arXiv:1601.00149v2 [cs.CV] 7 Jan 2016

A Submodule Clustering Method for Multi-way Data by Sparse and Low-Rank Representation

Xinglin Piao[1], Yongli Hu[1], Junbin Gao[2], Yanfeng Sun[1] and Zhouchen Lin[3]
[1]College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China
[2]Business Analytics Discipline. The University of Sydney Business School, Camperdown NSW 2006, Australia
[3]Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, Beijing 100871, China
`piaoxinglin@126.com; {huyongli,yfsun}@bjut.edu.cn; junbin.gao@sydney.edu.au; zlin@pku.edu.cn`

Abstract

A new submodule clustering method via sparse and low-rank representation for multi-way data is proposed in this paper. Instead of reshaping multi-way data into vectors, this method maintains their natural orders to preserve data intrinsic structures, e.g., image data kept as matrices. To implement clustering, the multi-way data, viewed as tensors, are represented by the proposed tensor sparse and low-rank model to obtain its submodule representation, called a free module, which is finally used for spectral clustering. The proposed method extends the conventional subspace clustering method based on sparse and low-rank representation to multi-way data submodule clustering by combining t-product operator. The new method is tested on several public datasets, including synthetical data, video sequences and toy images. The experiments show that the new method outperforms the state-of-the-art methods, such as Sparse Subspace Clustering (SSC), Low-Rank Representation (LRR), Ordered Subspace Clustering (OSC), Robust Latent Low Rank Representation (RobustLatLRR) and Sparse Submodule Clustering method (SSmC).

1. Introduction

The clustering or segmentation problem of multi-way data, particularly images and videos, has attracted great interest in computer vision, pattern recognition and signal processing [3, 2, 1]. In the traditional setting, data samples are assumed to be embedded in linear spaces and are in high dimensional and multi-dimensional (a.k.a. multi-way) format demanding huge amount of computation time and memory to conduct data analysis. Fortunately, it has been demonstrated that high dimensional and multi-way data often lie in lower dimensional subspaces and have intrinsic subspace structures [4]. From this observation, one can assume that data are drawn from multiple subspaces and each datum in a subspace could be linearly represented by a smaller number of data samples from the same subspace. Thus the main goal of subspace clustering is to group data into different clusters such that data in each cluster just come from one particular subspace. Based on the aforementioned subspace hypothesis, researchers proposed many subspace clustering methods. The most representatives are Sparse Subspace Clustering (SSC) [5], Ordered Subspace Clustering (OSC) [6], Low-Rank Representation (LRR) [7] and Robust Latent Low Rank Representation (RobustLatLRR) [8] . In these methods, an affinity matrix is firstly learned from sample data and the clustering results are obtained by a clustering algorithm such as K-means or Normalized Cuts (NCut) [9].

To deal with high dimensional and multi-way data, the classic and straight way is to vectorize them as vectors which are then fed to a learning algorithm designed for vectorial data. However this vectorization procedure certainly destroys any intrinsic structure such as spatial information contained in data. For example, if we reshape or map a 2D image in size of $H \times D$ into a vector of length HD, the correlation of the local area of pixels and other inherent properties such as texture will be destroyed. This leads to lose some useful intrinsic information in following-up applications, such as clustering and recognition. A better strategy to deal with multi-way data is to maintain their intrinsic structure. Many newly proposed methods take advantage of matrix structure by adopting dimensionality reduction approach or finding the best subspace to approximate the matrix data, and achieve successful performance [11, 10].

Observing that the most typical clustering methods use the vector form to represent the data and ignore the multi-way data intrinsic structure, we concentrate on exploiting the tensor representation of multi-way data, especially the 2D images, and propose a new image submodule clustering method. In this method, different from the traditional clustering methods which map the matrix data to vector data, the $H \times D$ matrix of image data are kept and modeled as a tensor form by twisting it into a third order tensor of size $H \times 1 \times D$. Particularly, in video segmentation applications, samples of video frames are sometimes arbitrarily shifted or in some cases rotated due to camera movement or changes in the pose. Images with this kind of dynamic variation would generate a submodule [13, 12]. However, the traditional scalar product is always adopted to represent the linear combination of samples in subspace but lacks of ability to describe the shifted copies in submodule. Therefore, we adopt the t-product [14] based on the circular convolution to describe the dynamic character in consecutive images sequences from a submodule. By the t-product, the image data samples will be grouped into a third-order tensor and represented by the proposed tensor low-rank model in terms of the union of free submodules, in which the new affinity information will be used for the final clustering. By the t-product and factorization operations of tensor, we successfully extend the traditional LRR [7] clustering method to the case of multi-way data. The proposed method has been evaluated on both synthetic data and real-world data and outperforms state-of-the-art methods.

The paper is organized as follows. We introduce the notations and preliminaries about t-product in Section 2 and review the related works in Section 3. Section 4 presents the proposed multi-way data clustering method and Section 5 is dedicated to solving the optimization problem. Section 6 assesses the performance of the proposed method on several databases against several state-of-the-art methods. Finally, conclusions are discussed in Section 7.

2. Notations and Preliminaries

In this section, we will introduce some notations and basic definitions for tensors, and the preliminaries of linear algebra for tensor with t-product used in the proposed method.

2.1. Notations and basic definitions for tensor

We use calligraphy letters for tensors, e.g. $\mathcal{Y}$, bold lowercase letters for vectors, e.g. $\mathbf{y}$, bold uppercase for matrices, e.g. $\mathbf{Y}$, lowercase letters for entries, e.g. y, uppercase letters for dimension numbers, e.g. H, D, L, N. For convenience, we adopt Matlab notation to denote the elements in tensors. We use $\mathcal{Y}(:,:,i)$, $\mathcal{Y}(:,i,:)$ and $\mathcal{Y}(i,:,:)$ to represent the i-th frontal, lateral and horizontal slice, respectively. $\mathcal{Y}(:,i,j)$, $\mathcal{Y}(i,:,j)$ and $\mathcal{Y}(i,j,:)$ represent the mode-1, mode-2 and mode-3 fiber, respectively. We use $\hat{\mathcal{Y}}$=fft$(\mathcal{Y},[\ ],3)$ to denote the Discrete Fourier Transform (DFT) along the third dimension for third order tensor $\mathcal{Y}$. Also we use $\mathbf{Y}^{(i)}$ and $\hat{\mathbf{Y}}^{(i)}$ to denote the i-th frontal slice of $\mathcal{Y}$ and $\hat{\mathcal{Y}}$, respectively, and $\vec{\mathbf{Y}}^{(i)}$ as the i-th lateral slice of tensor $\mathcal{Y}$.

It is necessary to introduce five block-based operators, i.e., bcirc, bvec, bvfold, bdiag and bdfold [15]. For a tensor $\mathcal{Y} \in \Re^{n_1 \times n_2 \times n_3}$, the $\mathbf{Y}^{(i)}$ could be used to form the block circulant matrix as below:

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

The block vectorizing and its opposite operation are

$$\text{bvec}(\mathcal{Y}) = \begin{bmatrix} \mathbf{Y}^{(1)} \\ \mathbf{Y}^{(2)} \\ \vdots \\ \mathbf{Y}^{(n_3)} \end{bmatrix}, \quad \text{bvfold}(\text{bvec}(\mathcal{Y})) = \mathcal{Y}. \quad (2)$$

And the block diagonal matrix and its opposite operation are

$$\text{bdiag}(\mathcal{Y}) = \begin{bmatrix} \mathbf{Y}^{(1)} & & & \\ & \mathbf{Y}^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{Y}^{(n_3)} \end{bmatrix}, \quad (3)$$
$$\text{bdfold}(\text{bdiag}(\mathcal{Y})) = \mathcal{Y}.$$

To develop our method, we also need some norm definitions for tensor as below:

Definition 1. *The Frobenius norm of a tensor* $\mathcal{Y}$ *is* $\|\mathcal{Y}\|_F = (\sum_{i,j,k} \mathcal{Y}(i,j,k)^2)^{\frac{1}{2}}$.

Definition 2. *The F1 norm of a tensor* $\mathcal{Y}$ *is* $\|\mathcal{Y}\|_{F1} = \sum_{i,j} \|\mathcal{Y}(i,j,:)\|_F$.

Definition 3. *The FF1 norm of a tensor* $\mathcal{Y}$ *is* $\|\mathcal{Y}\|_{FF1} = \sum_i \|\mathcal{Y}(i,:,:)\|_F$.

Definition 4. *The F-Nuclear norm of a tensor* $\mathcal{Y}$ *is* $\|\mathcal{Y}\|_{*f} = \sum_i \|\mathcal{Y}(:,:,i)\|_*$.

Definition 5. *The F-Infinite norm of a tensor* $\mathcal{Y}$ *is* $\|\mathcal{Y}\|_\infty = max_i\{\|\mathcal{Y}(:,:,i)\|_\infty\}$.

In the above definitions, $\|\cdot\|_F$, $\|\cdot\|_*$ and $\|\cdot\|_\infty$ are matrix Frobenius norm, nuclear norm and maximal infinity norm, respectively.

2.2. Linear algebra for tensor with t-product

As described above, we orient an $H \times D$ image data $\mathbf{Y}$ by twisting it into the page which will change each image data as a $H \times 1 \times D$ third order tensor as shown in Figure 1 instead of vectorizing from typical clustering methods.

Therefore, an $H \times D$ matrix image data has been changed as a vector with length of H where each element is a $1 \times 1 \times D$ tube fiber and N image samples will be organized as a $H \times N \times D$ tensor $\mathcal{Y}$. We denote $\mathbb{K}_D$ as the set

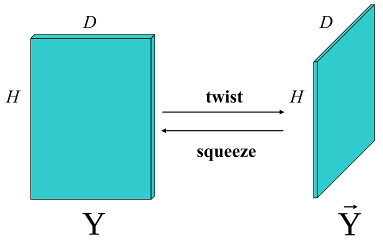

Figure 1. The twist and squeeze operations for matrix.

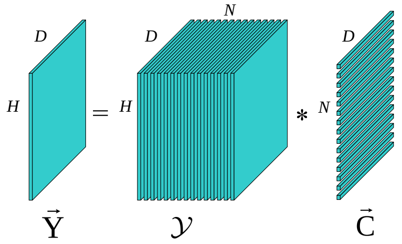

Figure 2. An illustration of t-linear combination.

of tube fibers with the size of $1 \times 1 \times D$ and $\mathbb{K}_D^H$ as the set of oriented matrices of size $H \times 1 \times D$. The first task is to find a method to multiply two tube fibers which means to "linearly" combine oriented matrices where the weights are tube fibers and not scalars. In the spirit of [16], *the set of oriented matrices could be regarded as a module over the ring* $\mathbb{K}_D$. Therefore, we adopted the method of t-product proposed in [14] to implement the fibers multiplication. It has been proved that the t-product is a useful generalization of matrix multiplication for tensors [17]. It is generally defined by unfolding tensors into block circulant matrices, multiplying the matrices, and folding the result back up into a three-dimensional array. For 3-way tensors, the t-product is defined as [15]:

Definition 6. *Let* $\mathcal{X} \in \Re^{n_1 \times n_2 \times n_3}$ *and* $\mathcal{Y} \in \Re^{n_2 \times n_4 \times n_3}$, *then the t-product* $\mathcal{X} * \mathcal{Y}$ *is* $\mathcal{M} \in \Re^{n_1 \times n_4 \times n_3}$ *as follows:*

$$\mathcal{M} = \mathcal{X} * \mathcal{Y} = \mathrm{bvfold}(\mathrm{bcirc}(\mathcal{X})\mathrm{bvec}(\mathcal{Y})). \tag{4}$$

where $*$ *denotes the circular convolution.*

The t-product is analogous to the matrix multiplication except that the circular convolution replaces the multiplication operation between the elements, which are now mode-3 fibers as follows:

$$\mathcal{M}(i,j,:) = \sum_{k=1}^{n_2} \mathcal{X}(i,k,:) * \mathcal{Y}(k,j,:). \tag{5}$$

The t-product could be efficiently computed by using the FFT [14]. Therefore, the t-product in the original domain corresponds to the matrix multiplication of the frontal slices in the Fourier domain as follows:

$$\hat{\mathbf{M}}^{(i)} = \hat{\mathbf{X}}^{(i)}\hat{\mathbf{Y}}^{(i)}, \quad \mathcal{M} = \mathrm{ifft}(\hat{\mathcal{M}}, [\,], 3). \tag{6}$$

where $\mathrm{ifft}(\hat{\mathcal{M}}, [\,], 3)$ denotes the inverse Fourier Transform along the third dimension of $\hat{\mathcal{M}}$. We denote $(\mathbb{K}_D^H, *)$ as the $\mathbb{K}_D^H$ equipped with the t-product $*$. Although $(\mathbb{K}_D^H, *)$ does not form a field because there are non-zero tubes which are not invertible, it does form what is referred to as a ring with unity [14]. A module over a ring could be regarded as a generalization of the concept of a vector space over a field, where the corresponding "scalars" are the elements of the ring [12]. Therefore, the t-product is an appropriate operator to model the orient matrices and have the "tensor linear (t-linear) combination" for free submodule. The t-linear combination means a sum of oriented matrices form $\mathbb{K}_H^D$ multiplied by coefficients from $\mathbb{K}_D$ shown in Figure 2.

As shown in Figure 2, $\vec{\mathbf{Y}}$ represents an oriented matrix image with the size of $H \times 1 \times D$. $\mathcal{Y}$ represents the set of N oriented matrix image samples with the size of $H \times N \times D$. $\vec{\mathbf{C}}$ represents the t-linear combination matrix with the size of $N \times 1 \times D$ and each $1 \times 1 \times D$ tube fiber in $\vec{\mathbf{C}}$ represents the coefficient for oriented matrix image sample $\vec{\mathbf{Y}}$ with the t-product.

3. Related Work

There is little existing prior work on the submodule clustering of multi-way data, particulary 2D images. Therefore, we will provide an overview of the recent developments in subspace clustering which is closely related to the submodule clustering. Consider a 2D image set $\mathbb{Y} = \{\mathbf{Y}_i\}_{i=1}^N$, where $\mathbf{Y}_i \in \Re^{H \times D}$ and N represents the number of image samples. The common preprocessing approach of typical subspace clustering methods is mapping each 2D image sample $\mathbf{Y}_i$ to 1-D vector $\mathbf{y}_i$ and form all samples as a matrix, i.e. $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N] \in \Re^{L \times N}$, where $\mathbf{y}_i \in \Re^L$, $i = 1, 2, ..., N$ and $L = HD$. It is assumed that all these vectors are drawn from a union of K subspaces $\{\mathcal{S}_k\}_{k=1}^K$. The task of subspace clustering or segmentation is to segment the sample set $\mathbf{Y}$ according to the underlying subspaces.

In the past decade, sparse and low rank theories have been applied to subspace clustering successfully. Elhamifar and Vidal [5] proposed Sparse Subspace Clustering (SSC) method. In this method, the authors aims to find the sparsest representation by using ℓ_1 norm. The detailed SSC model is defined as follows:

$$\begin{aligned} \min_{\mathbf{C},\mathbf{E},\mathbf{Z}} \quad & \|\mathbf{C}\|_1 + \lambda_e \|\mathbf{E}\|_1 + \frac{\lambda_z}{2}\|\mathbf{Z}\|_F^2, \\ \text{s.t.} \quad & \mathbf{Y} = \mathbf{YC} + \mathbf{E} + \mathbf{Z}, \ \mathrm{diag}(\mathbf{C}) = \mathbf{0}. \end{aligned} \tag{7}$$

where $\mathbf{Z}$ represents the Gaussian noise, $\mathbf{E}$ is high magnitude sparse noise and $\mathbf{C}$ represents the affinity matrix. The learned $\mathbf{C}$ can be used for final clustering by NCut [9].

Instead of sparse representation, Liu *et al.* [7] proposed LRR method for clustering by using low rank constraint or nuclear norm $\|\cdot\|_*$ for the coefficient matrix. The general LRR model is shown as follows:

$$\min_{\mathbf{C},\mathbf{E}} \quad \|\mathbf{C}\|_* + \lambda\|\mathbf{E}\|_{2,1}, \quad \text{s.t.} \quad \mathbf{Y} = \mathbf{YC} + \mathbf{E}. \tag{8}$$

where $\|\cdot\|_{2,1}$ represents the $\ell_{2,1}$ norm which is used to improve the robustness of the model for gross noise and data outliers. Furthermore, researchers extended SSC and LRR by introducing some other extra penalty terms to model the spatial correlation between sample data. Tierney *et al.* [6] proposed Ordered Subspace Clustering (OSC) method which provides a more robust model to deal with the sequential data clustering problem. Zhang *et al.* [8] proposed a Robust latent low rank representation (RobustLatLRR) method for subspace clustering. After learning the coefficient matrix, we could construct an affinity matrix to simulate the similarity among data. Then the affinity matrix could be used in the followup spectral clustering.

Although these vector subspace clustering methods have achieved lots of success, the development of multi-way representation method [16] makes a more promised way to the multi-way data clustering. Recently, the t-product has been adopted as an advantageous generalization of matrix multiplication for third or higher order tensors [15, 17]. It has also been applied in some image applications successfully, i.e. face recognition [11] and video completion [18]. Kernfeld *et al.* [12] used t-product for 2D images clustering by keeping the matrix structure and proposed a Sparse Submodule Clustering method (SSmC). This is the most related work to this study. The general SSmC model is defined as follows:

$$\begin{aligned} \min_{\mathcal{C}} \quad & \|\mathcal{C}\|_{F1} + \lambda_h\|\mathcal{C}\|_{FF1} + \lambda_g\|\mathcal{Y} - \mathcal{Y} * \mathcal{C}\|_F^2, \\ \text{s.t.} \quad & \mathcal{C}(i,i,k) = 0, \end{aligned} \tag{9}$$

where $\mathcal{Y}$ is a tensor with 2D image samples, $\mathcal{C}$ is affinity tensor, $*$ denotes the t-product, $\|\cdot\|_{F1}$ and $\|\cdot\|_{FF1}$ are the tensor sparse norms extended from matrix sparse norms as described in Section 2. Compared with the conventional subspace clustering methods, SSmC has the most significant advantage of keeping the 2D structure for images instead of mapping them to vectors and adopts t-product to multiplying two matrices instead of scalar product. However, SSmC considers the sparse structure only and ignores the inner correlation of image samples from the same submodule. To explore such inner correlation, we extend the low-rank structure for tensors and utilize the t-product operation to construct a submodule clustering method. The new method offers good clustering results as demonstrated in experiments.

4. Submodule Clustering by Sparse and Low-Rank Representation

In this section, we describe the proposed method in detail. As described above, we assume all the oriented matrix data are lying on a union of disjoint free submodules instead of subspace. Our goal is to find these submodules and group the data into their respective clusters. We denote $\mathcal{Y}$ as the 2D matrix image samples tensor with the size of $H \times N \times D$, which is obtained by twisting each $H \times D$ image sample into a $H \times 1 \times D$ tensor. Further denote by $\mathcal{C}$ the t-linear combination tensor of size of $N \times N \times D$ which is used to represent the samples tensor $\mathcal{Y}$ itself by minimizing the error $\|\mathcal{Y} - \mathcal{Y} * \mathcal{C}\|_F^2$. Under the t-linear combination of tensors specified in Definitions 1 to 4, we extend the sparse and low-rank representations for tensors. The sparse t-linear combination model for 2D images submodule clustering, which is similar to [12], is defined as follows:

$$\min_{\mathcal{C}} \quad \alpha\|\mathcal{C}\|_{F1} + \frac{1}{2}\|\mathcal{Y} - \mathcal{Y} * \mathcal{C}\|_F^2. \tag{10}$$

where α is a tunable parameter. Under the $F1$ norm, the model enforces that each oriented 2D image matrix can be sparsely represented as a t-linear representation by all other oriented 2D images. Similar to the sparse subspace representations, the non-zero fibers in $\mathcal{C}$ correspond to the samples from the same submodule and the zero ones from the samples from different submodules. We have an illustration of sparse t-linear combination for one sample shown in Figure 3:

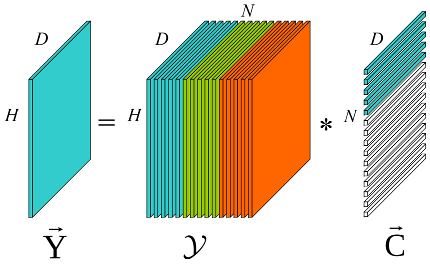

Figure 3. An illustration of sparse t-linear combination. The samples with same color in $\mathcal{Y}$ are from the same submodule. The white tube fibers in $\vec{\mathbf{C}}$ represent the zeros fibers.

As demonstrated in Figure 3, model (10) is equivalent to a number of individual models. To further explore intrinsic correlation information among the entire dataset, inspired by the LRR [7], we propose the following new sparse and low-rank submodule clustering model:

$$\min_{\mathcal{C}} \quad \alpha\|\mathcal{C}\|_{F1} + \lambda\|\mathcal{C}\|_{*f} + \frac{1}{2}\|\mathcal{Y} - \mathcal{Y} * \mathcal{C}\|_F^2. \tag{11}$$

In this model, we adopt the F-nuclear norm, see Definition 4, which is extended from the nuclear norm for t-linear combination. In clustering applications, the number of samples is much larger than the number of classes. So there exists high correlation within the samples. In many typical clustering methods, such as LRR and RobustLatLRR, researchers always use low-rank constraint for the representation matrix to express the inner correlation for samples. The nuclear norm is the most suitable substitute for low-rank constraint. In the case of submodule clustering, we cannot directly use the nuclear norm for t-linear representation tensor, but use the F-nuclear norm defined in Definition 4 instead. With the F-nuclear norm, the low-rank constraint has been adopted for each frontal slice from the representation tensor $\mathcal{C}$. Therefore, the entries could be constrained with high correlation. Figure 4 illustrates low-rank t-linear combination.

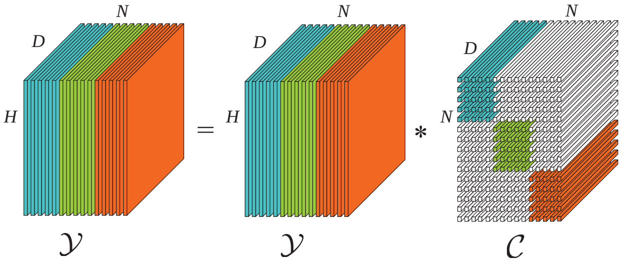

Figure 4. An illustration of low-rank t-linear combination. The samples with same color in $\mathcal{Y}$ are from the same submodule. The white tube fibers in $\mathcal{C}$ represent the zeros fibers.

Furthermore, we should eliminate the self-correlation for each sample in the t-linear combination. Therefore, we add the zero-constraint for the diagonal fibers in the representation tensor. Finally, we obtain a completed sparse and low-rank submodule clustering model as follows,

$$\begin{aligned} \min_{\mathcal{C}} \quad & \alpha\|\mathcal{C}\|_{F1} + \lambda\|\mathcal{C}\|_{*f} + \frac{1}{2}\|\mathcal{Y} - \mathcal{Y} * \mathcal{C}\|_F^2, \\ \text{s.t.} \quad & \mathcal{C}(i,i,k) = 0. \end{aligned} \tag{12}$$

We call (12) the Sparse and Low-Rank Submodule Clustering (SLRSmC) Method based on t-product. Compared with the majority of existing clustering methods, the most significant difference is that we keep the 2D matrix structure of image samples and cluster them in terms of submodules. In addition, we adopt $F1$ and F-nuclear norms for t-linear combination. After solving the above model and obtaining the t-linear combination tensor $\mathcal{C}$, we build the following affinity matrix $\mathbf{W}$

$$\mathbf{W}(i,j) = \|\mathcal{C}(i,j,:)\|_F + \|\mathcal{C}(j,i,:)\|_F \tag{13}$$

which can be pipelined into the NCut to obtain final clustering results. In the next section, we will propose an efficient algorithm to solve the optimization problem (12).

5. Optimization

To solve problem (12), we adopt an alternating direction method by dividing (12) into several subproblems. First, we introduce two auxiliary variables $\mathcal{Z} = \mathcal{C}$ and $\mathcal{X} = \mathcal{C}$ to separate the first two terms in the objective (12). The original model could be re-written as follows:

$$\begin{aligned} \min_{\mathcal{Z},\mathcal{X},\mathcal{C}} \quad & \alpha\|\mathcal{X}\|_{F1} + \lambda\|\mathcal{Z}\|_{*f} + \frac{1}{2}\|\mathcal{Y} - \mathcal{Y} * \mathcal{C}\|_F^2, \\ \text{s.t.} \quad & \mathcal{X}(i,i,k) = 0, \mathcal{X} = \mathcal{C}, \mathcal{Z} = \mathcal{C}. \end{aligned} \tag{14}$$

We adopt the alternating direction method of multipliers (ADMM) [19] to solve the above problem. Then the Augmented Lagrangian for the two introduced constraints is

$$\begin{aligned} \mathcal{L}(\mathcal{X},\mathcal{Z},\mathcal{C}) = & \alpha\|\mathcal{X}\|_{F1} + \lambda\|\mathcal{Z}\|_{*f} + \frac{1}{2}\|\mathcal{Y} - \mathcal{Y} * \mathcal{C}\|_F^2 \\ & + \langle \mathcal{G}_1, \mathcal{Z} - \mathcal{X} \rangle + \langle \mathcal{G}_2, \mathcal{X} - \mathcal{C} \rangle \\ & + \frac{\gamma}{2}(\|\mathcal{Z} - \mathcal{C}\|_F^2 + \|\mathcal{X} - \mathcal{C}\|_F^2), \\ \text{s.t.} \quad & \mathcal{X}(i,i,k) = 0. \end{aligned} \tag{15}$$

where $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_i \text{tr}(\mathbf{A}^{(i)^T}\mathbf{B}^{(i)})$, $\mathcal{G}_1$ and $\mathcal{G}_2$ are Lagrange multipliers and $\gamma > 0$ is a penalty parameter. Therefore, the overall algorithm can be decomposed into solving three subproblems:

1. Updating $\mathcal{X}$ with fixed $\mathcal{Z}$ and $\mathcal{C}$

$$\begin{aligned} \mathcal{X}^{t+1} &= \arg\min_{\mathcal{X}} \alpha\|\mathcal{X}\|_{F1} + \langle \mathcal{G}_2, \mathcal{X} - \mathcal{C} \rangle + \frac{\gamma}{2}\|\mathcal{X} - \mathcal{C}\|_F^2, \\ &= \arg\min_{\mathcal{X}} \alpha\|\mathcal{X}\|_{F1} + \frac{\gamma}{2}\|\mathcal{X} - \mathcal{A}\|_F^2. \end{aligned} \tag{16}$$

where $\mathcal{A} = \mathcal{C} - \frac{\mathcal{G}_2}{\gamma}$. Therefore, we could solve $\mathcal{X}$ fiber-by-fiber from the third dimension with the constraint of $\mathcal{X}_{iik} = 0$ as follows:

$$\mathcal{X}(i,j,:)^{t+1} = \begin{cases} \frac{\|\mathcal{A}(i,j,:)\|_F - \frac{\alpha}{\gamma}}{\|\mathcal{A}(i,j,:)\|_F}\mathcal{A}(i,j,:), & \text{if } i \neq j \text{ and} \\ & \|\mathcal{A}(i,j,:)\|_F > \frac{\alpha}{\gamma}, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

2. Updating $\mathcal{Z}$ with fixed $\mathcal{X}$ and $\mathcal{C}$

$$\begin{aligned} \mathcal{Z}^{t+1} &= \arg\min_{\mathcal{Z}} \lambda\|\mathcal{Z}\|_{*f} + \langle \mathcal{G}_1, \mathcal{Z} - \mathcal{C} \rangle + \frac{\gamma}{2}\|\mathcal{Z} - \mathcal{C}\|_F^2, \\ &= \arg\min_{\mathcal{Z}} \lambda\|\mathcal{Z}\|_{*f} + \frac{\gamma}{2}\|\mathcal{Z} - \mathcal{B}\|_F^2. \end{aligned} \tag{18}$$

where $\mathcal{B} = \mathcal{C} - \frac{\mathcal{G}_1}{\gamma}$. Therefore, we could solve $\mathcal{Z}$ slice-by-slice from the frontal side as follows:

$$\mathbf{Z}^{(i)^{t+1}} = \arg\min_{\mathbf{Z}^{(i)}} \lambda\|\mathbf{Z}^{(i)}\|_* + \frac{\gamma}{2}\|\mathbf{Z}^{(i)} - \mathbf{B}^{(i)}\|_F^2. \tag{19}$$

This problem has closed-form solution according to [20]

$$\mathbf{Z}^{(i)^{t+1}} = \mathbf{U}S_{\frac{\lambda}{\gamma}}[\mathbf{\Sigma}]\mathbf{V}^T, \tag{20}$$

where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the singular value decomposition (SVD) of $\mathbf{B}^{(i)}$. $S_{\frac{\lambda}{\gamma}}[\cdot]$ is the soft-thresholding operator with the following definition:

$$S_{\frac{\lambda}{\gamma}}[x] = \text{sign}(x)\max\{|x| - \frac{\lambda}{\gamma}, 0\}. \tag{21}$$

3. Updating $\mathcal{C}$ with fixed $\mathcal{Z}$ and $\mathcal{X}$

$$\begin{aligned}\mathcal{C}^{t+1} &= \arg\min_{\mathcal{C}} \frac{1}{2}\|\mathcal{Y} - \mathcal{Y} * \mathcal{C}\|_F^2 + \langle \mathcal{G}_1, \mathcal{Z} - \mathcal{C}\rangle \\ &+ \langle \mathcal{G}_2, \mathcal{X} - \mathcal{C}\rangle + \frac{\gamma}{2}(\|\mathcal{Z} - \mathcal{C}\|_F^2 + \|\mathcal{X} - \mathcal{C}\|_F^2) \\ &= \arg\min_{\mathcal{C}} \frac{1}{2}\|\mathcal{Y} - \mathcal{Y} * \mathcal{C}\|_F^2 + \frac{\gamma}{2}(\|\mathcal{C} - \mathcal{P}_1\|_F^2 + \|\mathcal{C} - \mathcal{P}_2\|_F^2).\end{aligned} \tag{22}$$

where $\mathcal{P}_1 = \mathcal{Z} + \frac{\mathcal{G}_1}{\gamma}$ and $\mathcal{P}_2 = \mathcal{X} + \frac{\mathcal{G}_2}{\gamma}$. According to [14], we can solve the problem above in Fourier domain.

$$\begin{aligned}\hat{\mathcal{C}}^{t+1} = \arg\min_{\hat{\mathcal{C}}} \frac{1}{2}\|\hat{\mathcal{Y}} - \hat{\mathcal{Y}} \odot \hat{\mathcal{C}}\|_F^2 + \frac{\gamma}{2}(\|\hat{\mathcal{C}} - \hat{\mathcal{P}}_1\|_F^2 \\ + \|\hat{\mathcal{C}} - \hat{\mathcal{P}}_2\|_F^2).\end{aligned} \tag{23}$$

where $\odot$ represents the face-product. Therefore, we can solve $\hat{\mathcal{C}}$ slice-by-slice from the frontal side.

$$\begin{aligned}\hat{\mathbf{C}}^{(i)^{t+1}} = \arg\min_{\hat{\mathbf{C}}^{(i)}} \frac{1}{2}\|\hat{\mathbf{Y}}^{(i)} - \hat{\mathbf{Y}}^{(i)}\hat{\mathbf{C}}^{(i)}\|_F^2 \\ + \frac{\gamma}{2}(\|\hat{\mathbf{C}}^{(i)} - \hat{\mathbf{P}}_1^{(i)}\|_F^2 + \|\hat{\mathbf{C}}^{(i)} - \hat{\mathbf{P}}_2^{(i)}\|_F^2).\end{aligned} \tag{24}$$

Let $f(\hat{\mathbf{C}}^{(i)}) = \frac{1}{2}\|\hat{\mathbf{Y}}^{(i)} - \hat{\mathbf{Y}}^{(i)}\hat{\mathbf{C}}^{(i)}\|_F^2 + \frac{\gamma}{2}(\|\hat{\mathbf{C}}^{(i)} - \hat{\mathbf{P}}_1^{(i)}\|_F^2 + \|\hat{\mathbf{C}}^{(i)} - \hat{\mathbf{P}}_2^{(i)}\|_F^2)$ and set $\frac{\partial f}{\partial \hat{\mathbf{C}}^{(i)}} = 0$, we have:

$$\hat{\mathbf{C}}^{(i)^{t+1}} = (\hat{\mathbf{Y}}^{(i)T}\hat{\mathbf{Y}}^{(i)} + 2\gamma\mathbf{I})^{-1}(\hat{\mathbf{Y}}^{(i)T}\hat{\mathbf{Y}}^{(i)} + \gamma(\hat{\mathbf{P}}_1^{(i)} + \hat{\mathbf{P}}_2^{(i)})). \tag{25}$$

where $\mathbf{I}$ represents identity matrix. After solving each frontal slice of $\hat{\mathcal{C}}$, we could get the solution of $\mathcal{C}$ as follows,

$$\mathcal{C}^{t+1} = \text{ifft}(\hat{\mathcal{C}}^{t+1}, [\,], 3). \tag{26}$$

4. Updating $\mathcal{G}_1, \mathcal{G}_2$ and γ

$$\mathcal{G}_1^{t+1} = \mathcal{G}_1^t + \gamma^t(\mathcal{Z}^{t+1} - \mathcal{C}^{t+1}). \tag{27}$$

$$\mathcal{G}_2^{t+1} = \mathcal{G}_2^t + \gamma^t(\mathcal{X}^{t+1} - \mathcal{C}^{t+1}). \tag{28}$$

$$\gamma^{t+1} = \min\{\rho\gamma^t, \gamma^{max}\}. \tag{29}$$

where $\rho > 1$ is a constant and γ^{max} is the upper bound of γ.

The overall algorithm is summarized in Algorithm 1.

Remark 1: In the algorithm, the stopping criterion is measured by the following condition:

$$\max\left\{\begin{array}{c}\|\mathcal{Z}^{t+1} - \mathcal{C}^{t+1}\|_\infty, \|\mathcal{X}^{t+1} - \mathcal{C}^{t+1}\|_\infty, \\ \|\mathcal{Z}^{t+1} - \mathcal{Z}^t\|_\infty, \|\mathcal{X}^{t+1} - \mathcal{X}^t\|_\infty, \\ \|\mathcal{C}^{t+1} - \mathcal{C}^t\|_\infty\end{array}\right\} \leq \varepsilon. \tag{30}$$

Algorithm 1 The solution of SLRSmC

```
Require: The image sample data tensor $\mathcal{Y}$, and the parameters $\lambda$, $\alpha$
1: Initialize : $\mathcal{Z}^0 = \mathcal{X}^0 = \mathbf{0} \in \Re^{N\times N\times D}$, $\mathcal{G}_1^0 = \mathcal{G}_2^1 = \mathbf{1} \in \Re^{N\times N\times D}$, $\gamma^0 = 0.1$, $\rho = 1.9$, $\gamma^{max} = 10^{10}$, $\varepsilon = 10^{-5}$, the number of maximum iteration $MaxIter = 500$
2: $t = 0$;
3: while not converged and $t \leq MaxIter$ do
4:     Update $\mathcal{X}$ by (16) and (17);
5:     Update $\mathcal{Z}$ by (18) and (21);
6:     Update $\mathcal{C}$ by (22) to (26);
7:     Update $\mathcal{G}_1$ and $\mathcal{G}_2$ by (27) and (28);
8:     Update $\gamma$ by (29);
9:     Check the convergence conditions defined as (30);
10:    $t = t + 1$.
11: end while
Ensure:
    The tensor $\mathcal{Z}, \mathcal{X}, \mathcal{C}$.
```

Remark 2: We can make steps 4-8 in the Algorithm parallel as suggested in [19], then it can be proved that Algorithm 1 is convergent, see [19]. Our experiments show that the convergence speed is relatively high and a value of MaxIter < 100 is sufficient for good results.

6. Experimental Results

To evaluate the proposed method, we implement clustering experiments on four databases of three types: synthetic, videos and images. We aim to cluster the image data from video clips or image sequences for the purpose of video segmentation/images clustering. The performance of the proposed method is compared with some state-of-the-art clustering algorithms, such as SSC [5], OSC [6], LRR [7], RobustLatLRR [8] and SSmC [12].

6.1. Synthetic Experiment

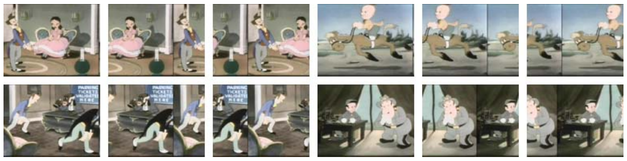

Figure 5. Some images of the synthetic data.

As described above, all the sample data can be arranged as a set of oriented matrices which is regarded as module over the ring $\mathbb{K}_D$. In this experiment, we design a synthetic set of ring data to evaluate the performance of the proposed method. We choose ten different scenes from a video sequence freely available from the Internet Archive[1]. For each

---

[1] `http://archive.org/`

K	Noise	SSC	OSC	LRR	RobustLatLRR	SSmC	Ours
3	0%	**0** (±0)	54.59 (±15.70)	40.59 (±19.19)	39.86 (±18.25)	**0** (±0)	**0** (±0)
	20%	32.71 (±21.99)	42.32 (±12.15)	41.24 (±16.92)	39.56 (±18.31)	0.14 (±0.72)	**0.03** (±0.13)
	50%	35.12 (±5.88)	43.98 (±21.08)	56.71 (±10.81)	40.26 (±3.97)	4.40 (±3.82)	**3.13** (±8.12)
	80%	36.88 (±13.90)	44.69 (±9.73)	55.66 (±3.63)	46.95 (±3.27)	6.04 (±10.70)	**5.94** (±10.26)
4	0%	**0** (±0)	58.75 (±10.76)	48.64 (±9.83)	41.56 (±7.25)	**0** (±0)	**0** (±0)
	20%	40.81 (±14.28)	54.59 (±13.73)	49.07 (±4.88)	47.26 (±4.92)	1.15 (±3.73)	**1.04** (±3.76)
	50%	42.55 (±15.62)	51.62 (±10.78)	68.69 (±1.67)	54.62 (±2.25)	3.28 (±5.20)	**2.79** (±4.18)
	80%	45.63 (±7.66)	54.53 (±6.06)	69.02 (±8.90)	59.56 (±8.72)	**6.97** (±7.62)	7.66 (±6.75)
5	0%	**0** (±0)	68.72 (±8.62)	51.25 (±10.25)	47.62 (±9.13)	**0** (±0)	**0** (±0)
	20%	40.70 (±10.01)	49.34 (±5.91)	51.01 (±7.49)	47.92 (±7.23)	1.00 (±3.47)	**0.42** (±0.86)
	50%	41.68 (±9.79)	53.60 (±6.64)	72.66 (±2.29)	65.95 (±2.46)	6.30 (±7.76)	**5.36** (±7.22)
	80%	38.13 (±13.32)	57.36 (±7.68)	71.89 (±2.02)	67.95 (±2.37)	11.28 (±6.86)	**8.87** (±7.03)
6	0%	**0** (±0)	69.98 (±4.88)	53.78 (±6.20)	48.56 (±6.42)	**0** (±0)	**0** (±0)
	20%	42.79 (±9.52)	54.04 (±7.88)	55.70 (±5.91)	49.56 (±4.23)	0.33 (±0.42)	**0.29** (±0.59)
	50%	45.34 (±12.35)	58.62 (±6.79)	69.24 (±9.80)	59.59 (±8.26)	7.29 (±5.64)	**6.64** (±5.80)
	80%	47.58 (±10.62)	63.49 (±5.11)	57.58 (±5.55)	52.03 (±5.21)	12.54 (±5.66)	**11.04** (±6.60)
7	0%	**0** (±0)	71.94 (±5.88)	55.94 (±5.78)	49.56 (±5.02)	**0** (±0)	**0** (±0)
	20%	43.05 (±13.37)	53.86 (±5.10)	58.93 (±4.83)	51.94 (±3.25)	**0.43** (±0.72)	0.45 (±0.64)
	50%	45.09 (±10.86)	59.21 (±5.89)	58.21 (±5.21)	53.62 (±5.02)	7.84 (±3.24)	**7.68** (±2.87)
	80%	52.87 (±4.96)	66.26 (±2.70)	64.30 (±4.69)	59.87 (±4.12)	17.70 (±4.11)	**16.63** (±8.15)

Table 1. Misclassification rates (%) on the synthetic data with different noise, lower is better. Numbers in brackets indicate the standard deviation in each case.

of the ten randomly chosen frames (images), we shift the first 10 columns to the right end of the image and repeat 64 times. By this way, we synthetically produce module subspace along image rows. Finally, we obtain 10 image sequences in which each sequence consists of 64 continuously shifted images. The pre-processing of sequences includes converting color images to grayscale and downsampling to the resolution of 90×120. The sequence examples are shown in Figure 5. From these 10 sequences, we randomly selected K= [3, 4, 5, 6, 7] sequences. Thus in total we have $64 \times K$ 2D frames (images) in each clustering test. To further test the robustness of the proposed method, we add 3 different magnitudes (20%, 50%, 80%) of Gaussian noise into the samples. For each K, we conduct 20 experiments. The average results are shown in Table 1. The best results is in bold text and the second one is underlined. From the results, we observe that our method outperforms all other methods in most cases especially with larger magnitudes of noise. The experimental results demonstrate that our method is more efficient than others.

6.2. UCSD dynamic scenes benchmarking datasets

Figure 6. Some images of the UCSD dynamic scenes dataset.

This dataset[2] provides a wide range of different challenges and environmental settings including occlusion, camera motion, clustered background and especially highly dynamic backgrounds such as smoke, fire, swing trees as well as water waves. We select 10 categories from the dataset, in which the background changes evidently including *birds, boats, bottle, chopper, cyclists, flock, hockey, landing, ocean, skiing*. Each category has 30 to 246 2D images. We down sampling all the images to the resolution of 90×135. The examples are shown in Figure 6. In this set of experiments, we select K =[3, 4, 5, 6, 7] categories, each of which characterize certain degree of scene shifting from left to right. All the 2D images from the selected sequences are collected for clustering. We repeat the experiment 20 times for each K and the average cluster results are shown in Table 2. The best results is in bold text and the second one is underlined.

It shows that the proposed SLRSmC method has superior performance against others though the performance degrades with the increasing number of class.

6.3. Olympic Sports Dataset

The Olympic Sports Dataset [22] contains videos of athletes practicing different sports obtained from YouTube and annotated using Amazon Mechanical Turk. There are 16 sports actions among a total of 783 video sequences. We choose six scenes with slightly larger change including *high-jump, long-jump, pole-vault, basketball lay-up, javelin*

---

[2] `http://www.svcl.ucsd.edu/projects/background_subtraction/`

K	SSC	OSC	LRR	RobustLatLRR	SSmC	Ours
3	28.79 (±13.26)	17.32 (±16.60)	10.14 (±14.79)	9.23 (±14.21)	9.02 (±11.93)	**4.50** (±10.21)
4	29.29 (±7.92)	21.22 (±14.48)	17.47 (±14.37)	16.25 (±14.52)	12.76 (±8.40)	**7.66** (±10.82)
5	30.41 (±8.75)	24.83 (±9.12)	24.48 (±15.96)	17.19 (±15.32)	13.65 (±7.73)	**12.41** (±8.17)
6	31.32 (±7.40)	27.57 (±8.65)	26.04 (±13.49)	20.15 (±12.14)	16.41 (±8.63)	**14.27** (±8.52)
7	33.68 (±5.76)	28.48 (±6.88)	30.85 (±12.28)	28.12 (±12.15)	21.01 (±9.50)	**19.02** (±8.32)

Table 2. Misclassification rates (%) on the UCSD dynamic scenes dataset with different class numbers. The lower the better. Numbers in brackets indicate the standard deviation in each case.

K	SSC	OSC	LRR	RobustLatLRR	SSmC	Ours
2	27.14 (±18.11)	35.13 (±11.34)	32.84 (±10.21)	29.56 (±10.46)	12.35 (±14.51)	**11.57** (±14.72)
3	35.72 (±10.46)	48.73 (±10.59)	43.10 (±5.99)	38.26 (±5.72)	20.43 (±13.10)	**19.65** (±12.91)
4	35.92 (±10.92)	47.77 (±6.78)	53.51 (±7.51)	48.62 (±8.21)	33.41 (±13.21)	**25.11** (±11.75)
5	43.91 (±11.26)	47.03 (±5.51)	47.72 (±4.92)	42.68 (±4.72)	33.01 (±9.11)	**27.40** (±15.81)

Table 3. Misclassification rates (%) on the Olympic Sports Dataset with different class numbers. The lower the better. Numbers in brackets indicate the standard deviation in each case.

K	SSC	LRR	RobustLatLRR	SSmC	Ours
5	16.37 (±11.64)	18.11 (±11.51)	15.22 (±12.69)	12.56 (±12.22)	**11.23** (±11.75)
6	26.51 (±10.99)	23.46 (±9.30)	21.28 (±11.20)	16.78 (±11.03)	**13.46** (±10.03)
7	26.07 (±12.28)	**15.07 (±9.91)**	18.47 (±8.02)	19.87 (±9.68)	17.47 (±11.39)
8	27.37 (±11.13)	18.83 (±9.82)	22.96 (±10.14)	20.60 (±11.60)	**17.56** (±9.96)
9	27.11 (±9.38)	21.91 (±8.99)	23.11 (±9.20)	24.08 (±7.77)	**20.47** (±7.93)
10	27.88 (±9.77)	25.59 (±8.03)	25.17 (±7.62)	26.36 (±11.29)	**21.64** (±8.90)
11	30.48 (±11.02)	28.25 (±8.13)	26.01 (±8.33)	30.56 (±13.58)	**22.15** (±6.90)

Table 4. Misclassification rates (%) on the COIL20 Image Database with different class numbers, lower is better. Numbers in brackets indicate the standard deviation in each case.

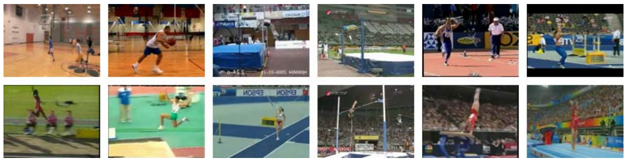

Figure 7. Some images of the Olympic Sports Dataset.

and vault. For each class, we down-sample all the images at the resolution of 90×120. The examples are shown in Figure 7. In this set of experiments, we set the cluster number K to [2, 3, 4, 5], then collect the frames for clustering. Similarly we repeat the experiment 20 times for each K and the average results are reported in Table 3.

It can be observed that our method outperforms all the other methods in all cases. This demonstrates that the introduced method enhances the clustering performance due to the t-linear combination.

6.4. COIL20 Image Database

Finally we test the clustering performance of our method on the COIL image database [21] for image clustering. The database contains 20 objects viewed from varying angels, as demonstrated in Figure 8. In this experiment, similarly we consider $K = [5, 6, 7, 8, 9, 10, 11]$ objects and pick up 36 images from each class for clustering. Each image sample is down-sampled to the resolution of 32×32. As there is no special order specified for image samples, the OSC method is not suitable for this experiment. The experiment has been repeated 20 times for each K. Table 4 reports the overall results, which indicate that our SLRSmC consistently outperforms all the other methods in most cases.

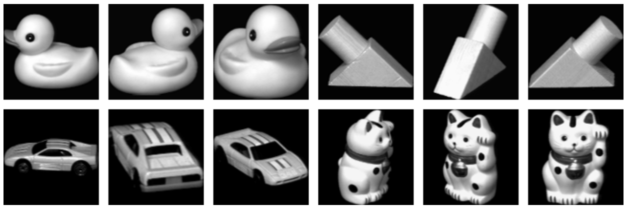

Figure 8. Some images of the COIL20 Image Database.

7. Conclusion

In this paper, we proposed a new image submodule clustering method by combining sparse representation, low-rank representation and t-product. Different from other typical clustering methods, we keep the 2D structure of image data without vectorizing them. The data affinity information is learned by exploring the embedded submodule structure by using the sparse and low-rank representation over tensors with t-product operation. Our experiment results have demonstrated that the t-product assisted low-rank representation does facilitate clustering based on submodule information, evidenced by the better clustering results.

References

[1] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *IEEE TNN*, vol. 16, no. 3, pp. 645–678, 2005.

[2] S. Krinidis and V. Chatzis, “A robust fuzzy local information c-means clustering algorithm,” *IEEE TIP*, vol. 19, no. 5, pp. 1328–1337, 2010.

[3] A. Biswas and D. Jacobs, “Active image clustering: Seeking constraints from humans to complement algorithms,” in *CVPR*, 2012. pp. 2152–2159.

[4] R. Vidal. “ A tutorial on subspace clustering,” *IEEE Signal Proc. Mag.*, vol. 28, no. 2, pp. 52–68, 2011.

[5] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE TPAMI*, vol. 35, no. 1, pp. 2765–2781, 2013.

[6] S. Tierney, J. Gao, and Y. Guo, “Subspace clustering for sequential data,” in *CVPR*, 2014, pp. 1019-1026.

[7] G. Liu, Z. Lin, and Y. Yu, “Robust subspace segmentation by low-rank representation,” in *ICML*, 2010. pp. 663–670.

[8] H. Zhang, Z. Lin, C. Zhang, and J. Gao, “Robust latent low rank representation for subspace clustering.” *Neurocomputing*, vol. 145, pp. 369–373, Dec. 2014.

[9] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE TPAMI*, vol. 22, no. 1, pp. 888–905, 2000.

[10] X. He, D. Cai, and P. Niyogi, “Tensor subspace analysis,” in *NIPS*, 2005. pp. 499–506.

[11] N. Hao, M. Kilmer, K. Braman, and R. Hoover. “Facial recognition using tensor-tensor decompositions,” *SIAM J on Imaging Sciences,* vol. 6, no. 1, pp. 437–463, 2013.

[12] E. Kernfeld, S. Aeron, and M. Kilmer, “Clustering multiway data: a novel algebraic approach,” *arXiv preprint arXiv:1412.7056.*, 2015.

[13] S. Aeron and E. Kernfeld, “Group-Invariant subspace clustering,” in *arXiv preprint arXiv:1510.04356*, 2015.

[14] M. E. Kilmer and C. D. Martin, “Factorization strategies for third-order tensors,” *Lin. Alg. and its Appls*, vol. 435, no. 3, pp. 641–658, 2011.

[15] M. Kilmer, K. Braman, N. Hao, and R. Hoover, “Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging,” *SIAM J Matr Anal and Appls*, vol. 34, no. 1, pp. 148–172, 2013.

[16] K. Braman, “Third-order tensors as linear operators on a space of matrices,” *Lin Alg and its Appls*, vol. 433, no. 7, pp. 1241–1253, 2010.

[17] C. D. Martin, R. Shafer, B. LaRue, “An order-p tensor factorization with applications in imaging,” *SIAM J on Sci Comp*, vol. 35, no. 1, pp. A474–A490, 2013.

[18] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, “A New Low-Rank Tensor Model for Video Completion,” *arXiv preprint arXiv:1509.02027.*, 2015.

[19] 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,*, vol. 99, no. 2, pp. 287–325, 2015.

[20] J. F. Cai, E. J. Candès, Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM J on Opt.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[21] S. A. Nene, S. K. Nayar, and H. Murase, “Columbia object image library (COIL-20),” *Technical Report CUCS-005-96*, Feb. 1996.

[22] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, “Modeling temporal structure of decomposable motion segments for activity classification,” in *ECCV*, 2010. pp. 392–405.