 1474 pictures of objects that belong to seven classes (i.e., Face, Motorbikes, Dollar-Bill, Garfield, Snoopy, Stop-Sign, and Windsor-Chair). All images are with six types of features: 254 dimensional CENTRIST features, 48 dimensional Gabor features, 512 dimensional GIST features, 1984 dimensional HOG features, 928 dimensional LBP features, and 40 dimensional wavelet moments.
- **BBCsport dataset**³: This dataset comprises 737 sports news articles, which were selected from the BBC Sport website from 2004 to 2005. It contains five thematic clusters (i.e., Athletics, Cricket, Football, Rugby, and Tennis). In our experiments, we use the two view and three view BBC datasets, which consist of 544 and 282 reports respectively.
- **ORL dataset**⁴: This dataset is composed of 400 faces of 40 distinct subjects. Each subject has 10 different images. For these subjects, the images were taken at different times, with varied lighting, facial expressions, and facial details. We used three types of feature sets in our experiments: 4096 dimensional raw pixel values, 3304 dimensional LBP features, and 6750 dimensional Gabor features.

4.2. Evaluation metrics

To ensure the comprehensiveness and fairness of the experiments, six widely used metrics, that is, accuracy (**ACC**), **F-score**, **Precision**, **Recall**, normalized mutual information (**NMI**), and adjusted Rand index (**ARI**), were employed to evaluate the clustering performance. Each of the metrics measures a specific property in the clustering. For these qualitative metrics, the higher the value, the better the performance.

4.3. Baseline methods

Several benchmark methods were borrowed to serve as baselines for comparison. A brief introduction of these approaches was as follows:

1. **FeaConcat** [30] concatenated the features of all views and conducts spectral clustering.
2. **Co-Reg** [31] proposed a centroid-based co-regularization method for spectral clustering.
3. **Co-Pair** [31] introduced a pairwise-based co-regularization method for spectral clustering.
4. **DIMSC** [32] utilized the Hilbert–Schmidt independence criterion to explore subspace representations.
5. **ECMSC** [33] attempted to harness complementary information by introducing a novel position-aware exclusivity term.
6. **LMSC** [34] performed data reconstruction based on the learned latent representation.
7. **HOPES** [28] fused a novel type of similarity derived from various sources.
8. **SM²SC** [35] introduced a multiplicative decomposition scheme and a variable splitting scheme to clustering task.
9. **LT-MS** [14] developed multi-view clustering via integrating the subspace representation matrices of all views as a tensor to discover the underlying structure of data.
10. **SCMV-3DT** [16] proposed a multi-view clustering by executing sparse and low-rank self-expressive tensor learning.

² <http://www.vision.caltech.edu/archive.html>.

³ <http://mlg.ucd.ie/datasets/bbc.html>.

⁴ <https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>.

¹ [http://archive.ics.uci.edu/ml/datasets/Multiple + Features](http://archive.ics.uci.edu/ml/datasets/Multiple+Features).

Table 1
Statistics on the four tested datasets.

Dataset	Type	#Instances	#Views	#Clusters
UCI digits	Digit	2000	5	10
Caltech-7	Image	1474	6	7
two-view BBCSport	Text	544	2	5
three-view BBCSport	Text	282	3	5
ORL	Face	400	3	40



Fig. 2. Samples of datasets.

4.4. Results and discussion

We report the detailed clustering results on four benchmark datasets in Tables 2–5. In each table, the tensor-based methods are highlighted in gray background. The optimal and suboptimal results are denoted by red and blue, respectively.

For a fair comparison, we run all methods for 20 trials and present the means and standard deviations of metrics. Moreover, the parameters of all compared methods are set to be the best as suggested.

On all datasets, our proposed algorithm achieves the top three best results. From Tables 2,3 and Table 5, we can easily see that our method consistently outperforms all the competitors on UCI digits, Caltech-7, and ORL datasets. On the UCI digits dataset, as the table shows, our method achieves significant improvements around 2.1%, 4.0%, 4.0%, 4.2%, 4.0%, and 4.5% over the second-best results achieved by SM²SC. On the Caltech-7 dataset, we improved around 14.4%, 10.1%, –15.8%, 27.3%, –3.1%, and 6.1% over the second-best results achieved by HOPES. As it is shown, due to the inherent characteristics of Caltech-7, the Precision value of all comparison methods is higher, while the Recall value is rel-

Table 2

Clustering Performance (Mean $\pm$ Standard Deviation) on the UCI digits Dataset. The optimal and sub-optimal are highlighted in red and blue, respectively.

Method	ACC	F-score	Precision	Recall	NMI	ARI
FeaConcat [30]	0.759 $\pm$ 0.000	0.652 $\pm$ 0.000	0.636 $\pm$ 0.000	0.670 $\pm$ 0.000	0.726 $\pm$ 0.000	0.613 $\pm$ 0.000
Co-Reg [31]	0.738 $\pm$ 0.020	0.660 $\pm$ 0.011	0.642 $\pm$ 0.016	0.680 $\pm$ 0.006	0.713 $\pm$ 0.006	0.621 $\pm$ 0.013
Co-Pair [31]	0.797 $\pm$ 0.012	0.726 $\pm$ 0.009	0.712 $\pm$ 0.013	0.741 $\pm$ 0.005	0.767 $\pm$ 0.005	0.695 $\pm$ 0.010
DiMSC [32]	0.649 $\pm$ 0.004	0.503 $\pm$ 0.001	0.496 $\pm$ 0.003	0.510 $\pm$ 0.004	0.569 $\pm$ 0.000	0.447 $\pm$ 0.003
ECMSC [33]	0.813 $\pm$ 0.000	0.812 $\pm$ 0.000	0.771 $\pm$ 0.000	0.857 $\pm$ 0.000	0.854 $\pm$ 0.000	0.790 $\pm$ 0.000
LMSC [34]	0.819 $\pm$ 0.105	0.752 $\pm$ 0.078	0.741 $\pm$ 0.089	0.764 $\pm$ 0.066	0.785 $\pm$ 0.046	0.724 $\pm$ 0.087
HOPES [28]	0.813 $\pm$ 0.034	0.765 $\pm$ 0.021	0.748 $\pm$ 0.024	0.782 $\pm$ 0.016	0.812 $\pm$ 0.010	0.738 $\pm$ 0.023
SM ² SC [35]	0.952$\pm$0.000	0.907$\pm$0.000	0.906$\pm$0.007	0.907$\pm$0.000	0.902$\pm$0.000	0.897$\pm$0.000
LT-MSC [14]	0.842 $\pm$ 0.000	0.783 $\pm$ 0.001	0.771 $\pm$ 0.001	0.795 $\pm$ 0.001	0.822 $\pm$ 0.001	0.758 $\pm$ 0.001
SCMV-3DT [16]	0.930 $\pm$ 0.000	0.861 $\pm$ 0.000	0.859 $\pm$ 0.000	0.866 $\pm$ 0.000	0.861 $\pm$ 0.000	0.846 $\pm$ 0.000
Exsavi	0.973$\pm$0.004	0.947$\pm$0.007	0.946$\pm$0.008	0.949$\pm$0.007	0.942$\pm$0.006	0.942$\pm$0.008

Table 3

Clustering Performance (Mean $\pm$ Standard Deviation) on the Caltech-7 Dataset. The optimal and sub-optimal are highlighted in red and blue, respectively.

Method	ACC	F-score	Precision	Recall	NMI	ARI
FeaConcat [30]	0.422 $\pm$ 0.000	0.417 $\pm$ 0.003	0.672 $\pm$ 0.002	0.302 $\pm$ 0.002	0.387 $\pm$ 0.001	0.234 $\pm$ 0.003
Co-Reg [31]	0.405 $\pm$ 0.012	0.440 $\pm$ 0.006	0.785 $\pm$ 0.000	0.306 $\pm$ 0.006	0.468 $\pm$ 0.008	0.286 $\pm$ 0.005
Co-Pair [31]	0.413 $\pm$ 0.001	0.433 $\pm$ 0.003	0.768 $\pm$ 0.005	0.302 $\pm$ 0.002	0.435 $\pm$ 0.006	0.276 $\pm$ 0.004
DiMSC [32]	0.450 $\pm$ 0.004	0.455 $\pm$ 0.004	0.782 $\pm$ 0.007	0.321 $\pm$ 0.006	0.425 $\pm$ 0.006	0.297 $\pm$ 0.005
ECMSC [33]	0.510 $\pm$ 0.000	0.503 $\pm$ 0.000	0.734 $\pm$ 0.000	0.383 $\pm$ 0.000	0.518 $\pm$ 0.000	0.325 $\pm$ 0.000
LMSC [34]	0.578 $\pm$ 0.028	0.567 $\pm$ 0.035	0.853 $\pm$ 0.019	0.425 $\pm$ 0.034	0.570 $\pm$ 0.026	0.418 $\pm$ 0.039
HOPES [28]	0.628$\pm$0.006	0.616$\pm$0.008	0.846 $\pm$ 0.008	0.484$\pm$0.007	0.607$\pm$0.006	0.466 $\pm$ 0.010
SM ² SC [35]	0.562 $\pm$ 0.000	0.577 $\pm$ 0.000	0.873 $\pm$ 0.007	0.431 $\pm$ 0.000	0.566 $\pm$ 0.000	0.432 $\pm$ 0.000
LT-MSC [14]	0.567 $\pm$ 0.000	0.562 $\pm$ 0.004	0.877$\pm$0.003	0.413 $\pm$ 0.003	0.591 $\pm$ 0.007	0.418 $\pm$ 0.004
SCMV-3DT [16]	0.625 $\pm$ 0.002	0.610 $\pm$ 0.002	0.889$\pm$0.010	0.464 $\pm$ 0.002	0.603$\pm$0.003	0.469$\pm$0.004
Exsavi	0.772$\pm$0.066	0.717$\pm$0.092	0.688 $\pm$ 0.094	0.757$\pm$0.121	0.576 $\pm$ 0.109	0.527$\pm$0.157

Table 4

Clustering Performance (Mean $\pm$ Standard Deviation) on the two-view BBCSport Dataset. The optimal and sub-optimal are highlighted in red and blue, respectively.

Method	ACC	F-score	Precision	Recall	NMI	ARI
FeaConcat [30]	0.669 $\pm$ 0.000	0.695 $\pm$ 0.000	0.571 $\pm$ 0.000	0.886 $\pm$ 0.000	0.758 $\pm$ 0.000	0.570 $\pm$ 0.000
Co-Reg [31]	0.583 $\pm$ 0.028	0.500 $\pm$ 0.013	0.429 $\pm$ 0.012	0.605 $\pm$ 0.020	0.416 $\pm$ 0.009	0.307 $\pm$ 0.019
Co-Pair [31]	0.620 $\pm$ 0.009	0.549 $\pm$ 0.005	0.492 $\pm$ 0.008	0.701 $\pm$ 0.024	0.489 $\pm$ 0.007	0.369 $\pm$ 0.006
DiMSC [32]	0.679 $\pm$ 0.005	0.566 $\pm$ 0.004	0.493 $\pm$ 0.009	0.666 $\pm$ 0.011	0.492 $\pm$ 0.005	0.403 $\pm$ 0.007
ECMSC [33]	0.313 $\pm$ 0.025	0.303 $\pm$ 0.022	0.243 $\pm$ 0.011	0.411 $\pm$ 0.082	0.045 $\pm$ 0.016	0.008 $\pm$ 0.020
LMSC [34]	0.944 $\pm$ 0.009	0.917 $\pm$ 0.018	0.929 $\pm$ 0.007	0.906 $\pm$ 0.029	0.855 $\pm$ 0.027	0.892 $\pm$ 0.024
HOPES [28]	0.728 $\pm$ 0.023	0.673 $\pm$ 0.017	0.556 $\pm$ 0.016	0.852 $\pm$ 0.023	0.639 $\pm$ 0.028	0.540 $\pm$ 0.024
SM ² SC [35]	0.982$\pm$0.000	0.963$\pm$0.000	0.970$\pm$0.000	0.957$\pm$0.000	0.937$\pm$0.000	0.952$\pm$0.000
LT-MSC [14]	0.717 $\pm$ 0.000	0.634 $\pm$ 0.000	0.552 $\pm$ 0.000	0.743 $\pm$ 0.000	0.557 $\pm$ 0.000	0.496 $\pm$ 0.000
SCMV-3DT [16]	0.980$\pm$0.000	0.951$\pm$0.000	0.959$\pm$0.000	0.942$\pm$0.000	0.930$\pm$0.000	0.935$\pm$0.000
Exsavi	0.967 $\pm$ 0.006	0.934 $\pm$ 0.010	0.931 $\pm$ 0.017	0.937 $\pm$ 0.009	0.898 $\pm$ 0.014	0.913 $\pm$ 0.014

Table 5

Clustering Performance (Mean $\pm$ Standard Deviation) on the ORL Dataset. The optimal and sub-optimal are highlighted in red and blue, respectively.

Method	ACC	F-score	Precision	Recall	NMI	ARI
FeaConcat [30]	0.666 $\pm$ 0.013	0.548 $\pm$ 0.016	0.524 $\pm$ 0.018	0.574 $\pm$ 0.014	0.809 $\pm$ 0.007	0.537 $\pm$ 0.016
Co-Reg [31]	0.723 $\pm$ 0.009	0.650 $\pm$ 0.010	0.602 $\pm$ 0.010	0.707 $\pm$ 0.009	0.872 $\pm$ 0.004	0.642 $\pm$ 0.010
Co-Pair [31]	0.742 $\pm$ 0.010	0.637 $\pm$ 0.011	0.589 $\pm$ 0.011	0.694 $\pm$ 0.010	0.866 $\pm$ 0.004	0.628 $\pm$ 0.011
DiMSC [32]	0.808 $\pm$ 0.034	0.754 $\pm$ 0.038	0.711 $\pm$ 0.046	0.803 $\pm$ 0.031	0.917 $\pm$ 0.013	0.748 $\pm$ 0.039
ECMSC [33]	0.729 $\pm$ 0.021	0.649 $\pm$ 0.022	0.592 $\pm$ 0.029	0.721 $\pm$ 0.017	0.879 $\pm$ 0.008	0.640 $\pm$ 0.023
LMSC [34]	0.833 $\pm$ 0.011	0.773 $\pm$ 0.038	0.721 $\pm$ 0.040	0.833 $\pm$ 0.034	0.925 $\pm$ 0.015	0.767 $\pm$ 0.039
HOPES [28]	0.776 $\pm$ 0.028	0.703 $\pm$ 0.028	0.634 $\pm$ 0.044	0.792 $\pm$ 0.013	0.897 $\pm$ 0.008	0.696 $\pm$ 0.029
SM ² SC [35]	0.845$\pm$0.024	0.802$\pm$0.031	0.768$\pm$0.032	0.839$\pm$0.032	0.932$\pm$0.000	0.797$\pm$0.000
LT-MSC [14]	0.803 $\pm$ 0.028	0.750 $\pm$ 0.030	0.703 $\pm$ 0.035	0.803 $\pm$ 0.024	0.913 $\pm$ 0.011	0.743 $\pm$ 0.031
SCMV-3DT [16]	0.795 $\pm$ 0.028	0.744 $\pm$ 0.030	0.694 $\pm$ 0.040	0.804 $\pm$ 0.019	0.909 $\pm$ 0.010	0.738 $\pm$ 0.031
Exsavi	0.875$\pm$0.013	0.828$\pm$0.017	0.802$\pm$0.020	0.857$\pm$0.015	0.942$\pm$0.006	0.824$\pm$0.017

atively much lower. Exsavi significantly improves the clustering results of Caltech-7 from multiple evaluation terms, i.e., ACC, F-score, Recall, NMI, and ARI. To be excited, the improvement of our method to the Recall result is noteworthy in all comparison methods. In the meantime, the Precision of Exsavi has declined relative to other metrics. The reason is that Recall and Precision usually have opposing influences. When considering such multi-view datasets, the performance of Precision will be affected by the rapidly rising Recall index. On the ORL dataset, we improved around 3.0%, 2.6%, 3.4%, 1.8%, 1.0%, and 2.7% over the second-best results achieved by SM²SC.

For the two-view BBCSport dataset, our method achieves the third-best performance over all baseline methods. As the Table 4 indicates, our results are close to the most competitive methods SM²SC and SCMV-3DT in terms of all evaluation benchmarks.

To be excited, the improvement of our method to the result is significant in most cases, as illustrated by Tables 2–5. Combined with the statistics in Table 1, it can be seen that our method is most effective when the number of views is greater than 2. The reason is that the merit of Exsavi is the ability to learn view-wise higher-order statistics. The more numbers of views in data, the more efficient to explore the view-wise relationship. For two-view BBCSport, it is the number of views affects the exertion of this advantage.

Next, we have re-conducted an experiment on three-view BBCSport further to demonstrate the performances on the sparse coding term. It consists of 282 reports with three views of feature sets. We report the clustering results on three-view BBCSport in Table 6. The experimental results show that the SCMV-3DT has good performance yet is inferior to Exsavi occasionally. The merits

Table 6

Clustering Performance (Mean $\pm$ Standard Deviation) on three-view BBCSport. The optimal and sub-optimal are highlighted in red and blue, respectively.

Method	ACC	F-score	Precision	Recall	NMI	ARI
FeaConcat [30]	0.809$\pm$0.000	0.832$\pm$0.000	0.832$\pm$0.000	0.832$\pm$0.000	0.758$\pm$0.000	0.774$\pm$0.000
Co-Reg [31]	0.497 $\pm$ 0.017	0.415 $\pm$ 0.014	0.351 $\pm$ 0.006	0.510 $\pm$ 0.031	0.250 $\pm$ 0.002	0.160 $\pm$ 0.014
Co-Pair [31]	0.491 $\pm$ 0.010	0.404 $\pm$ 0.005	0.331 $\pm$ 0.002	0.524 $\pm$ 0.014	0.235 $\pm$ 0.002	0.131 $\pm$ 0.004
DiMSC [32]	0.733 $\pm$ 0.004	0.667 $\pm$ 0.007	0.725 $\pm$ 0.008	0.618 $\pm$ 0.006	0.639 $\pm$ 0.005	0.564 $\pm$ 0.009
ECMSC [33]	0.596 $\pm$ 0.084	0.549 $\pm$ 0.058	0.527 $\pm$ 0.107	0.589 $\pm$ 0.050	0.471 $\pm$ 0.077	0.377 $\pm$ 0.101
LMSC [34]	0.745 $\pm$ 0.010	0.652 $\pm$ 0.002	0.708 $\pm$ 0.035	0.604 $\pm$ 0.031	0.557 $\pm$ 0.015	0.544 $\pm$ 0.003
HOPES [28]	0.767 $\pm$ 0.065	0.654 $\pm$ 0.060	0.572 $\pm$ 0.076	0.768 $\pm$ 0.029	0.584 $\pm$ 0.065	0.506 $\pm$ 0.098
SM ² SC [35]	0.833$\pm$0.000	0.827$\pm$0.000	0.849$\pm$0.000	0.805$\pm$0.000	0.790$\pm$0.000	0.769$\pm$0.000
LT-MSC [14]	0.715 $\pm$ 0.008	0.632 $\pm$ 0.002	0.631 $\pm$ 0.012	0.633 $\pm$ 0.017	0.592 $\pm$ 0.004	0.504 $\pm$ 0.001
SCMV-3DT [16]	0.706 $\pm$ 0.008	0.606 $\pm$ 0.012	0.638 $\pm$ 0.011	0.576 $\pm$ 0.013	0.551 $\pm$ 0.013	0.478 $\pm$ 0.015
Exsavi	0.790 $\pm$ 0.027	0.745 $\pm$ 0.032	0.699 $\pm$ 0.050	0.800 $\pm$ 0.025	0.681 $\pm$ 0.029	0.648 $\pm$ 0.048

of SCMV-3DT lie in its low-rank tensor construction strategy so that it can make good use of information of all views. However, compared with Exsavi, the latent subspace structure within data is captured directly from self-expressive tensor other than by simultaneously utilizing view-wise statistics from multi-view subspaces. Through employing the sparse coding term in Exsavi, the view-wise relationship is exploited and evacuated. Therefore, the latent clustering structure by Exsavi is more robust than SCMV-3DT. Fig. 8 displays the visualizations of latent representations on three-view BBCSport with t-SNE. The first image shows the raw sampling space, and the second and third images show the feature space learned by SCMV-3DT and Exsavi respectively. From Fig. 8, we observe that Exsavi captures more discriminative features. Combined with above results, one can find the performance

of the sparse coding term in Exsavi is stable. For example, except for two-view BBCSport, Exsavi outperforms SCMV-3DT on all benchmark datasets. The reason is that Exsavi can learn the view-wise statistics via the sparse coding. The more views there are, the sufficient view-wise relationship can be mined. For BBCSport, the experimental results show that it is not because sparse coding term is invalid for such dataset, but the number of views affects the extraction of view-wise relationship.

Fig. 3 shows the similarity matrices of our method and SCMV-3DT on UCI digits, Caltech-7, two-view BBCSport, three-view BBCSport, and ORL, respectively. One could clearly see that our approach can reveal the underlying diagonal-block structures very well, which brings benefits to the subsequent clustering task and further validates the advantages of our method.

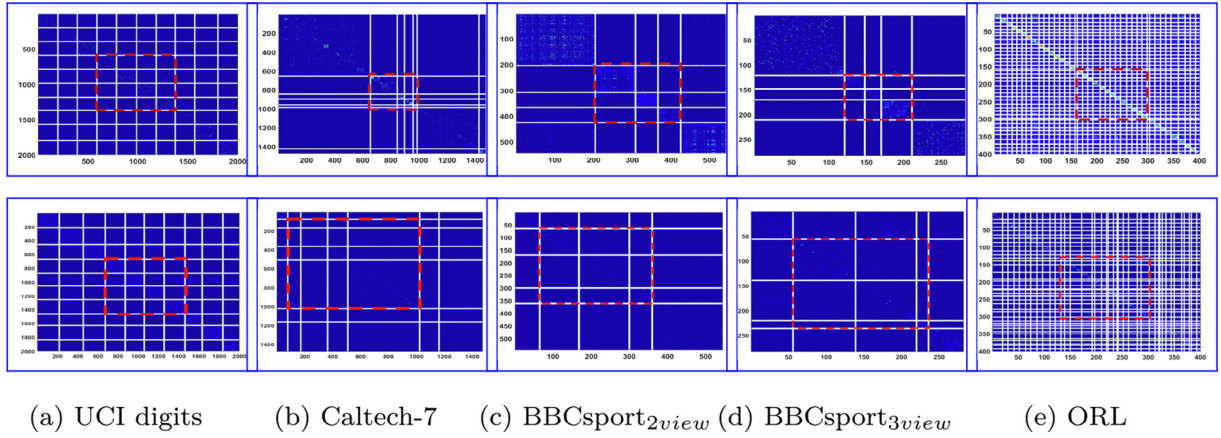


Fig. 3. Visualization of affinity matrices by proposed method and SCMV-3DT on (a) UCI digits; (b) Caltech-7; (c) two-view BBCSport; (d) three-view BBCSport; (e) ORL. The upper and lower row represent the Exsavi and the SCMV-3DT, respectively. Our method yields highly informative in extracting the view-dependent characteristics, and thus the similarity matrices are visualized distinctly with a clearer diagonal-block structure. The regions bounded by the red dashed boxes indicate the difference between SCMV-3DT and our method.

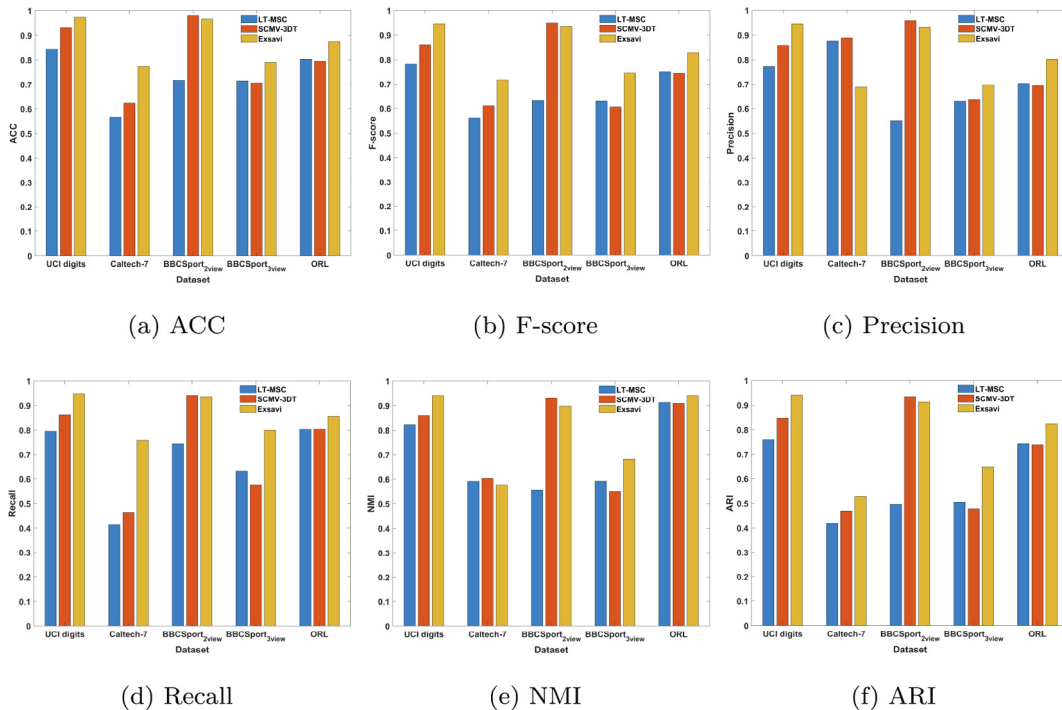


Fig. 4. Clustering results of three tensor-based methods on UCI digits, Caltech-7, two-view BBCSport, three-view BBCSport, and ORL, respectively.

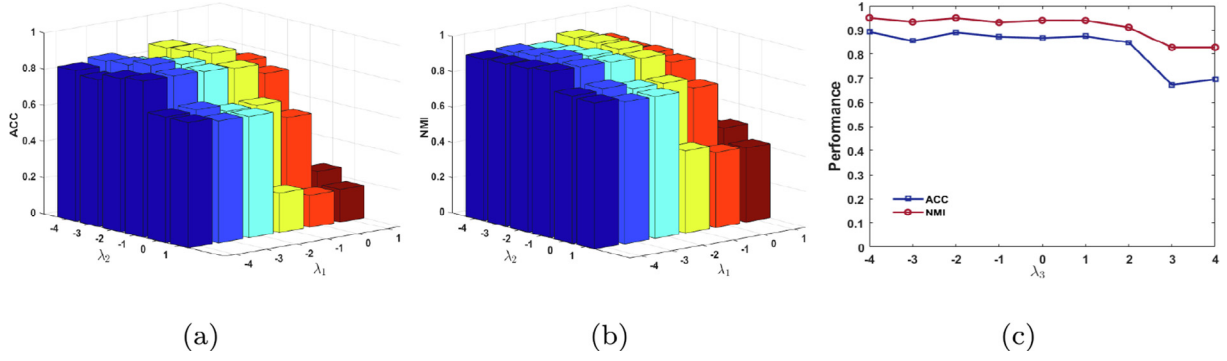


Fig. 5. Clustering performance on the ORL dataset with different parameter settings ($\log_{10}$ value). (a) ACC and (b) NMI varying λ_1 and λ_2 with $\lambda_3 = 1.1$. (c) Performance varying with λ_3 fixing $\lambda_1 = 10^{-3}$ and $\lambda_2 = 10^{-1}$.

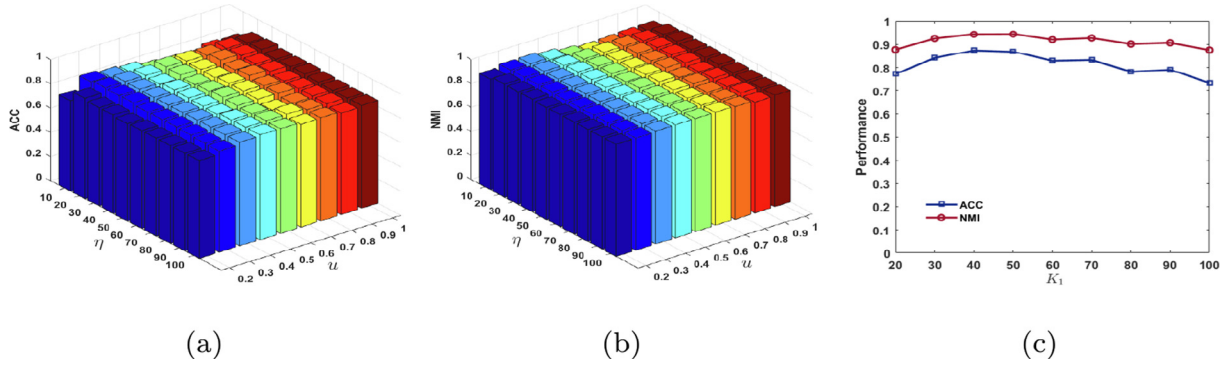


Fig. 6. Clustering performance on the ORL dataset with different parameter settings. (a) ACC and (b) NMI varying η and u with $K_1 = 40$. (c) Performance varying with K_1 fixing $\eta = 30$ and $u = 0.8$.

Table 7
Parameter settings.

Dataset	K_1	K_2	η	u
UCI digits	50	2	60	0.29
Caltech-7	40	2	148	0.21
Two-view BBCSport	8	1	80	0.5
Three-view BBCSport	6	2	20	0.3
ORL	40	2	30	0.8

Fig. 4 gives the visual clustering results on different benchmark metrics. As it is shown, the proposed method outperforms other methods on most datasets, except for two-view BBCSport. It is worth to flag the fact that our method is better than the most recently published methods [30–34,28,14] on the two-view BBCSport's experiment.

The superior performance of Exsavi is lying in its ability to extract both intrinsic sample-wise and view-wise statistics from

multi-view data, which results in effective improvements of clustering on benchmark datasets. Directly learning low-rank tensor like SCMV-3DT ignores mining higher-order attributes in its similarity statistics. If removing the view-wise higher-order statistics, our method will degenerate into SCMV-3DT. The view-wise statistics brings benefits to the improvement of clustering performance, which is confirmed by our experimental results.

4.5. Parameter sensitivity

In our model, there are four regularization parameters balancing the effect of sparse term, low-rank term, consensus term, and higher-order statistics term. We empirically set the parameters, that is, $\lambda_1 = 0.001$, $\lambda_2 = 0.1$, $\lambda_3 = 1.1$, and $\lambda_4 = 0$ in the tensor self-representation learning. To conserve space, Fig. 5 shows an example of parameter testing on ORL.

In the multi-dimensional sparse representation, there are two parameters (i.e., K_1, K_2) balancing the size of the core tensor. In

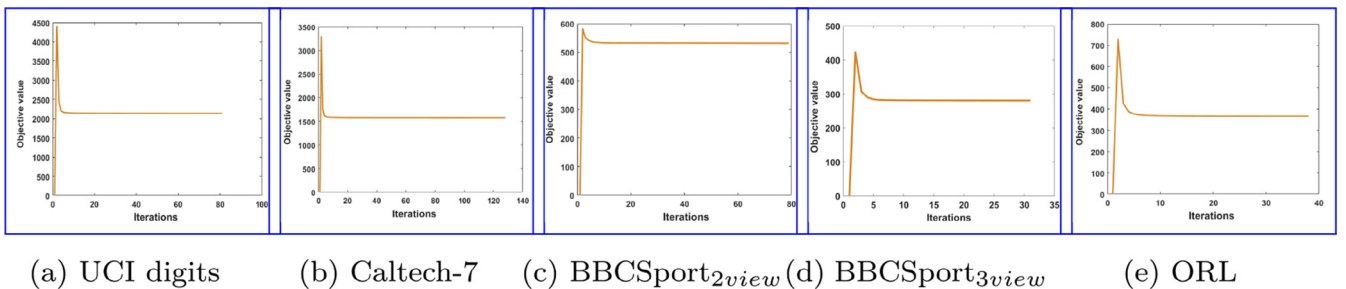


Fig. 7. Convergence curve of the proposed method. (a) UCI digits; (b) Caltech-7; (c) two-view BBCSport; (d) three-view BBCSport; (e) ORL.

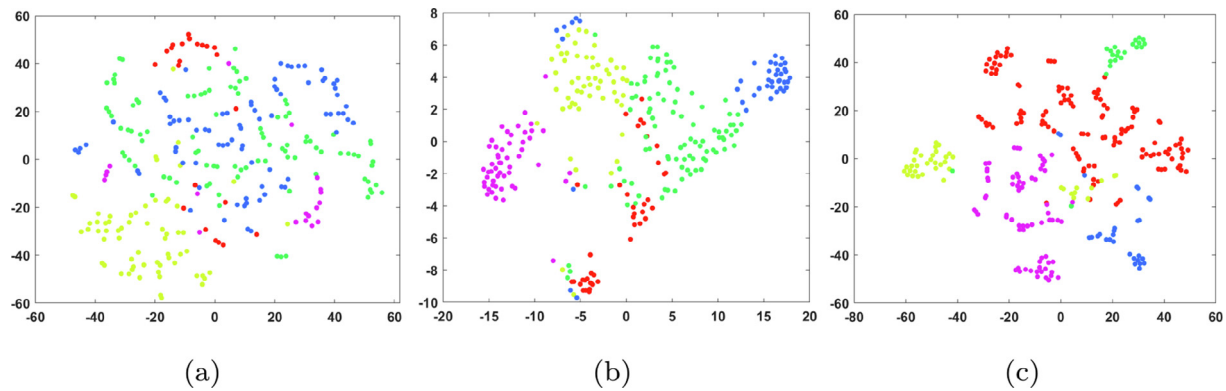


Fig. 8. Visualization of learned features on three-view BBCSport with t-SNE. The samples are visualized with respect to (a) the raw sampling space; (b) the learned feature subspace using SCMV-3DT; (c) the learned feature subspace using Exsavi.

the spectral clustering phase, there are two parameters (i.e., η, u) involved in computing an affinity matrix containing the nearest neighbor, where η is the number of nearest neighbors and u is a hyperparameter that can be empirically set. In the following, we study the influence of parameters K_1, η , and u on ORL in terms of ACC and NMI by setting them to different values, e.g., $[10, 20, \dots, 100]$. We vary each parameter individually while keeping the others fixed. Fig. 6 shows the ACC and NMI on ORL for changing the value of different parameters. From Fig. 6, we can see that our method is relatively insensitive to its parameters when the settings are provided in a suitable range. Thus, we choose $K_1 = 40, \eta = 30$, and $u = 0.8$ in our experiments.

Table 7 summarizes the parameters in our experiments.

4.6. Convergence analysis

To demonstrate the convergence of the proposed method, we draw the curves of the objective function relative to the iteration in the stage of multi-dimensional sparse coding in Fig. 7,8. On all datasets, the algorithm reaches a stable status within 20 iterations.

5. Conclusion

In this paper, we proposed a novel method for clustering multi-view data. The data is represented in tensor form and mapped into a latent tensor subspace to exploit its sample-wise relationship. By taking advantage of circular convolution of t -product and its shifting invariance structure, the spatial correlation within samples is largely preserved through subspace learning. As such, the self-expressive tensor will first be reconstructed, where its frontal slices stand for view-specific self-expressive matrices. Then, higher-order statistics in its similarity tensor are further analyzed to dig the view-wise relationship. Our philosophy is to learn view-wise higher-order statistics. Via multi-dimensional sparse coding all lateral slices of similarity tensor, it can extract a sparse core tensor and two factor matrices (i.e., dictionaries in each dimension). The core tensor is the centralized higher-order representation in similarity relationship, which can essentially capture the objective clustering structure. Each core tensor can be taken as the new representation of multi-view data. The core tensor is task-oriented and highly informative in extracting the view-dependent characteristics.

The two relationships are jointly learned, and thus the novel and essential representation hidden in the multi-view data can be further learned. We applied our method to four types of real-world datasets. Extensive experimental results demonstrated its superior performances in clustering.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (61472145, 61771007), Science and Technology Planning Project of Guangdong Province (2016A010101013, 2017B020226004), Applied Science and Technology Research and Development Project of Guangdong Province (2016B010127003), Guangdong Natural Science Foundation (2017A030312008), Fundamental Research Fund for the Central Universities (2017ZD051), and Health & Medical Collaborative Innovation Project of Guangzhou City (201803010021).

References

- [1] M. Cheng, L. Jing, M.K. Ng, Tensor-based low-dimensional representation learning for multi-view clustering, *IEEE Trans. Image Process.* 28 (5) (2018) 2399–2414.
- [2] S. Kanaan-Izquierdo, A. Ziyatdinov, A. Perera-Lluna, Multiview and multifeature spectral clustering using common eigenvectors, *Pattern Recogn. Lett.* 102 (2018) 30–36.
- [3] L. Houthuys, R. Langone, J.A. Suykens, Multi-view kernel spectral clustering, *Inf. Fusion* 44 (2018) 46–56.
- [4] C. Lu, S. Yan, Z. Lin, Convex sparse spectral clustering: Single-view to multi-view, *IEEE Trans. Image Process.* 25 (6) (2016) 2833–2843.
- [5] W. Weng, W. Zhou, J. Chen, H. Peng, H. Cai, Enhancing multi-view clustering through common subspace integration by considering both global similarities and local structures, *Neurocomputing* 378 (2020) 375–386.
- [6] D. Huang, C.-D. Wang, J. Wu, J.-H. Lai, C.K. Kwok, Ultra-scalable spectral clustering and ensemble clustering, *IEEE Trans. Knowl. Data Eng.* 32 (6) (2019) 1212–1226.
- [7] L. Huang, H.-Y. Chao, C.-D. Wang, Multi-view intact space clustering, *Pattern Recogn.* 86 (2019) 344–353.
- [8] Q. Zou, G. Lin, X. Jiang, X. Liu, X. Zeng, Sequence clustering in bioinformatics: an empirical study, *Brief. Bioinf.* 21 (1) (2020) 1–10.
- [9] Q. Huang, Y. Zhang, H. Peng, T. Dan, W. Weng, H. Cai, Deep subspace clustering to achieve jointly latent feature extraction and discriminative learning, *Neurocomputing* 1 (1) (2020) 1, <https://doi.org/10.1016/j.neucom.2020.04.120>.
- [10] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (11) (2013) 2765–2781.
- [11] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (1) (2012) 171–184.
- [12] J. Liu, C. Wang, J. Gao, J. Han, Multi-view clustering via joint nonnegative matrix factorization, in: *Proceedings of the 2013 SIAM International Conference on Data Mining*, SIAM, 2013, pp. 252–260.
- [13] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, in: *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

- [14] C. Zhang, H. Fu, S. Liu, G. Liu, X. Cao, Low-rank tensor constrained multiview subspace clustering, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1582–1590.
- [15] M. Yin, J. Gao, S. Xie, Y. Guo, Low-rank multi-view clustering in third-order tensor space, *arXiv preprint arXiv:1608.08336*.
- [16] M. Yin, J. Gao, S. Xie, Y. Guo, Multiview subspace clustering via tensorial t-product representation, *IEEE Trans. Neural Networks Learn. Syst.* (99) (2018) 1–14.
- [17] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, Y. Qu, On unifying multi-view self-representations for clustering by tensor multi-rank minimization, *Int. J. Comput. Vis.* 126 (11) (2018) 1157–1179.
- [18] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (3) (2009) 455–500.
- [19] M.E. Kilmer, K. Braman, N. Hao, R.C. Hoover, Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging, *SIAM J. Matrix Anal. Appl.* 34 (1) (2013) 148–172.
- [20] E. Kernfeld, S. Aeron, M. Kilmer, Clustering multi-way data: A novel algebraic approach, *arXiv preprint arXiv:1412.7056*.
- [21] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T.S. Huang, S. Yan, Sparse representation for computer vision and pattern recognition, *Proc. IEEE* 98 (6) (2010) 1031–1044.
- [22] N. Qi, Y. Shi, X. Sun, J. Wang, B. Yin, J. Gao, Multi-dimensional sparse models, *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (1) (2017) 163–178.
- [23] H. Yin, Tensor sparse representation for 3-d medical image fusion using weighted average rule, *IEEE Trans. Biomed. Eng.* 65 (11) (2018) 2622–2633.
- [24] Z. Wen, D. Goldfarb, W. Yin, Alternating direction augmented lagrangian methods for semidefinite programming, *Math. Program. Comput.* 2 (3–4) (2010) 203–230.
- [25] G.B. Arfken, H.J. Weber, *Mathematical methods for physicists*, 1999.
- [26] J. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM J. Optim.* 20 (4) (2010) 1956–1982.
- [27] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imag. Sci.* 2 (1) (2009) 183–202.
- [28] A. Xu, J. Chen, H. Peng, G. Han, H. Cai, Simultaneous interrogation of cancer omics to identify subtypes with significant clinical differences, *Front. Genet.* 10 (2019) 236.
- [29] B. Wang, A.M. Mezlini, F. Demir, M. Fiume, Z. Tu, M. Brudno, B. Haibe-Kains, A. Goldenberg, Similarity network fusion for aggregating data types on a genomic scale, *Nat. Methods* 11 (3) (2014) 333.
- [30] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: *Advances in Neural Information Processing Systems*, 2002, pp. 849–856.
- [31] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: *Advances in Neural Information Processing Systems*, 2011, pp. 1413–1421.
- [32] X. Cao, C. Zhang, H. Fu, S. Liu, H. Zhang, Diversity-induced multi-view subspace clustering, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 586–594.
- [33] X. Wang, X. Guo, Z. Lei, C. Zhang, S.Z. Li, Exclusivity-consistency regularized multi-view subspace clustering, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 923–931.
- [34] C. Zhang, Q. Hu, H. Fu, P. Zhu, X. Cao, Latent multi-view subspace clustering, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4279–4287.
- [35] Z. Yang, Q. Xu, W. Zhang, X. Cao, Q. Huang, Split multiplicative multi-view subspace clustering, *IEEE Trans. Image Process.* 28 (10) (2019) 5147–5160.



Guoqiang Han received the B.S. degree from Zhejiang University, Hangzhou, China, in 1982 and the M.S. and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 1985 and 1988, respectively. He is a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. He has published over 100 research papers. His current research interests include multimedia, computational intelligence, machine learning, and computer graphics.



Bin Zhang is currently working toward the Ph.D. degree in School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He received the B.S. and M.S. degrees in School of Information Science and Engineering, from Shandong Normal University, Jinan, China. His research interests include multimedia, machine learning, and image processing.



Guihua Tao is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He received the B.S. degree in Network Engineering from Guangdong University of Technology, Guangzhou, China. His current research interests include deep learning and image processing.



Hongmin Cai is a Professor at the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He received the B.S. and M.S. degrees in mathematics from the Harbin Institute of Technology, Harbin, China, in 2001 and 2003, respectively, and the Ph.D. degree in applied mathematics from Hong Kong University in 2007. His areas of research interests include biomedical image processing, machine learning, and omics data integration.



Haiyan Wang is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. She received the B.S. degree in Computer Science and Technology from Huaibei Coal Industry Teachers College, Huaibei, China, and M.S. degree in Computer Science and Technology from Anhui University, Hefei, China. Her current research interests include machine learning and pattern recognition.