2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Tensor LRR Based Subspace Clustering

Yifan Fu, Junbin Gao and David Tien
School of Computing and Mathematics
Charles Sturt University
Bathurst, NSW 2795, Australia
Email:{yfu, jbgao, dtien}@csu.edu.au

Zhouchen Lin [1]
Key Laboratory of Machine Perception (MOE)
School of Electronics Engineering and Computer Science
Peking University, Beijing, China
Email: zlin@pku.edu.cn

***Abstract*—Subspace clustering groups a set of samples (vectors) into clusters by approximating this set with a mixture of several linear subspaces, so that the samples in the same cluster are drawn from the same linear subspace. In majority of existing works on subspace clustering, samples are simply regarded as being independent and identically distributed, that is, arbitrarily ordering samples when necessary. However, this setting ignores sample correlations in their original spatial structure. To address this issue, we propose a tensor low-rank representation (TLRR) for subspace clustering by keeping available spatial information of data. TLRR seeks a lowest-rank representation over all the candidates while maintaining the inherent spatial structures among samples, and the affinity matrix used for spectral clustering is built from the combination of similarities along all data spatial directions. TLRR better captures the global structures of data and provides a robust subspace segmentation from corrupted data. Experimental results on both synthetic and real-world datasets show that TLRR outperforms several established state-of-the-art methods.**

I. Introduction

Due to rapid development of storage, sensing, networking, and communication technologies, recent years have witnessed a gigantic increase in the availability of multidimensional data. These massive multidimensional data are often high-dimensional with a large amount of redundancy. This prompts the development of finding a low-dimensional representation that best fits a set of samples from a high-dimensional space. *Linear subspace learning* is a kind of traditional dimensionality reduction techniques that finds an optimal linear mapping to a lower dimensional space. For example, Principle Component Analysis (PCA) [1] is essentially based on the hypothesis that the data are drawn from a low-dimensional subspace. However, a data set is not often well described by a *single* subspace in practice. Therefore, it is more reasonable to consider data lying on a mixture of multiple low-dimensional subspaces, with each subspace fitting a subgroup of data. The objective of subspace clustering is to assign data to their relevant subspace clusters based on, for example, a low-dimensional representation for each high-dimensional sample. In the last decade, subspace clustering (SC) has been widely applied to many real-world applications, including motion segmentation [2], [3], social community identification [4], and image clustering [5]. A famous survey on subspace clustering [6] classifies most existing SC algorithms into three categories: statistical methods [7] , algebraic methods [8] and spectral clustering-based methods [3], [8].

In SC algorithms, a given high dimensional data set is represented by a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$, with each sample $\mathbf{x}_i$ $(1 \leq i \leq N)$ denoted by a column vector. In many applications, samples may have multi-dimensional spatial structure forms as shown in Fig. 1 (a 2-dimension/mode scene image and a 3-dimension/mode silhouette sequence). In the 2-dimensional scene case, one wishes to cluster all the pixels, each of which is represented as a feature vector such as RGB features. For example, an image of size 32×32 is considered as a matrix of size $\mathbb{R}^{3\times 1024}$ by rearranging the pixels into a list of vectors along row/column direction. The necessity of such "*unfolding*" process is due to the fact that most of the current SC algorithms can only be applied to vectorial data, which breaks the inherent structure and correlations in the original data (the scene in the above case). Take Fig.2 as an example, two samples $\mathbf{x}_{11:}$ and $\mathbf{x}_{21:}$ are close in the 2-dimension structure, so they are highly possible from the same subspace. However, in a traditional SC algorithm, this closeness information is simply ignored as all the pixels are simply regarded as a group of features. Fig. 2(a) shows an example in which the pixels/features are rearranged into a matrix by row-by-row order. Clearly in the row ordering, above two samples are not adjacent in the new sample sequence. A specific order may have some impact over the results from a clustering algorithm. Fortunately, *tensor* is a suitable representation for such multi-dimensional data like image scenes, with a format of a multi-way array. The *order* of a tensor is the number of dimensions, also known as *ways* or *modes*. Thus, a set of sample vectors with an $(N-1)$-dimension spatial structure (2D structure for an image scene and 3D structure for a silhouette sequence) is denoted by an N-mode tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, with mode-$i$ ($1 \leq i \leq N-1$) denoting sample's position along its direction, and the mode-N denoting the sample feature direction, e.g. RGB features in image scenes. Take a 3-mode tensor as an example, mode-1 and mode-2 denote the spatial row and column information of samples, the sample vectors/features are listed among mode-3 as shown in Fig. 2(b).

Therefore, we propose a novel subspace clustering method where the input data are represented in their original structural form as a *tensor*. Our model finds a *lowest-rank representation* for the input *tensor*, which can be further used to build an affinity matrix. The affinity matrix used for spectral clustering records pairwise similarity along all the spatial modes. For a 3-mode tensor, the affinity matrix evaluates their similarities from both row and column directions as shown in Fig. 2(b). In summary, the contribution of our work is twofold:

- We propose a tensor low-rank representation for sub-

[1] Corresponding author

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

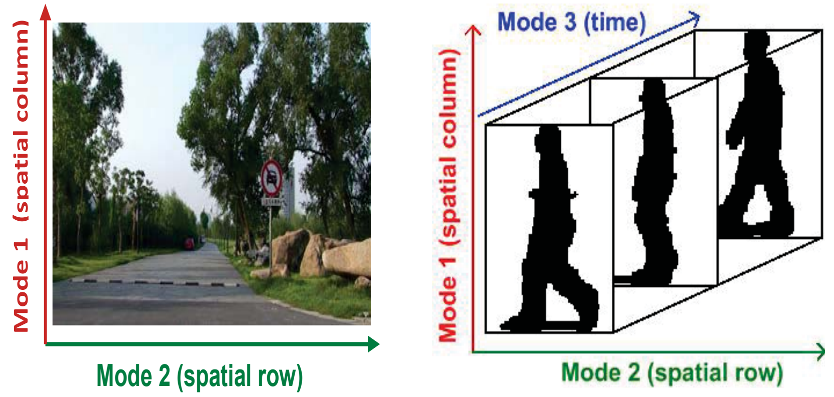
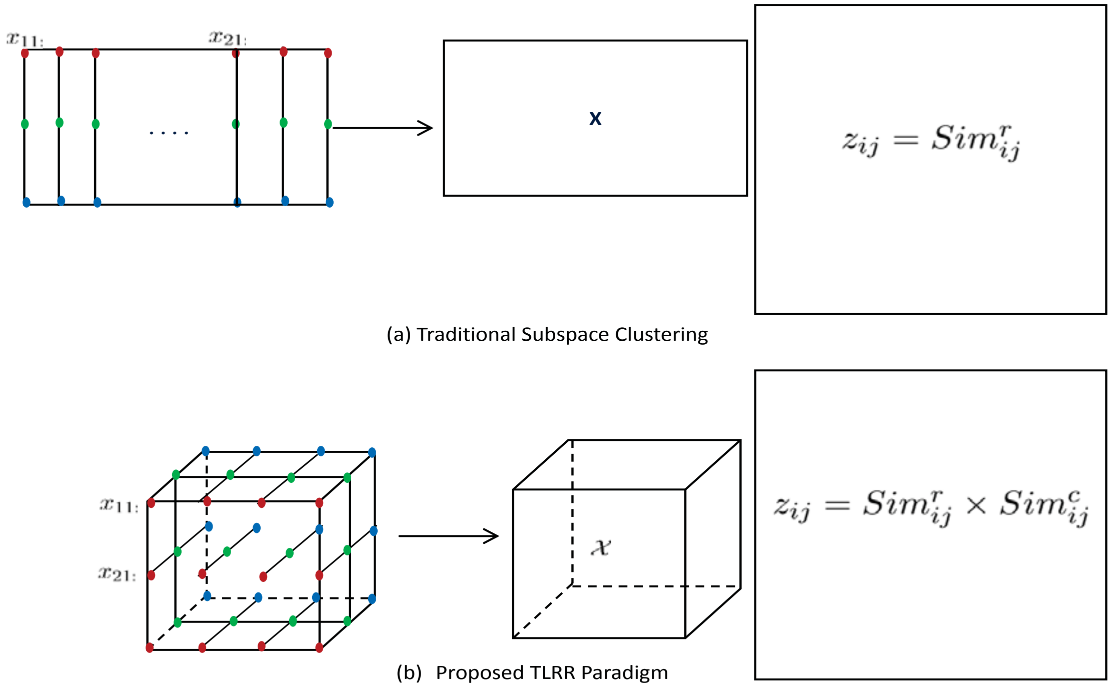


Fig. 1: Illustration of real-world data with multi-dimensional spatial structure information.



(a) Traditional Subspace Clustering

(b) Proposed TLRR Paradigm

Fig. 2: Traditional subspace clustering paradigm *vs.* the proposed TLRR paradigm.

space clustering (TLRR). Unlike previous work treating each individual sample as an independent and identically distributed (i.i.d.) one, our model takes sample spatial structure and correlations into account. Specifically, our method directly seeks for a low-rank representation to samples' natural structural form — a high-order tensor.

- The new SC algorithm based on our model is robust, capable to handle *noise* in the data and segments all samples into their respective subspaces simultaneously.

II. Related Work

The author of [6] classifies existing SC algorithms into three categories: statistical methods, algebraic methods and spectral clustering-based methods.

Statistical models assume that mixed data are formed by a set of independent samples from a mixture of a certain distribution such as Gaussian. Each Gaussian distribution can be considered as a single subspace, then subspace clustering is transformed to be a mixture of Gaussian model estimation problem. This estimation can be obtained by Expectation Maximization (EM) algorithm in Mixture of Probabilistic PCA [9], or serial subspace searching in Random Sample Consensus (RANSAC) [10]. Unfortunately, these solutions are sensitive to noises and outliers. Some efforts have been made to improve algorithm robustness. For example, Agglomerative Lossy Compression (ALC) [11] finds the optimal segmentation that minimizes the overall coding length of the segmented data, subject to each subspace is modeled as a degenerate Gaussian.

However, optimization difficulty is still the bottleneck to solve this problem.

General Principle Component Analysis (GPCA) [12] is an algebraic based method to estimate a mixture of linear subspaces from sample data. It factors a homogeneous polynomial whose degree is the number of subspaces and whose factors (roots) represent normal vectors to each subspace. GPCA has no restriction on subspaces, and works well under certain condition. Nevertheless, the performance of algebraic based methods in the presence of noise deteriorates as the number of subspaces increases. Robust Algebraic Segmentation (RAS) [8] is proposed to improve robustness performance, but the complexity issue still exists.Iterative methods improve the performance of algebraic based algorithms to handle noisy data in a repeated refinement. The *k*-subspace method [7], [13] extends the *k*-means clustering algorithm from data distributed around cluster centers to data drawn from subspaces of any dimensions. It alternates between assigning samples to subspaces and re-estimating subspaces. The *k*-subspace method can converge to a local optimum in a finite number of iterations. Nevertheless, the final solution depends on good initialization and is sensitive to outliers.

Both [3], [8] and [14] are representatives of spectral clustering-based methods. They aim to find a linear representation $\mathbf{Z}$ for all the samples in terms of all other samples, which is solved by finding the optimal solution of the following objective function:

$$\begin{aligned} &\min_{\mathbf{Z}} \parallel \mathbf{E} \parallel_q + \parallel \mathbf{Z} \parallel_b \\ &\text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E} \end{aligned} \tag{1}$$

where $\parallel \cdot \parallel_q$ and $\parallel \cdot \parallel_b$ denote the norms for error and the new representation matrix $\mathbf{Z}$ respectively. Using the resulting matrix $\mathbf{Z}$, an affinity matrix $|\mathbf{Z}|$+$|\mathbf{Z}^T|$ is built and used for spectral clustering. The Sparse Subspace Clustering (SSC) [3] uses the l_1 norm $\parallel \mathbf{Z} \parallel_{l_1}$ in favor of a sparse representation, with an expectation that the within-cluster affinities are sparse (but not zero) and the between-cluster affinities shrink to zero. However, this method is inaccurate at capturing the global structure of data and is not robust to noises in data. The Low-Rank Representation (LRR) [14] employs the nuclear norm $\parallel \mathbf{Z} \parallel_*$ to guarantee a low-rank structure, and the $l_{2,1}$ norm is used in error term to make it robust to outliers.

III. Notations and Problem Formulation

A. Definition and Notations

Before formulating the subspace clustering problem, we first introduce some tensor fundamentals and notations.

Definition 1 (Tensor Matricization): Matricization is the operation of rearranging the entries of a tensor so that it can be represented as a matrix. Let $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ be a tensor of order-N, the mode-n matricization of $\mathcal{X}$ reorders the mode-n vectors to be columns of the resulting matrix, denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (I_{n+1} I_{n+2} \ldots I_N I_1 I_2 \ldots I_{n-1})}$.

Definition 2 (The n-mode Product): The n-mode product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$, denoted as $\mathcal{X} \times_n \mathbf{U}$, is a tensor with entries:

$$(\mathcal{X} \times_n \mathbf{U})_{i_1,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n} \tag{2}$$

The n-mode product is also denoted by each mode-n vector multiplied by the matrix $\mathbf{U}$. Thus, it can be expressed in terms of tensor matricization as well:

$$\mathcal{Y} = \mathcal{X} \times_n \mathbf{U} \qquad \Leftrightarrow \qquad \mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)} \tag{3}$$

Definition 3 (Tucker Decomposition): Given an N-way tensor $\mathcal{X}$, its Tucker decomposition is an approximated tensor defined by,

$$\begin{aligned} \hat{\mathcal{X}} &\equiv [\![\mathcal{G}; \mathbf{U}_1, ..., \mathbf{U}_N]\!] = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \ldots \times_N \mathbf{U}_N \\ &= \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \ldots r_N} \mathbf{u}_{r_1} \circ \mathbf{u}_{r_2} \ldots \circ \mathbf{u}_{r_N} \end{aligned} \tag{4}$$

where $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \ldots \ R_N}$ is called a core tensor and $\mathbf{U}^{(i)} \in \mathbb{R}^{I_i \times R_i} (1 \leq i \leq N)$ are the factor matrices at each mode. The symbol $\circ$ represents the vector outer product.

B. Problem Formulation

Given an N-mode tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, we consider a data set of all the I_N dimensional vectors/features along $\mathcal{X}$'s N-mode (also called N-mode fibres). The size of data set is $(I_1 \times I_2 \times \cdots \times I_{N-1})$. Assume that these samples are drawn from a union of k independent subspaces $\{S_i\}_{i=1}^k$ of unknown dimensions, i.e., $\sum_{i=1}^k S_i = \bigoplus_{i=1}^k S_i$, where $\bigoplus$ is the direct sum. Our purpose is to cluster all the I_N-dimensional vectors from the tensor $\mathcal{X}$ into k subspaces by incorporating their relevant spatial information in the tensor.

IV. Subspace Clustering via Tensor Low-Rank Representation

A. Tensor Low-Rank Representation

The new approach Low-Rank Representation (LRR) [14] is very successful in subspace clustering for even highly corrupted data, outliers or missing entries. LRR is more robust than Sparse Subspace Clustering [3].

Inspired by the idea used in LRR, we consider a model of low-rank representation for an input tensor $\mathcal{X}$ similar to problem (1). Specifically, we decompose the input tensor $\mathcal{X}$ into a Tucker decomposition in which the core tensor $\mathcal{G}$ is the input tensor itself along with a factor matrix $\mathbf{U}_n$ at each mode $n \leq N$. That is, the proposed data representation model is

$$\mathcal{X} = \mathcal{X} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times \ldots \times_N \mathbf{U}_N + \mathcal{E}. \tag{5}$$

Here we are particularly interested in the case where $\mathbf{U}_N = \mathbf{I}$ (identity matrix of order I_N). If we define $\mathbf{Z} = \mathbf{U}_1 \otimes \mathbf{U}_2 \otimes \cdots \otimes \mathbf{U}_{N-1}$, where $\otimes$ denotes *the Kronecker product* [15] of matrices, then based on the above multiple linear model, we may interpret the entries of $\mathbf{Z}$ as the similarities between the pairs of all the vectors along the N-mode of the data tensor $\mathcal{X}$. These similarities are calculated based on the similarities along all the $N-1$ modes through the factor matrices $\mathbf{U}_n$ $(n = 1, ..., N-1)$, each of which measures the similarity at the n-mode.

As in LRR, model (5) uses the data to represent itself, therefore we can expect low-rank factor matrices $\mathbf{U}_n$. It is well known that it is very hard to solve an optimization problem with matrix rank constraints. A common practice is to relax the rank constraint by replacing it with the nuclear norm [16] as suggested by matrix completion methods [17], [18]. Thus, we finally formulate our model as follows,

$$\begin{aligned}\min_{\mathbf{U}_1,\ldots,\mathbf{U}_{N-1}} & \frac{\lambda}{2} \parallel \mathcal{E} \parallel_F^2 + \sum_{n=1}^{N-1} \parallel \mathbf{U}_n \parallel_* \\ \text{s.t. } & \mathcal{X} = \mathcal{X} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_{(N-1)} \mathbf{U}_{N-1} \times_N \mathbf{I} + \mathcal{E}\end{aligned} \tag{6}$$

where $\parallel \cdot \parallel_*$ denotes the *nuclear norm* of a matrix, defined as the sum of singular values of the matrix, $\parallel \cdot \parallel_F$ denotes the Frobenius norm of a tensor, i.e. the square root of the sum of the squares of all its elements, and $\lambda > 0$ is a parameter to balance the two terms, which can be tuned empirically. That is, TLRR seeks optimal low-rank solutions $\mathbf{U}_n (1 \leq n < N)$ of the structured data $\mathcal{X}$ using itself as a dictionary.

B. Solving the Optimization Problem

1) Block Coordinate Descent Algorithm: We employ an iterative algorithm called the Block Coordinate Descent (BCD) [19] to solve the optimization problem (6) by fixing all the other modes variables to solve for one variable at a time alternatively. For instance, TLRR fixes $\mathbf{U}_1, \ldots, \mathbf{U}_{n-1}, \mathbf{U}_{n+1}, \ldots, \mathbf{U}_{N-1}$ to minimize the variable $\mathbf{U}_n (n = 1, 2, \ldots, N)$, which is equivalent to solve the following optimization subproblem:

$$\begin{aligned}\min_{\mathbf{U}_n} & \frac{\lambda}{2} \parallel \mathcal{E} \parallel^2 + \parallel \mathbf{U}_n \parallel_* \\ \text{s.t. } & \mathcal{X} = \mathcal{X} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_{(N-1)} \mathbf{U}_{N-1} \times_N \mathbf{I} + \mathcal{E}\end{aligned} \tag{7}$$

Using tensorial matricization, problem (7) can be rewritten in terms of matrices as follows:

$$\begin{aligned}\min_{\mathbf{U}_n} & \frac{\lambda}{2} \parallel \mathbf{E}_{(n)} \parallel_F^2 + \parallel \mathbf{U}_n \parallel_* \\ \text{s.t. } & \quad \mathbf{X}_{(n)} = \mathbf{U}_n \mathbf{B}_{(n)} + \mathbf{E}_{(n)}\end{aligned} \tag{8}$$

where $\mathbf{B}_{(n)} = \mathbf{X}_{(n)} (\mathbf{U}_{N-1} \otimes \cdots \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \cdots \otimes \mathbf{U}_1)^T$.

Based on Eq.(8), each matrix $\mathbf{U}_n (1 \leq n < N)$ is optimized individually, while the other matrices are held fixed. All the matrices update iteratively until the change in fit drops below a threshold or when the number of iterations reaches a maximum, whichever comes first. The general process of BCD is illustrated by Algorithm 1.

Algorithm 1 Solving Problem (6) by BCD

```
Require: data tensor $\mathcal{X}$, parameters $\lambda$
Ensure: factor matrices $\mathbf{U}_n$ $(n = 1, 2, \ldots, N-1)$
randomly initialize $\mathbf{U}_n \in \mathbb{R}^{I_n \times R_n}$ for $n = 1, \ldots, N-1$
for $n = 1, \ldots, N-1$ do
    $\mathbf{X}_{(n)} \leftarrow$ the mode-n matricization of the tensor $\mathcal{X}$
end for
while reach maximum iterations or converge to stop do
    for $n = 1, \ldots, N-1$ do
        $\mathbf{B}_{(n)} \leftarrow \mathbf{X}_{(n)} (\mathbf{U}_N \otimes \cdots \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \cdots \otimes \mathbf{U}_1)^T$
        $\mathbf{U}_n \leftarrow$ solve the subproblem (8)
    end for
end while
```

Remark 1: Using the Frobenius norm means we are dealing with Gaussian noises in the tensor data. If based on some domain knowledge, we know some noise patterns along a particular mode, for example, in multispectral imaging data, noises in some spectral bands are significant, we may adapt the so-called robust noise models like $l_{2,1}$-norm [20] instead.

Remark 2: There is a clear link between LRR and TLRR. If we consider the mode-N matricization in (6), we will see that it can be converted to an LRR model with $\mathbf{Z} = \mathbf{U}_1 \otimes \mathbf{U}_2 \otimes \cdots \otimes \mathbf{U}_{N-1}$. However, in the standard LRR, such an explicit Kronecker structure in $\mathbf{Z}$ has been ignored, so the number of unknown parameters in $\mathbf{Z}$ is $(I_1 \times I_2 \times \cdots I_{N-1})^2$. This will cause difficulty in LRR algorithm doing SVD. However, TLRR exploits the Kronecker structure with number of unknown parameter reduced to $I_1^2 + I_2^2 + \cdots + I_{N-1}^2$. Our experiments demonstrate TLRR is much faster than LRR.

2) Augmented Lagrange Multiplier: In this subsection, we consider how to solve the subproblem (8).

We use the Augmented Lagrange Multiplier (ALM) method [21] to solve the constrained optimization problem (8). The reason we choose ALM to solve this optimization problem is threefold: (1) Superior convergence property of ALM makes it very attractive; (2) Parameter tuning is much easier than the iterative thresholding algorithm; and (3) It converges to an exact optimal solution.

First of all, the augmented Lagrange problem of (8) can be written as

$$\begin{aligned}L(\mathbf{E}_{(n)}, \mathbf{U}_n, \mathbf{Y}_n) = & \frac{\lambda}{2} \parallel \mathbf{E}_{(n)} \parallel_F^2 + \parallel \mathbf{U}_n \parallel_* \\ & + \text{tr}[\mathbf{Y}_n^T (\mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{B}_{(n)} - \mathbf{E}_{(n)})] \\ & + \frac{\mu_n}{2} \parallel \mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{B}_{(n)} - \mathbf{E}_{(n)} \parallel_F^2 .\end{aligned} \tag{9}$$

The problem (9) can be solved by updating one variable at a time with all the other variables fixed. More specifically, the iterations of ALM go as follows

1) Fix all others to update $\mathbf{E}_{(n)}$ by

$$\min_{\mathbf{E}_{(n)}} \frac{\lambda}{\mu_n} \|\mathbf{E}_{(n)}\|_F^2 + \|\mathbf{E}_{(n)} - (\mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{B}_{(n)} + \frac{\mathbf{Y}_n}{\mu_n})\|_F^2 \tag{10}$$

which is equivalent to a least square problem. The solution is given by

$$\mathbf{E}_n = \frac{\lambda}{\lambda + \mu_n} \left(\mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{B}_{(n)} + \frac{\mathbf{Y}_n}{\mu_n} \right) \tag{11}$$

2) Fix all others to update $\mathbf{U}_n$ by

$$\begin{aligned}\min_{\mathbf{U}_n} & \|\mathbf{U}_n\|_* - \text{tr}[\mathbf{Y}_n^T \mathbf{U}_n \mathbf{B}_{(n)}] \\ & + \frac{\mu_n}{2} \parallel (\mathbf{X}_{(n)} - \mathbf{E}_{(n)}) - \mathbf{U}_n \mathbf{B}_{(n)} \parallel_F^2\end{aligned} \tag{12}$$

3) Fix all others to update $\mathbf{Y}_n$ by

$$\mathbf{Y}_n \leftarrow \mathbf{Y}_n + \mu_n (\mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{B}_{(n)} - \mathbf{E}_{(n)}) \tag{13}$$

However, there is no closed-form solution to problem (12) because of the coefficient $\mathbf{B}_{(n)}$ in the third term. We propose to use the linearized approximation with an added proximal term to approximate the objective in (12) as described in [22]. Suppose that $\mathbf{U}_{(n)}^k$ is the current approximated solution to (12) and the sum of the last two terms is denoted by L, then the first order Taylor expansion at $\mathbf{U}^k_{(n)}$ plus a proximal term is given by

$$\begin{aligned} L \approx & \mu_n \langle (\mathbf{U}^k_n \mathbf{B}_{(n)} + \mathbf{E}_n - \mathbf{X}_{(n)} - \frac{\mathbf{Y}_n}{\mu_n})\mathbf{B}^T_{(n)}, \mathbf{U}_n - \mathbf{U}^k_n \rangle \\ & + \frac{\mu_n \eta_n}{2} \|\mathbf{U}_n - \mathbf{U}^k_n\|^2_F + \text{consts} \end{aligned}$$

Thus, solving (12) can be converted to iteratively solve the following problem

$$\min_{\mathbf{U}_n} \|\mathbf{U}_n\|_* + \frac{\mu_n \eta_n}{2} \|\mathbf{U}_n - \mathbf{U}^k_n + \mathbf{P}_n\|^2_F$$

where $\mathbf{P}_n = \frac{1}{\eta_n}(\mathbf{U}^k_n \mathbf{B}_{(n)} + \mathbf{E}_n - \mathbf{X}_{(n)} - \frac{\mathbf{Y}_n}{\mu_n})\mathbf{B}^T_{(n)}$. The above problem can be solved by applying the SVD thresholding operator to $\mathbf{M}_n = \mathbf{U}^k_n - \frac{1}{\eta_n}(\mathbf{U}^k_n \mathbf{B}_{(n)} + \mathbf{E}_n - \mathbf{X}_{(n)} - \frac{\mathbf{Y}_n}{\mu_n})\mathbf{B}^T_{(n)}$. Take SVD for $\mathbf{M}_n = \mathbf{W}_n \Sigma_n \mathbf{V}^T_n$. Then the new iteration is given by

$$\mathbf{U}^{k+1}_n = \mathbf{W}_n \Sigma_n(\eta_n \mu_n) \mathbf{V}^T_n \tag{14}$$

where $\Sigma_n(\eta_n \mu_n)$ is diagonal with elements $\Sigma_n(\eta_n \mu_n)_{ii} = \max\{0, (\Sigma_n)_{ii} - \frac{1}{\eta_n \mu_n}\}$, see [23].

Algorithm 2 Solving Problem (8) by ALM

```
Require: matrices $\mathbf{X}_{(n)}$and $\mathbf{B}_{(n)}$, parameter $\lambda$
Ensure: : factor matrices $\mathbf{U}_n$
initialize: $\mathbf{U}_n = 0, \mathbf{E}_{(n)} = 0, \mathbf{Y}_n = 0, \mu_n = 10^{-6}, max_u = 10^{10}, \rho = 1.1, \varepsilon = 10^{-8}$ and $\eta_n = \|\mathbf{B}_{(n)}\|^2$.
while $\| \mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{B}_{(n)} - \mathbf{E}_{(n)} \|_\infty \geq \varepsilon$ do
    $\mathbf{E}_{(n)} \leftarrow$ the solution (11) to the subproblem (10);
    $\mathbf{U}_n \leftarrow$ the iterative solution by (14) by for example five iterations;
    $\mathbf{Y}_n \leftarrow \mathbf{Y}_n + \mu_n(\mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{B}_{(n)} - \mathbf{E}_{(n)})$
    $\mu_n \leftarrow \min(\rho \mu_n, max_u)$
end while
```

C. The Complete Subspace Clustering Algorithm

After finding a low-rank representation given by $\mathbf{U}_i (i = 1, 2, \ldots, N-1)$ for the data $\mathcal{X}$, we can create a similarity matrix $\mathbf{Z} = \mathbf{U}_1 \otimes \mathbf{U}_2 \otimes \cdots \otimes U_{N-1}$. The affinity matrix is then defined by $|\mathbf{Z}| + |\mathbf{Z}^T|$. Each element of the affinity matrix is the joint similarity between a pair of mode-N vectorial samples across all the $N-1$ modes/directions. Finally, we employ the Normalized Cuts clustering method [24] to divide the samples into their respective subspaces. Algorithm 3 outlines the whole subspace clustering method of TLRR.

Algorithm 3 Subspace Clustering by TLRR

```
Require: structured data: tensor $\mathcal{X}$, number of subspaces $k$
Ensure: : the cluster indicator vector $\mathbf{l}$ with terms of all samples
lowest-rank representation $\mathbf{U}_n (n = 1, 2, \ldots, N-1) \leftarrow$ solve the problem (6)
$\mathbf{Z} \leftarrow \mathbf{U}_1 \otimes \mathbf{U}_2 \otimes \cdots \otimes \mathbf{U}_{N-1}$
$\mathbf{l} \leftarrow NormalizedCuts(|\mathbf{Z}| + |\mathbf{Z}^T|)$
```

V. Experiments and Results

A. Synthetic Datasets

In this section, we evaluate TLRR against state-of-the-art subspace clustering methods on synthetic datasets. We use a synthetic data set containing 3 subspaces, each of which is formed by N_i samples of 5 dimensions, where $i \in \{1, 2, 3\}, N_1 = 30, N_2 = 24$, and $N_3 = 10$. The generation process is as follows: 1) Select 3 cluster center points $c_i \in \mathbb{R}^5$ for above subspaces respectively, which are far from each other. 2) Generate a matrix $\mathbf{C}^i \in \mathbb{R}^{5 \times N_i}$, each column of which is drawn from a Gaussian distribution $\mathcal{N}(\cdot|c_i, \Sigma^i)$, where $\Sigma^i \in \mathbb{R}^{5\times5}$ is a diagonal matrix with $\Sigma^i_{ii} = 0.01$, and 1s in all other diagonal positions. This setting guarantees the low-rank property in each subspace. 3) Combine samples in each subspace to form an entire data set $\mathbf{X} = \cup \mathbf{C}^i$.

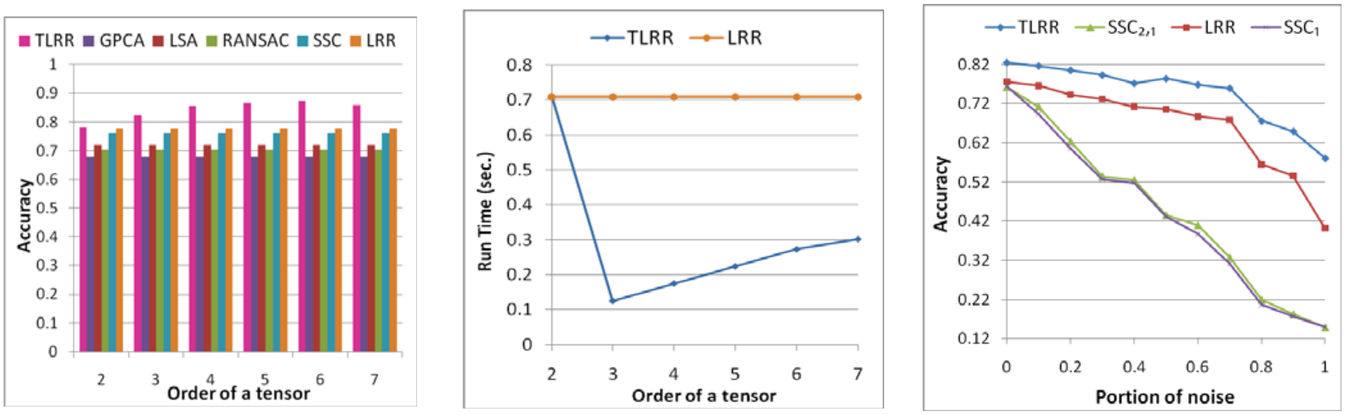
1) Performance with high order tensorial data: To show TLRR's advantage of handling high order tensorial data over other baseline methods, we create other 5 synthetic datasets from the above data $\mathbf{X}$ by reshaping it into higher j-mode tensor ($3 \leq j \leq 7$). A j-mode tensor $\mathcal{X}^j \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_{j-1} \times 5}$ is generated by rearranging the column vectors of $\mathbf{X}$ into higher spatial spaces $\mathbb{R}^{I_1 \times I_2 \cdots \times I_{j-1}}$ subject to $I_1 \times I_2 \cdots \times I_{j-1} = 64$. Since all other baseline methods conduct subspace clustering on an input matrix, i.e. a 2-mode tensor, we use $\mathbf{X}$ on all these baseline methods for the purpose of fair comparisons. Fig. 3(a) reports the results on all the baseline methods, including GPCA, Local Subspace Analysis (LSA) [25], RANSAC, SSC and LRR.

As we can see, our model TLRR performs much better than other methods in the higher mode of tensor. This observation suggests that incorporating data structure information into subspace clustering can boost clustering performance. While other methods' performances always stay still because these methods treat each sample independently, ignoring inherent data spatial structure information. As the order of tensor increases, the running time of TLRR is significantly reduced compared with LRR, as shown in Fig. 3(b), which suggests that the structure information has important impact on speeding up the subspace clustering process.

2) Performance with different portions of noisy samples: Consider the case when there exists noisy samples in the data. We randomly choose 0%, 10%,..., 100% of the samples of above $\mathbf{C}^i$ respectively, and add Gaussian noises $\mathcal{N}(\cdot|c_i, 0.3\Sigma^i)$ to these samples. Then a noisy data set $\mathbf{X}^{'}$ is generated by combining the corrupted $\mathbf{C}^i$ to one. For fair comparisons, we implement two versions of SSC, i.e., SSC_1 is a l_1-norm version and $SSC_{2,1}$ is a $l_{2,1}$-norm version. The performances on $SSC_{2,1}, SSC_1$, LRR and TLRR are listed in Fig. 3(c). Obviously, low-rank representation based subspace clustering methods TLRR and LRR maintain their accuracies even though 70% of samples are corrupted by noise. Moreover, TLRR and LRR significantly outperform both $SSC_{2,1}$ and SSC_1, as shown in Fig. 3(c), which suggests that low-rank representation is good at handling noisy data, while sparse representation is not because noise is unnecessary to decrease the sparsity. For low-rank based methods, LRR method is inferior to the structure based TLRR. This is mainly because TLRR integrates data spatial information into subspace clustering, it maintains good performance even 90% of data are corrupted .

B. Indianpines Dataset

We evaluate our model on the Indianpines dataset [26]. This dataset is gathered by AVIRIS sensor over the Indian Pines



(a)

(b)

(c)

Fig. 3: Comparison on the synthetic datasets. (a) Accuracy comparisons *w.r.t.* different orders of a tensor. (b) Run time comparisons *w.r.t.* different orders of a tensor. (c) Accuracy comparisons *w.r.t.* different potions of noisy samples.

TABLE I: Subspace clustering results on the Indianpines Dataset

	GPCA	LSA	RANSAC	SSC	LLR	TLRR
Accuracy	0.476	0.583	0.532	0.698	0.776	**0.786**
Time (min.)	6.87	177.84	5.90	745.73	380.07	**51.23**

test site in North-western Indiana, and consists of 145×145 pixels and 224 spectral reflectance bands in the wavelength range 0.4-$2.5 \times 10^{(-6)}$ meters. The whole data set is formed by 16 different classes having an available ground truth. In our experiments, 24 bands covering the region of water absorption are discarded. The task is to group pixels into clusters according to their spectral reflectance bands information. Table. I shows the results of all baseline methods on Indianpines.

Clearly our method TLRR has the highest accuracy among the other five baselines on this dataset. The advantage of TLRR mainly comes from its ability of incorporating 2 dimensional data structure information into the low-rank representation. G-PCA and RANSAC do not work well because their accuracies deteriorate quickly as the number of subspaces increases (i.e. 16 subspaces on Indianpine). The performance of LSA is marginally better than RANSAC and GPCA as LSA fits a subspace locally around each projected point, while GPCA uses the gradients of a polynomial that is globally fit to the projected data. However, LSA has the problem that selected neighbor is near the intersection of two subspaces, which may result in poor performance. Although TLRR costs more computational time than GPCA and RANSAC methods due to its optimization procedure needs more iterations to converge, the accuracy of TLRR is superior to them. The results regarding time cost on TLRR and LRR are consistent with Remark 2 in section IV-B, which shows that TLRR significantly reduces time cost by exploiting the Kronecker structure along each space dimension.

We further study the performance of our proposed method with different values of parameter λ. The parameter $\lambda > 0$ is used to balance the effects of the two parts in problem (6). Generally speaking, the choice of this parameter depends on the prior knowledge of the error level of data. When the errors are slight, a relatively large λ should be used; while the error are heavy, we should set a small value. The blue curve in Fig. 4 is the evaluation results on Indianpines data set. Wile λ ranges from 0.04 to 0.2, the clustering accuracy slightly varies from 80.34 % to 81.98 %. This phenomenon is mainly because TLRR employs LRR representation to explore data structure information. It has been proved that LRR works well on clean data (the indianpines is a clean data set), and there is an "invariance" in LRR that implies that it can be partially stable while λ is varying (For the proof of this property see Theorem 4.3 in [14]).

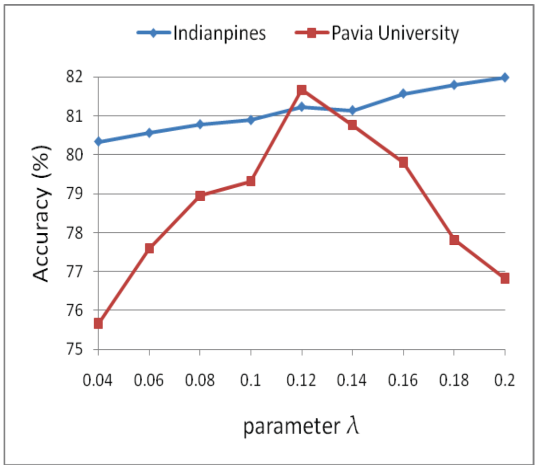


Fig. 4: The influences of the parameter λ of TLRR. These results are collected from the Indianpines data set and the Pavia University data set.

C. Slightly Corrupted Data set Pavia University

In this section, we evaluate TLRR on the Pavia University database [27]. The database is acquired by the ROSIS sensor with a geometric resolution of 1.3 meters, during a flight

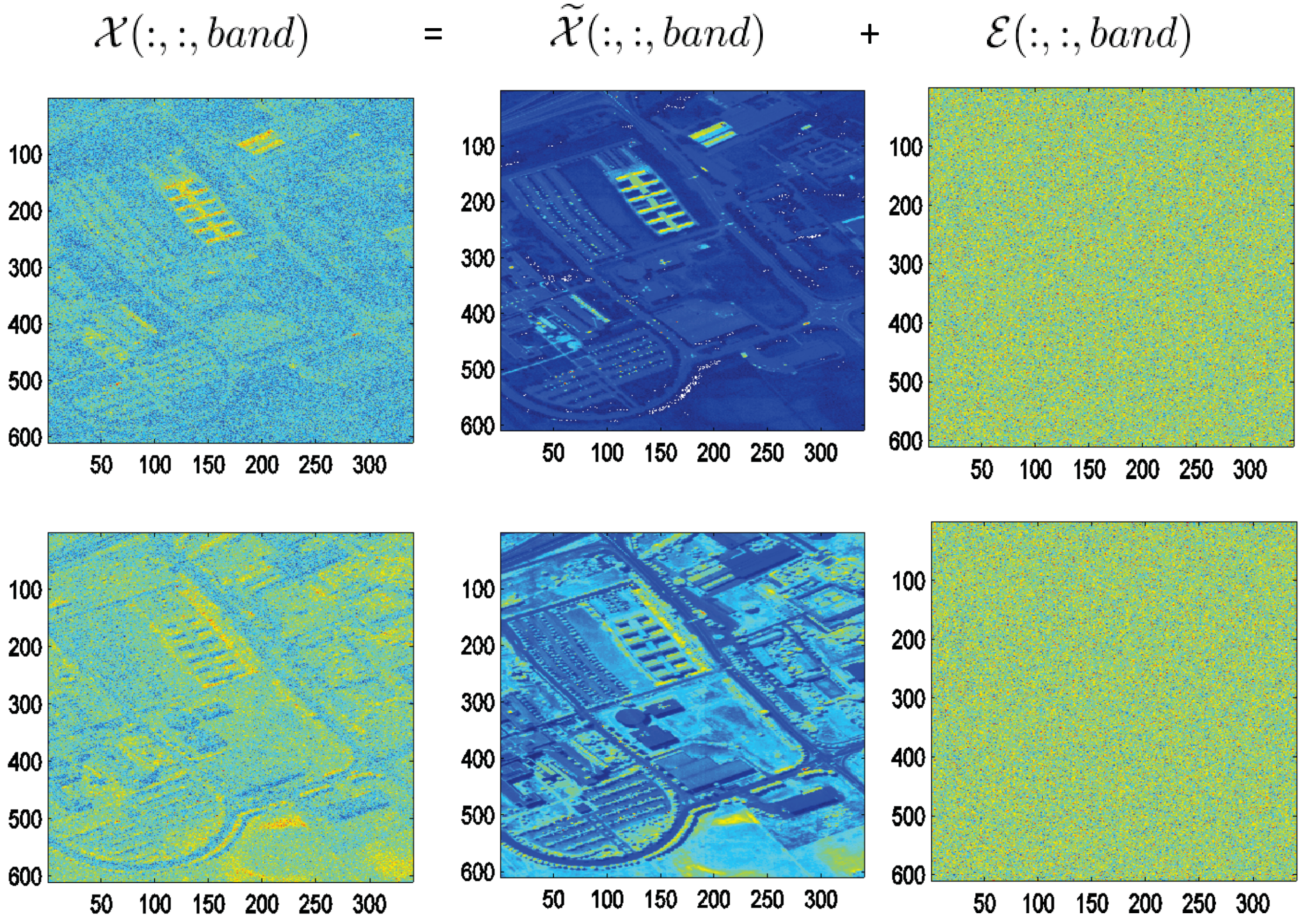


Fig. 5: Recovery Results of Pavia University for spectral band 7 (in the first row) and band 101 (in the second row).

campaign over Pavia, northern Italy. Pavia University consists of 610*340 pixels, each of which has 103 spectral bands covering 0.43 to 0.86 μm. The data set contains 9 different classes with available groundtruths. We examine the noise robustness of the proposed model by adding Guassian white noises with an intensity of 60 to the spectral band 7-10 and band 101-103.

Table II details the clustering performance on all the baselines. The performance of SSC is inferior to all LRR based methods, which shows that sparse representation is not good at handling corrupted data like LRR. Other baselines like GPCA, LSA and RANSAC do not work very well when the data is contaminated with noise. Our model TLRR performs best among all the methods on the corrupted data set. This is because TLRR explores data spatial correlation information with a low-rank representation, which guarantees accurately clustering data into different subgroups.

About the parameter λ, the red curve in Fig. 4 shows the performance of TLRRSC on Pavia University dataset, when the parameter λ varies from 0.04 to 0.2. Notice that TLRRSC is more sensitive to λ on this data set than on Indianpines. This is because the samples in Indianpines are clean, whereas Pavia University contains some corrupted information.

TABLE II: Subspace clustering results on the Pavia University Dataset

	Subspace clustering accuracy(%)					
	GPCA	LSA	RANSAC	SSC	LRR	TLRR
Mean	31.9	52.5	48.9	60.8	73.6	**76.6**
Std.	13.23	11.98	7.98	7.56	6.38	**4.12**
Max	61.2	69.0	72.0	79.8	82.8	**85.2**
Time (hr.)	1.04	27.01	0.89	98.18	47.02	**7.2**

To visualize TLRR's effectiveness in noise correction, we reconstruct the data $\widetilde{\mathcal{X}}$ with the learnt dictionary, sparse representation and the spatial factors. Fig. 5 shows the recovery results on spectral band 7 and band 101. Clearly, our model TLRRSC can extract the noise from the corrupted band, which proves that LRR is robust to noise.

VI. Conclusions

We propose a tensor based low-rank representation (TLRR) for subspace clustering in this paper. Unlike existing subspace clustering methods work on an unfolded matrix, TLRR builds a model on data original structure form (i.e. tensor) and explores data similarities along all spatial dimensions. On the synthetic higher mode tensorial datasets, we show that our model considering data structure keeps good performance. Moreover, the experimental results with different noise rates show our model maintains good performance on highly corrupted data. On the real-world dataset, our method shows promising results and significant computation gains. Moreover, our model is robust to noises, and capable of recovering corrupted data.

Acknowledgment

This work is supported by the Australian Research Council (ARC) through Discovery Project Grant DP130100364. Zhouchen Lin is supported by NSF China grants (Nos. 61272341, 61231002, 61121002)and MSRA.

References

[1] J. Shlens, "A tutorial on principal component analysis," in *Systems Neurobiology Laboratory, Salk Institute for Biological Studies*, 2005.

[2] K. Kanatani, "Motion segmentation by subspace separation and model selection," in *ICCV*, vol. 2, 2001, pp. 586–591.

[3] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *CVPR*, 2009, pp. 2790–2797.

[4] Y. Chen, A. Jalali, S. Sanghavi, and H. Xu, "Clustering partially observed graphs via convex optimization," in *ICML*, 2011.

[5] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[6] R. Vidal, "Subspace clustering," *Signal Processing Magazine, IEEE*, vol. 28, pp. 52–68, 2011.

[7] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *CVPR*, vol. 1, 2003, pp. 11–18.

[8] S. Rao, A. Yang, S. Sastry, and Y. Ma, "Robust algebraic segmentation of mixed rigid-body and planar motions from two views," *International Journal of Computer Vision*, vol. 88, no. 3, pp. 425–446, 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0314-1

[9] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, no. 2, pp. 443–482, Feb. 1999. [Online]. Available: http://dx.doi.org/10.1162/089976699300016728

[10] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[11] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1546–1562, 2007.

[12] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," in *CVPR*, vol. 1, 2003, pp. 621–628.

[13] M. Bouguessa, S. Wang, and Q. Jiang, "A k-means-based algorithm for projective clustering," in *ICPR*, vol. 1, 2006, pp. 888–891.

[14] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[15] G. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51(3), pp. 455–500, 2009.

[16] J. Martin and S. Marek, "A simple algorithm for nuclear norm regularized problems," in *ICML*, 2010, pp. 471–478. [Online]. Available: http://www.icml2010.org/papers/196.pdf

[17] R. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," *J. Mach. Learn. Res.*, vol. 11, pp. 2057–2078, 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1859920

[18] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Commun. ACM*, vol. 55, no. 6, pp. 111–119, 2012. [Online]. Available: http://doi.acm.org/10.1145/2184319.2184343

[19] M. Blondel, K. Seki, and K. Uehara, "Block coordinate descent algorithms for large-scale sparse multiclass classification," *Machine Learning*, vol. 93, no. 1, pp. 31–52, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10994-013-5367-2

[20] C. Ding, D. Zhou, X. He, and H. Zha, "R1-PCA: Rotational invariant l1-norm principal component analysis for robust subspace factorization," in *ICML*, 2006.

[21] Y. Shen, Z. Wen, and Y. Zhang, "Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optimization Methods and Software*, vol. (ahead-of-print), pp. 1–25, 2012.

[22] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *NIPS*, 2011.

[23] J. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010. [Online]. Available: http://dx.doi.org/10.1137/080738970

[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 1997.

[25] J. Yan and M. Pollefeys, "A general framework for motion segmentatise: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate." in *ECCV*, 2006.

[26] D. Landgrebe, "Multispectral data analysis: A signal theory perspective," Purdue Univ., West Lafayette, IN, Tech. Rep., 1998.

[27] L. Wei, S. Li, M. Zhang, Y. Wu, S. Su, and R. Ji, "Spectral-spatial classification of hyperspectral imagery based on random forests," in *Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service*, ser. ICIMCS '13. New York, NY, USA: ACM, 2013, pp. 163–168. [Online]. Available: http://doi.acm.org/10.1145/2499788.2499853