Robust structure low-rank representation in latent space

You, C., Palade, V. & Wu, X-J.

Author post-print (accepted) deposited by Coventry University's Repository

Original citation & hyperlink:
You, C, Palade, V & Wu, X-J 2019, 'Robust structure low-rank representation in latent space' Engineering Applications of Artificial Intelligence, vol. 77, pp. 117-124. https://dx.doi.org/10.1016/j.engappai.2018.09.008

DOI 10.1016/j.engappai.2018.09.008
ISSN 0952-1976

Publisher: Elsevier

NOTICE: this is the author's version of a work that was accepted for publication in *Engineering Applications of Artificial Intelligence.* Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in *Engineering Applications of Artificial Intelligence,* Vol 77, (2019) DOI: 10.1016/j.engappai.2018.09.008

© 2017, Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International http://creativecommons.org/licenses/by-nc-nd/4.0/

Copyright © and Moral Rights are retained by the author(s) and/ or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This item cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder(s). The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

This document is the author's post-print version, incorporating any revisions agreed during the peer-review process. Some differences between the published version and this version may remain and you are advised to consult the published version if you wish to cite from it.

Low-Rank Robust Structure Representation in Latent Space

Cong-Zhe You[1,2], Vasile Palade[2*], Xiao-Jun Wu[1]

[1] *School of IoT Engineering, Jiangnan University*
Wuxi, China
[2] *School of Computing, Electronics and Maths, Coventry University*
Coventry, United Kingdom

Abstract

Subspace clustering algorithms are usually used when processing high-dimensional data, such as in computer vision. This paper presents a robust low-rank representation (LRR) method that incorporates structure constraints and dimensionality reduction for subspace clustering. The existing LRR and its extensions use noise data as the dictionary, while this influences the final clustering results. The method proposed in this paper uses a discriminant dictionary for matrix recovery and completion in order to find the lowest rank representation of the data matrix. As the algorithm performs clustering operations in low-dimensional latent space, the computational efficiency of the algorithm is higher, which is also a major advantage of the proposed algorithm in this paper. A large number of experiments on standard datasets show the efficiency and effectiveness of the proposed method in subspace clustering problems.

Keywords: Low-Rank Representation; Subspace Clustering; Latent Space; Graph Regularization; Image Processing

[*]Corresponding author
Email addresses: youcongzhe@gmail.com (Cong-Zhe You[1,2]), vasile.palade@coventry.ac.uk (Vasile Palade[2]), wu_xiaojun@jiangnan.edu.cn (Xiao-Jun Wu[1])

1. Introduction

Many practical computer vision and image processing applications require high-dimensional data representation and processing. Many clustering techniques face the challenge of the curse of dimensionality [1] as well. In the actual environment, we often face high-dimensional data sets, such as face images under different lighting conditions, moving object trajectories, various other image data. In addition, for the analysis and processing of high-dimensional data, the data processing time and data storage requirements increases dramatically, while in the same time, the quality of data analysis and processing tends to decrease. Fortunately, however, a large number of studies have shown that the intrinsic dimension of these high-dimensional data is often much smaller than the actual dimension, in other words, these high-dimensional data samples can also be considered to be in a group of low-dimensional structures [2, 3].

For high-dimensional data, the feature dimensions of many data are usually irrelevant. For example, for a digital image, which often consists of billions of pixels, usually the main features of the image can be represented by only a few parameters. Therefore, many researchers point out that high-dimensional data can be approximated by a group of low-dimensional structures [4, 5], which is the focus of the so-called subspace algorithms. Subspace methods have been widely studied and applied in computer vision. These methods are also at the forefront of the research on high-dimensional data analysis and processing, especially for exploring and identifying the low-dimensional structures of high-dimensional data [4, 5, 6, 7]. At present, subspace clustering algorithms have been widely studied, and researchers have proposed a lot of related algorithms. These algorithms can be basically divided into four categories: subspace clustering algorithms based on iteration [8, 9]; subspace clustering based on spectral clustering algorithms [10, 11, 12]; subspace clustering based on statistical algorithms [13] and subspace clustering based on algebraic algorithms [14, 15]. In particular, subspace clustering algorithms based on low-rank and sparse representations [16, 17, 18, 4, 11] have received extensive attention and intensive

research in recent years. The usual flow of this kind of algorithms is to get the representation coefficients of the data matrix by using sparse or low rank representation, then construct the affinity matrix by using the representation coefficient matrix, and finally segment the affinity matrix by spectral clustering. One of the advantages of this method is robustness to noise and occlusion. In addition, some algorithms based on low-rank or sparse representation do not have to know the number and dimension of the subspace in advance.

The sparse subspace clustering algorithm (SSC) uses L1-norm minimization to obtain the sparse representation of the data matrix. This can be interpreted to mean that a data sample point can be obtained by linear representation of other data sample points in the same subspace. Then, we can use the obtained sparse coefficient matrix to construct the affinity matrix, and finally use the spectral clustering to segment the affinity matrix to get the final clustering result. The SSC algorithm has been widely successful in many fields. However, one of the main drawbacks of the SSC algorithm is that it can not capture the global characteristics of the data matrix, which leads to poor clustering performance of the SSC algorithm when the data sample matrix has noise or outlier data samples. In addition, the SSC algorithm needs to calculate the sparse representation of each data sample point, which means that the computational complexity of the SSC algorithm is relatively high.

Liu et al. [4] recently proposed a subspace clustering algorithm based on low rank representation (LRR). Like SSC, LRR assumes that a data sample point can be represented by linear representations of other data sample points in the same subspace. The LRR algorithm obtains the lowest rank representation of the matrix of the high-dimensional data. The LRR algorithm can capture the global structure of the data matrix. In addition, since the rank function minimization is NP-hard, the LRR algorithm uses the nuclear norm minimization to replace the rank function, this is also a common approximation. By now, many researches have improved the robustness of the LRR algorithm. Chen et al. [19] introduced a symmetric constraint in the low rank representation algorithm to extend the LRR algorithm, thus avoiding the subsequent steps of

symmetric operation on the affinity matrix. Zheng et al. [20] introduced a local constraint regularization term into the original LRR algorithm, which enables the algorithm to obtain both global and local structure information of the data matrix. Considering the subspace clustering problem in the presence of noise, [21] proposed a two-step procedure for robust subspace clustering. Li et.al [22] proposed a unified optimization framework for learning both the affinity and the segmentation. Because the norm of L0-norm is non-convex, therefore, in practical application, L1-norm or L2-norm is usually used for approximate substitution. However, these methods require certain assumptions. In practice, while, these assumptions are not fully guaranteed. [23] proposed an approximate L0-SSC method to tackle this problem. Unlike the traditional representation based subspace clustering methods, which transform the subspace clustering problem into a two-steps algorithm including building the affinity matrix and spectral clustering, [24, 25] directly learns the different subspaces indicator so that the low-rank based different groups are obtained clearly.

These algorithms have achieved great success in many areas, but the disadvantages are obvious. Because clustering is an unsupervised learning problem, there is no prior knowledge to use. Therefore, in the subspace clustering algorithm, the data matrix itself is usually used as the data dictionary. But, in practical applications, this leads to poor clustering performance, especially in certain specific cases, such as when the data matrix contains noise or data corruption. Therefore, it is hoped to recover a discriminant dictionary from the noisy data matrix and use it for sparse or low rank representation.

Computing sparse and low-rank representations of data matrices requires high computational cost, especially for data matrices with higher feature dimensions [11, 26, 27], which is also a disadvantage of subspace clustering algorithms based on sparse as well as low-rank representation. In order to solve this problem, traditional methods usually use dimensionality reduction algorithms to preprocess data before clustering. Dimensionality reduction algorithms, such as Random Projections (RP) algorithm and Principle Component Analysis (PCA), can effectively reduce data dimension. By using dimensionality reduction algo-

rithms, a better clustering result can also be obtained in the low dimensional latent space. To solve this problem, researchers have proposed some algorithms, that is, to compute the low rank representation or the sparse representation of the data in a latnet low-dimensional space [28, 29, 30]. However, these dimensional reduction algorithms are usually designed for supervised classification problems. For unsupervised clustering problems, these methods cannot be directly used for dimensionality reduction. Recently, Patel [31] et al. proposed an extended SSC algorithm, called latent space sparse subspace clustering (LS3C). For a given data matrix, the LS3C algorithm can simultaneously learn to get a low-dimensional space and a sparse representation coefficient matrix of the data matrix. And in the low-dimensional latent space, the sparse subspace clustering algorithm can get better clustering results.

Motivated by recent progresses in LRR techniques, in this paper, we further study the problem of subspace clustering in a new latent low-dimensional space by introducing the structure constraints and a discriminative dictionary. The algorithm proposed in this paper uses matrix recovery and completion techniques [32, 7] to obtain a discriminant dictionary from noisy data, and then it uses the dictionary to compute the low-rank representation of the data. Thus, our algorithm will be called the Robust Structure Low-Rank Representation in LATent space algorithms (LatRSLRR). The experimental results on the standard test databases also show that the proposed LatRSLRR algorithm is superior to the most state-of-the-art subspace clustering algorithms. The main contributions of this paper are summarized below:

1. In this paper, we propose a new latent space robust subspace segmentation method (LatRSLRR) based on low-rank and structure constraints.

2. The proposed algorithm uses matrix recovery and completion techniques to obtain a discriminant low-rank dictionary from noisy data, and then it uses the discriminant dictionary to compute the low-rank representation of the data matrix. The proposed algorithm has better robustness. Especially, it has good clustering performance for the data samples with noise or outlier.

The rest of this paper is organized as follows. Section 2 briefly reviews some

extensions of the LRR and LS3C algorithms. The robust structure low-rank representation in latent space (LatRSLRR) algorithm is proposed in Section 3. The experimental results are shown in Section 4, and finally we conclude this paper in Section 5.

2. Related work

In this section, we briefly introduce sparse representation, low-rank representation and the latent space sparse subspace clustering algorithms. For a given data matrix $X = [x_1, x_2, \cdots, x_n] \in R^{D \times N}$, its feature dimension is D, each data sample vector comes from a set of linear subspace $\{S_i\}_{i=1}^{m}$. The goal of subspace clustering is to divide the data sample vectors into corresponding m subspaces.

2.1. Sparse subspace clustering (SSC)

The Sparse Subspace Clustering (SSC) [11] is a representative subspace clustering algorithm. Its objective is to find the sparsest representation of a data matrix. The objective function is defined as:

$$\begin{aligned} &min_Z \ \|Z\|_0 \\ &s.t.\ X = XZ \ and \ diag(Z) = 0 \end{aligned} \tag{1}$$

Among them, Z is the sparse representation coefficient matrix. $\|\cdot\|_0$ represents the l_0 -norm of the vector, and its value represents the number of non-zero elements of the vector. $diag(Z) \in R^N$ represents the diagonal elements of a matrix Z. The solution Z^* to the above optimization problem is a sparse representation of matrix X. Because the l_0 -norm optimization problem is non-convex and NP-hard, the l_1 -norm is usually used to replace the l_0 -norm.

$$\begin{aligned} &min_Z \ \|Z\|_1 \\ &s.t.\ X = XZ \ and \ diag(Z) = 0 \end{aligned} \tag{2}$$

where $\|\cdot\|_1$ denotes the l_1-norm of a matrix.

When the data sample matrix contains noise or outliers, the optimization objective function of the SSC algorithm is defined as:

$$\begin{aligned} &min_Z \ \|Z\|_1 + \alpha \|E\|_1 + \frac{\beta}{2} \|C\|_F^2 \\ &s.t. \ X = XZ + E + C \ and \ diag(Z) = 0 \end{aligned} \tag{3}$$

Among them, E represents the sparse outliers and C represents the noise matrix. The non-negative parameters $\alpha > 0$ and $\beta > 0$ regulate the weights of the three items in the above objective optimization function. After the coefficient representation matrix Z is obtained, the affinity matrix $|Z| + |Z|^T$ can be constructed by the coefficient representation matrix Z in the subsequent processing. Finally, the affinity matrix can be partitioned with a spectral clustering algorithm to get the final segmentation results.

2.2. Low-Rank Representation (LRR)

Different from SSC, LRR finds the lowest-rank representation. The objection function of the LRR is defined as follows:

$$\begin{aligned} &min_Z \ rank(Z) \\ &s.t. \ X = AZ \end{aligned} \tag{4}$$

where, A represents the data dictionary. As the optimization of the rank function is a NP-hard problem, in practice, we usually use the nuclear norm for approximate substitution. Then the objective function is proposed as:

$$\begin{aligned} &min_Z \ \|Z\|_* \\ &s.t. \ X = AZ \end{aligned} \tag{5}$$

where $\|Z\|_*$ is the nuclear norm, defined as the sum of all singular values of Z, which is the convex envelope of the rank function. Considering the fact that

samples are usually noisy or even grossly corrupted, a more reasonable objective for LRR can be expressed as:

$$\begin{aligned} min_{Z,E} \ \ & \|Z\|_* + \lambda \|E\|_{2,1} \\ & s.t. \ X = AZ + E \end{aligned} \tag{6}$$

where the $l_{2,1}$-norm is defined as $\|E\|_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{d} e_{ij}^2}$ and the parameter λ is used to balance the effect of the low-rank term and the error term.

2.3. Latent space sparse subspace clustering (LS3C)

Traditional SSC and LRR algorithms calculate the sparse representation and low rank representation of each data sample vector in the original space. LS3C algorithm hopes to find a low-dimensional latent space firstly, and then calculate the sparse representation of data matrix in this low-dimensional space, and get the final clustering segmentation results. The objective function of the LS3C algorithm is defined as follows:

$$\begin{aligned} min_{P,Z} \ & \lambda_1 \left\|P^T X - P^T X Z\right\|_F^2 + \lambda_2 \left\|X - P P^T X\right\|_F^2 + \|Z\|_1 \\ & s.t. \ P^T P = I \ and \ diag(Z) = 0 \end{aligned} \tag{7}$$

Among them, λ_1 and λ_2 are two non-negative parameters, $P \in R^{D \times d}$ is the projection matrix, mapping data from high-dimensional space to low-dimensional space. The first and third terms of the objective function (7) describe the objectives of the LS3C algorithm. The second term of the objective function guarantees that the reconstruction of data from the original high-dimensional space to the low-dimensional space will not lose too much information. The objective function can solve P and Z by an iterative method. As the author claims, the LS3C algorithm can calculate the representation coefficients of each data sample in low-dimensional latent space, so the LS3C algorithm is more effective than the SSC algorithm.

3. Low-Rank Robust Structure Representation in Latent Space

In this section, we introduce the proposed robust structure low-rank representation in latent space (LatRSLRR) method. LatRSLRR learns a structured low-rank representation in a latent low-dimensional space, and can get a better clustering performance.

3.1. Motivation

One of the main drawbacks of the LS3C algorithm is that it uses the l_1-norm to compute the sparse representation of the matrix. However, the l_1-norm can not capture the global structure information of the data. However, the LRR algorithm and its extension are proved to be able to capture the global structure information of the data matrix.

The LRR algorithm can capture the global structure of the data matrix, but the local structure information of the data matrix is also very useful for the subspace clustering problem. Therefore, we hope that the coefficient matrix Z obtained by the optimization problem (6) can reflect the local structure information of the data at the same time. If the affinity matrix G constructed by $G = (|Z| + \left|Z^T\right|)/2$ can reflect the similarity between data samples, the following minimization optimization objective function needs to be satisfied:

$$\begin{aligned} \sum_{ij} |G_{ij}|\, d(x_i, x_j) &= 1/2 \sum_{ij} (|Z_{ij}| + \left|Z_{ij}^T\right|) d(x_i, x_j) \\ &= \sum_{ij} |Z_{ij}|\, d(x_i, x_j) = \|Z \odot M\|_1 \end{aligned} \tag{8}$$

where $M \in R^{n \times n}$, $M_{ij} = d(x_i, x_j)$. $d(x_i, x_j)$ is a kind of distance between x_i and x_j , $\odot$ denotes the Hadamard product. In this paper, we define

$$d_{ij} = 1 - exp(-\frac{1 - \left|x_i^{*T} x_j^*\right|}{\sigma}) \tag{9}$$

where x_i^* and x_j^* are the normalized data points of x_i and x_j , respectively, and σ is the average of the elements of matrix B (where B is defined as $B_{ij} =$

$1 - \left|x_i^{*T} x_j^*\right|$). We add Eq. (9) into the LRR objective as a l_1-regularizer, which can also help to keep Z to be sparse as much as possible.

The LRR and its extensions use the data matrix itself as the dictionary. However, in the practical data matrix, noise and data corruption is a very common phenomenon, and we can not expect tht the actual high-dimensional data are obtained under good control. Therefore, when the original data matrix containing noise is used as a dictionary, the clustering performance of the algorithm is often severely limited. Therefore, it is very important to learn a discriminant dictionary from a noisy data matrix. Many papers have also studied this problem in depth [32, 33, 34].

In practical applications, it is difficult to find a suitable method to remove all the noise in the data, because the noise comes in many forms. Different matrix recovery techniques are also suitable for different types of noise. According to the latest developments in low-rank matrix recovery and completion methods, we hope to find a more suitable data dictionary, such as a discriminatory low-rank dictionary, instead of learning the low-rank representation by using the noise-contained data matrix itself. When the data sample is only slightly corrupted by Gaussian noise with small variance, the PCA algorithm can determine the best low rank approximation. However, in practice, especially in data analysis and image processing, large errors are common, which seriously limits the application of the PCA algorithm. In recent studies, the RPCA algorithm [7] has been proposed, which can recover the discriminant low-rank dictionary from the corrupted data matrix. Its objective function is defined as follows:

$$\begin{aligned} min_{A,E} \ \ & \|A\|_* + \lambda \|E\|_1 \\ & s.t. \ X = A + E \end{aligned} \tag{10}$$

where $\|\cdot\|_*$ denotes the nuclear norm of a matrix, and $\|\cdot\|_1$ denotes the l_1 -norm of a matrix. This problem can be solved by the inexact Augmented Lagrange Multiplier (ALM) method [35]. By using the RPCA algorithm, we recover a discriminant low rank matrix A from the noisy data matrix, and use

this matrix as a data dictionary to learn the low rank representation.

In addition, we also use the recovered data matrix to construct the graph Laplacian matrix L. The weight matrix is defined as follows:

$W_{ij} = \|a_i - a_j\|^2$

where a_i and a_j are samples of the recovered data. Note that $L = D - W$ is the Laplacian matrix of the graph, where D is a diagonal matrix with diagonal entries $D_{ii} = \sum_j W_{ij}$.

Through the above analysis, then we have the so called latent space robust structure low-rank representation for subspace clustering, and the objective function can be expressed as:

$$\begin{aligned} min_{P,Z} \ \ & \|Z\|_* + \beta \|W \odot Z\|_1 + \frac{\alpha}{2} tr(Z^T L Z) \\ & + \lambda(\|PX - PAZ\|_F^2 + \left\|A - P^T P A\right\|_F^2) \\ & s.t.\ PP^T = I \ , \ X = AZ + E \end{aligned} \tag{11}$$

where $tr(\cdot)$ is the trace of a matrix. P is the projection transformation. The first term of (11) promotes the low-rankness of the data, the second term is a l_1-regularizer which can help to keep Z to be sparse as much as possible. The third term is the Laplacian graph. The last two term ensures that the projection does not loose too much information available in the original domain. α, β and λ are non-negative parameters to control the influence of each term.

3.2. Solution to the above optimization problem

For the basis projection transformation P, we have the following proposition.

Proposition 1: There exists an optimal solution P^* to (11) that has the following form:

$$P^* = \Phi^T A^T \tag{12}$$

for some $\Phi \in R^{N \times d}$, where N is the number of data sample and d is the dimension of the latent output space.

It should be pointed out that proposition 1 has been applied in subspace clustering and dictionary learning in [14]. With proposition 1, by substituting (12) in (11), the objective function can be represented as:

$$\begin{aligned} min_{\Phi,Z}\ \|Z\|_* + \beta\|W \odot Z\|_1 + \frac{\alpha}{2}tr(Z^T L Z) + \\ \lambda(\left\|\Phi^T A^T X - \Phi^T A^T A Z\right\|_F^2 + \left\|A - A\Phi\Phi^T A^T A\right\|_F^2) \end{aligned} \tag{13}$$

Let $K_1 = A^T X$ and $K = A^T A$, then the proposed method (11) can be illustrated as follows

$$\begin{aligned} min_{\Phi,Z}\ \|Z\|_* + \beta\|W \odot Z\|_1 + \frac{\alpha}{2}tr(Z^T L Z) + \\ \lambda(\left\|\Phi^T K_1 - \Phi^T K Z\right\|_F^2 + \left\|A - A\Phi\Phi^T K\right\|_F^2) \end{aligned} \tag{14}$$

s.t. $\Phi^T K \Phi = I$

It can be seen that the optimization problem (14) contains two variables. In order to solve this minimization optimization problem, we adopt the alternating optimization strategy, that is, fixing one variable to solve another variable. Therefore, the optimization problem is divided into two steps.

3.2.1. Update Φ with fixed Z

For solving Φ , the minimization problem (14) after fixing Z can be written as

$$min \left\|\Phi^T K_1 - \Phi^T K Z\right\|_F^2 + \left\|A - A\Phi\Phi^T K\right\|_F^2 \tag{15}$$

s.t. $\Phi^T K \Phi = I$

This cost function can be expanded as follows:

$$tr((K^{-1}K_1 - Z)(K^{-1}K_1 - Z)^T K^T Q^T K) + tr((K - 2K^T Q^T K + K^T Q^T K Q K)) \tag{16}$$

where $Q = \Phi\Phi^T \in R^{N\times N}$. The constraint $\Phi^T K\Phi = I$ leads to the new constraint $\Phi\Phi^T K\Phi\Phi^T = QKQ^T = \Phi\Phi^T = Q$. The objective function (16) can be further simplified as:

$$tr(((K^{-1}K_1 - Z)(K^{-1}K_1 - Z)^T - I)K^T Q^T K) \tag{17}$$

where we have made use of the quality constraint and used the fact that trace(K) is constant. Using the eigen decomposition of $K = VSV^T$, we get

$K^T Q^T K = VS^{\frac{1}{2}}MM^T S^{\frac{1}{2}}V^T$

where $M = S^{\frac{1}{2}}V^T\Phi$. As a result, (17) can be rewritten as:

$tr(M^T S^{\frac{1}{2}}V^T((K^{-1}K_1 - Z)(K^{-1}K_1 - Z)^T - I)VS^{\frac{1}{2}}M) = tr(M^T\Theta M)$

Where $\Theta = S^{\frac{1}{2}}V^T((K^{-1}K_1-Z)(K^{-1}K_1-Z)^T-I)VS^{\frac{1}{2}}$, $M^T M = \Phi^T VSV^T\Phi = \Phi^T K\Phi = I$

We arrive at the following optimization problem, which is equivalent to (15)

$$\begin{aligned} M^* &= min\ tr(M^T\Theta M) \\ &s.t.\ M^T M = I \end{aligned} \tag{18}$$

Once the optimal M^* is found, the optimal Φ^* can be recovered as

$$\Phi^* = VS^{-\frac{1}{2}}M^* \tag{19}$$

3.2.2. Update Z with fixed Φ

Once Φ is obtained, we can compute Z. Let $Y = \Phi^T K_1$, $B = \Phi^T K$, we have to solve the following problem to obtain Z.

$$min_Z\ \|Z\|_* + \beta\|W \odot Z\|_1 + \frac{\alpha}{2}tr(Z^T LZ) + \lambda\|Y - BZ\|_F^2 \tag{20}$$

This problem can be solved by using the Augmented Lagrange Multiplier (ALM) method. We introduce two auxiliary variables J and L in order to make

the objective function separable, and convert the above problem to the following equivalent problem:

$$\begin{aligned} min_{Z,J,L} \ \ \|J\|_* + \beta \|W \odot L\|_1 + \frac{\alpha}{2} tr(J^T L J) + \lambda \|Y - BZ\|_F^2 \\ s.t. \ Z = J \ , \ Z = L \end{aligned} \tag{21}$$

Then, the augmented Lagrangian function of Eq. (21) is

$$\begin{aligned} min_{Z,J,L} \ \ \|J\|_* + \beta \|W \odot L\|_1 + \frac{\alpha}{2} tr(J^T L J) \\ +\lambda \|Y - BZ\|_F^2 + \langle Y_1, Z - J \rangle + \langle Y_2, Z - L \rangle + \\ \frac{\mu}{2}(\|Z - J\|_F^2 + \|Z - L\|_F^2) \end{aligned} \tag{22}$$

Among them, Y_1 and Y_2 are Lagrange multipliers, and the non-negative parameter $\mu > 0$ is the penalty parameter. In order to solve this optimization problem, we adopt the alternative optimization method, that is, by fixing other variables, we get the variable J, L, Z in sequence. The detailed updating methods for variable J, L, Z are as follows.

A. Update J with other variables fixed. When we update J, we drop the irrelevant terms w.r.t to J in (22), then in the k-th iteration, we have:

$$J^{k+1} = arg \ min_{J_k} \ \frac{1}{\mu_k} \|J_k\|_* + \frac{\alpha}{2\mu_k} tr(J_k^T L J_k) + \frac{1}{2} \left\| J_k - (Z_k + \frac{Y_1^k}{\mu_k}) \right\|_F^2 \tag{23}$$

which does not have a closed-form solution. By the spirit of LADMAP [35], we denote the smooth component of above equation by

$$q(J, Z_k, Y_1^k) = \frac{\alpha}{2} tr(J^T L J) + \frac{\mu_k}{2} \left\| J - (Z_k + \frac{Y_1^k}{\mu_k}) \right\|_F^2 \tag{24}$$

Then according to LADMAP, minimizing (23) can be replaced by solving the following problem:

$$min_J \|J\|_* + \langle \nabla_J q(J_k), Z_k - J \rangle + \frac{\eta_1}{2} \|Z_k - J\|_F^2 \tag{25}$$

Where $q(J, Z_k, Y_1^k)$ is approximated by its linearization $\langle \nabla_J q(J_k), Z_k - J \rangle$ at J_k plus a proximal term $\frac{\eta_1}{2} \|Z_k - J\|_F^2$, and $\nabla_J q(J_k)$ is the gradient of q w.r.t. Z. As long as $\eta_1 > \alpha \|L\|_2$, where $\alpha \|\cdot\|_2$ is the spectral norm of a matrix, i.e., the largest singular value, the above replacement is valid. Then (25) can be reformulated as:

$$\begin{aligned} J_{k+1} &= argmin_{J_k} \|J_k\|_* + \langle \nabla_{J_k} q(J_k), Z_k - J_k \rangle + \frac{\eta_1}{2} \|Z_k - J_k\|_F^2 \\ &= argmin_{J_k} \frac{1}{\eta_1} \|J_k\|_* + \frac{1}{2} \left\| J_k - (Z_k + \frac{\nabla_{J_k} q(J_k)}{\eta 1}) \right\|_F^2 \end{aligned} \tag{26}$$

Suppose $U_k S_k V_k^T$ is the SVD of the matrix $Z_k + \frac{\nabla_{J_k} q(J_k)}{\eta 1}$ and $S_k = diag(\{s_i^k\}_{1 \le i \le r})$ (r is the rank of $Z_k + \frac{\nabla_{J_k} q(J_k)}{\eta 1}$), then $J_{k+1} = U_k \Theta_{\frac{1}{\eta_1}}(S_k) V_k^T$. Here Θ is the singular value thresholding operator [36].

B. Update L with other variables fixed. By ignoring the terms independent of L, we have:

$$\begin{aligned} L_{k+1} &= argmin_{L_k} \ \beta \|W \odot L_k\|_1 + \langle Y_2, Z_k - L_k \rangle + \frac{\mu_k}{2} (\|Z_k - L_k\|_F^2) \\ &= arg\ min_{L_k} \ \frac{\beta}{\mu_k} \|W \odot L_k\|_1 + \frac{1}{2} \left\| L_k - (Z_k + \frac{Y_2^k}{\mu_k}) \right\|_F^2 \end{aligned} \tag{27}$$

Then, the solution to Eq. (27) satisfies $[L_{k+1}]_{ij} = \Psi_{\varepsilon_{ij}}(\left[Z_k + \frac{Y_2^k}{\mu_k} \right]_{ij})$, where $\Psi_\varepsilon(x) = max(x - \varepsilon, 0) + min(x - \varepsilon, 0)$ and $\varepsilon_{ij} = \frac{\beta}{\mu_k} [W]_{ij}$.

C. Update Z with other variables fixed. Similar to the previous method, we collect the related terms of Z in Eq. (22), then we have:

$$\begin{aligned} Z_{k+1} &= arg\ min_{Z_k} \lambda \|X - BZ_k\|_F^2 + \left\langle Y_1^k, Z_k - J_k \right\rangle + \left\langle Y_2^k, Z_k - L_k \right\rangle + \frac{\mu}{2} (\|Z_k - J_k\|_F^2 + \|Z_k - L_k\|_F^2) \\ &= arg\ min_{Z_k} \ \lambda \|X - BZ_k\|_F^2 + \frac{\mu_k}{2} \left\| Z_k - J_k + \frac{Y_1^k}{\mu_k} \right\|_F^2 + \frac{\mu_k}{2} \left\| Z_k - L_k + \frac{Y_2^k}{\mu_k} \right\|_F^2 \end{aligned} \tag{28}$$

Therefore,

$$Z_{k+1} = \frac{1}{2} (\lambda X^T B + \mu_k I)^{-1} (2\lambda X^T B + \mu_k (J_k - \frac{Y_1^k}{\mu_k} + L_k - \frac{Y_2^k}{\mu_k})) \tag{29}$$

D. Update Y_1, Y_2 and μ

$$Y_1^{k+1} = Y_1^k + \mu_k(Z_{k+1} - J_{k+1}) \tag{30}$$

$$Y_2^{k+1} = Y_2^k + \mu_k(Z_{k+1} - L_{k+1}) \tag{31}$$

$$\mu_{k+1} = min(\mu_{max}, \rho\mu_k) \tag{32}$$

Where μ_{max} and ρ are two positive parameters, k denotes the number of iteration.

3.3. Convergence and complexity analysis

The criterion function $J = \|Z\|_* + \beta \|W \odot Z\|_1 + \frac{\alpha}{2} tr(Z^T L Z) + \lambda(\|PX - PAZ\|_F^2 + \left\|A - P^T PA\right\|_F^2)$ converges to a minimum. Firstly, according to the definition of J, we have $J > 0$ for any P, Z. Secondly, based on the algorithm, once Z_k is obtained, we have $P_{k+1} = arg\ min_P J(P_k, Z_k)$. Therefore, $J(P_{k+1}, Z_k) \leq J(P_k, Z_k)$. After P_{k+1} is obtained, we also have $Z_{k+1} = arg\ min_Z J(P_{k+1}, Z_k)$, so $J(P_{k+1}, Z_{k+1}) \leq J(P_{k+1}, Z_k)$. Finally, $J(P_{k+1}, Z_{k+1}) \leq J(P_k, Z_k)$, namely $J_{k+1} \leq J_k$. Hence, we can conclude that the proposed algorithm is convergent.

In general, we can assume that the dimension of data samples is larger than the number of data samples. That is, $m > n$. From the optimization process of the proposed algorithm, we can see that the computational complexity of the algorithm mainly comes from the eigen-decomposition of formula (20) and the ALM algorithm. The time complexity of eigen decomposition is $O(m^3)$, and the main calculation of ALM algorithm is the SVD decomposition. The complexity of each iteration algorithm is $O(m^3)$. If the algorithm converges within the iteration steps of its outer loop, the total computational complexity is up to $O(Tm^3 + Ttm^3)$, where t represents the number of iterations within the ALM

algorithm. If the data sample size and T are large, the algorithm complexity is very large. Fortunately, the outer iteration converges very fast, so the total computational complexity of the proposed algorithm is the same as that of the traditional PCA algorithm. In addition, when the dimension of the data sample is very large, we can also use the KL transformation to calculate the standard eigenvectors of the data matrix, which will greatly reduce the computational complexity of the algorithm.

4. Experiments

In this section, we experimented with the proposed LatRSLRR algorithm on three public data sets: the extended YaleB data set, the AR data set and the MINIST data set. The specific experimental settings and results are shown below. We compare the proposed LatRSLRR method with the graph-based clustering approaches of S3C [22], FeaMAC [37] , LRRSC [19], LRR [4], SSC [11], local subspace affinity (LSA) [38], and spectral clustering (SC) [39], which provide a good baseline for evaluation.

In the experiment, we use clustering error rate to evaluate the performance of the algorithm, and the clustering error rate is defined as

$$Error = \frac{N_{error}}{N_{total}} \tag{33}$$

where N_{error} represents the number of misclassified samples and N_{total} is the total number of samples. Smaller subspace clustering errors indicate better clustering performance.

4.1. Clustering experiments using the extended Yale B database

The extended YaleB dataset contains 2414 face images of 38 people. Each person probably collected 64 images, which were photographed under different poses and illumination conditions. In this experiment, for the sake of computational efficiency, we use the images of the first 10 people in the data set as the

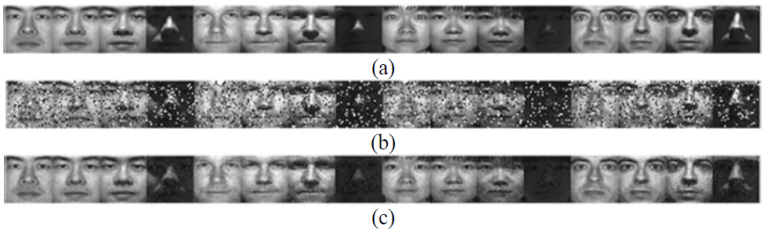

Figure 1: Representative examples in the extended Yale database B: (a) sample images under different illumination conditions, (b) sample images with random pixel corruptions, (c) sample images of a discriminative low-rank dictionary.

test data set, and manually cropped and normalized each image into 32×32 pixels.

Firstly, we validate the algorithm on the original data set, and the test data set does not receive any corruption. Some example images of the extended YaleB dataset is shown in Fig. 1a. Next, we consider the clustering performance of the algorithm when the data set is contaminated by noise. In order to simulate the noise, we use random pixel corruptions, in which the original image matrix is replaced by a random point with a uniformly distributed value in the range [0, 1]. The damage ratio of the face image matrix is from 5 to 20% . Figure 1b gives some sample examples of randomly damaged image matrices. Fig. 1C shows the face image matrix recovered from the randomly corrupted image data matrix by the RPCA algorithm, which is used as a data dictionary for learning the low rank representation.

There are three parameters affecting the performance of the LatRSLRR. In the experiments, the dataset images are without any artificial corruption, and we find that the consistent result is insensitive to the varying α values. This can be seen from the Fig. 2(a). In this subsection, we focus on the influence of β and λ . We set $\lambda = [0.01, 0.1, 0.3, 0.5, 1, 3, 5, 8, 10]$, $\beta = [1e^{-4}, 1e^{-3}, 1e^{-2}, 1e^{-1}, 1, 5, 10]$, and record the segmentation errors of the LatRSLRR on different pairs (λ, β). Fig. 2(b) shows the experimental results. It is obvious that the performance

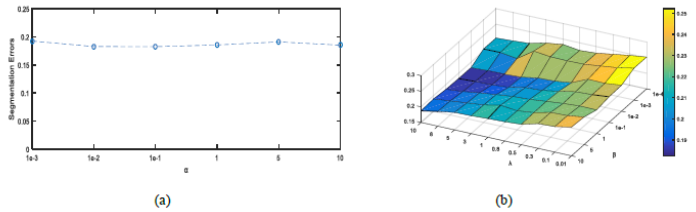

Figure 2: Segmentation errors of the LatRSLRR versus the variation of parameters

of the proposed LatRSLRR algorithm is stable when λ and β vary in a relatively large range. These experiments also show that the algorithm proposed in this paper is very effective for subspace clustering problems.

Next, we considered artificial occlusion, and the parameters of LatRSLRR were empirically set to $\alpha = 0.1$, $\beta = 0.01$ and $\lambda = 10$ for this experiment. We execute each clustering algorithm 10 times, and report the mean clustering error and standard deviation in Table 1. The results show that LatRSLRR consistently outperformed all the other methods (by about 2%), particularly for larger percentages of corrupted pixels. As the percentage of corrupted pixels increased, LatRSLRR retained this advantage over the other algorithms. These results clearly imply that LatRSLRR is much more robust than the other algorithms.

Table 1. Clustering error (%) of different algorithms on

the first ten classes of the extended Yale database B contaminated by random pixel corruptions

Figure 3: Example images of multiple individuals from the AR database

Ration(%)	Error	LatRSLRR	FeaMAC	S3C	LRRSC	LRR	SSC	LSA	SC
0	Mean	2.54	6.07	18.25	4.53	20.62	37.03	56.09	57.03
	Std	1.13	0.95	0.94	0.29	1.12	3.04	2.53	2.74
5	Mean	8.62	10.57	15.56	12.97	19.83	38.67	62.34	57.19
	Std	1.25	0.73	2.78	0.40	3.83	4.28	2.64	3.05
10	Mean	16.34	17.57	16.31	18.28	25.78	40.66	62.81	57.97
	Std	1.54	1.08	0.82	0.37	1.21	5.08	2.71	3.58
15	Mean	18.54	18.85	17.34	19.69	26.00	42.11	63.12	58.12
	Std	2.27	1.96	2.67	2.87	3.91	1.39	2.26	4.70
20	Mean	20.56	20.81	19.42	21.87	26.71	43.37	63.91	59.28
	Std	1.83	2.09	1.55	4.04	0.88	1.85	3.67	2.40

4.2. Clustering experiments using the AR database

The AR dataset contains more than 4000 frontal face images of 126 people. These images are captured under different illumination conditions, facial expressions and facial occlusion (sunglasses and scarves). In the experiments, these face images have been cropped to 165×120 pixel gray images. In this experiment, in order to calculate efficiently, we chose the face images from 5 male and 5 female to form a test data set. Some of the sample images are shown in Figure 3.

In the experiments, we considered artificial occlusion, the parameters of LatRSLRR were empirically set to $\alpha = 1$, $\beta = 0.01$ and $\lambda = 10$ for this experiment. Table 2 shows the results for all six algorithms. LatRSLRR outperformed other methods because it simultaneously considers the intrinsic local and global structure of the high-dimensional data using the low-rank criterion with graph regularization.

Table 2. Clustering error (%) of different algorithms on the ten classes of the AR

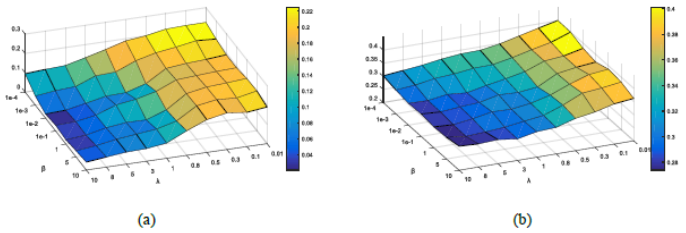

Figure 4: Segmentation errors of LatRSLRR versus the variation of parameters (a) AR database (b) MNIST database

database contaminated by random pixel corruptions

Ration(%)	Error	LatRSLRR	FeaMAC	S3C	LRRSC	LRR	SSC	LSA	SC
0	Mean	2.08	2.23	12.62	2.86	2.86	23.57	37.86	20.71
	Std	0	0	0.58	0	0	1.09	1.27	0.65
5	Mean	2.08	2.23	12.85	2.86	2.86	26.93	44.05	21.50
	Std	0	0	0.94	0	0.30	1.58	0.65	0.41
10	Mean	2.08	2.31	13.54	2.86	2.86	27.00	44.71	26.71
	Std	0	0	1.25	0	0	1.05	2.02	3.89
15	Mean	2.29	2.31	14.05	2.86	2.86	27.45	44.75	30.86
	Std	0	0	1.19	0	0	1.27	2.60	3.54
20	Mean	2.34	2.39	15.71	2.86	3.07	29.36	43.64	33.57
	Std	0.35	0	1.26	0	0.35	3.54	0.75	4.44

4.3. Clustering experiments using the MNIST database

In order to better verify the robustness and general adaptability of the LatRSLRR algorithm proposed in this paper, we use the MNIST handwritten digit set to carry out experimental testing. The data consist of 10 class images of handwritten difits, 0 to 9, with 60000 training images and 10000 test images. Each image is 28×28 pixels in size. In this experiment, we select 1000 images from the training data set as the experimental data set for each number in order to calculate the experimental efficiency. Figure 4 shows some examples of digits 0,1,3 and 8.

In the experiments, we considered artificial occlusion, the parameters of

Figure 5: Example images of the digits from the MNIST database

LatRSLRR were empirically set to $\alpha = 0.1$, $\beta = 0.01$ and $\lambda = 10$. Table 3 lists the clustering results for LatRSLRR and the five competing methods. From these results, we can observe that LatRSLRR outperformed the other algorithms.

Table 3. Clustering error (%) of different algorithms on the MNIST database contaminated by random pixel corruptions

Ration(%)	Error	LatRSLRR	FeaMAC	S3C	LRRSC	LRR	SSC	LSA	SC
0	Mean	28.30	30.85	31.28	33.50	47.50	32.80	36.00	43.70
	Std	1.91	2.53	2.15	0.71	2.73	3.17	2.58	1.74
5	Mean	30.39	31.18	33.56	35.78	48.84	34.97	42.45	45.49
	Std	1.74	1.86	2.18	0.85	1.60	3.68	3.00	1.69
10	Mean	30.47	31.87	34.82	36.12	50.13	35.08	44.27	46.86
	Std	1.98	1.63	1.93	0.79	2.81	3.30	1.91	1.58
15	Mean	33.62	32.36	35.29	36.71	51.13	36.20	47.23	48.02
	Std	1.62	1.90	2.05	0.75	3.10	2.52	1.29	2.02
20	Mean	34.87	33.72	37.08	37.56	51.90	38.32	51.27	48.24
	Std	2.01	2.18	1.80	1.47	3.03	3.42	3.32	2.05

5. Conclusions

The recent technological developments have brought a great deal of data, especially high-dimensional data, which puts forward a higher test to the traditional data analysis and processing algorithms. It is the unremitting pursuit of researchers to study and mine the intrinsic structural characteristics of these high-dimensional data and to analyze and process the data accurately. In this paper, we proposed a new subspace segmentation algorithm, termed the robust structure low-rank representation in latent space (LatRSLRR), to reveal the structures of high-dimensional datasets. We show that the LatRSLRR can

be explained in terms of a robust low-rank representation method, so that its good performance can be guaranteed in theory. Various experiments for subspace segmentation have proven that the LatRSLRR algorithm achieved significantly better results than the competitive state-of-art subspace segmentation algorithms.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 61373055) and the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20130093110009).

References

[1] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, Acm Sigkdd Explorations Newsletter 6 (1) (2004) 90–105.

[2] R. Basri, D. W. Jacobs, Lambertian reflectance and linear subspaces, IEEE transactions on pattern analysis and machine intelligence 25 (2) (2003) 218–233.

[3] K.-C. Lee, J. Ho, D. J. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Transactions on pattern analysis and machine intelligence 27 (5) (2005) 684–698.

[4] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, in: Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 663–670.

[5] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 171–184.

[6] E. J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, Journal of the ACM (JACM) 58 (3) (2011) 11.

[7] J. Wright, A. Ganesh, S. Rao, Y. Peng, Y. Ma, Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization, in: Advances in neural information processing systems, 2009, pp. 2080–2088.

[8] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, D. Kriegman, Clustering appearances of objects under varying illumination conditions, in: Computer vision and pattern recognition, 2003. Proceedings. 2003 IEEE computer society conference on, Vol. 1, IEEE, 2003, pp. I–I.

[9] T. Zhang, A. Szlam, G. Lerman, Median k-flats for hybrid linear modeling with many outliers, in: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 234–241.

[10] G. Chen, G. Lerman, Spectral curvature clustering (scc), International Journal of Computer Vision 81 (3) (2009) 317–330.

[11] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, IEEE transactions on pattern analysis and machine intelligence 35 (11) (2013) 2765–2781.

[12] A. Goh, R. Vidal, Segmenting motions of different types by unsupervised manifold clustering, in: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–6.

[13] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[14] R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (gpca), IEEE transactions on pattern analysis and machine intelligence 27 (12) (2005) 1945–1959.

[15] K.-i. Kanatani, Motion segmentation by subspace separation and model selection, in: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, Vol. 2, IEEE, 2001, pp. 586–591.

[16] P. Favaro, R. Vidal, A. Ravichandran, A closed form solution to robust subspace estimation and clustering, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1801–1807.

[17] R. Vidal, P. Favaro, Low rank subspace clustering (lrsc), Pattern Recognition Letters 43 (2014) 47–61.

[18] G. Liu, S. Yan, Latent low-rank representation for subspace segmentation and feature extraction, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1615–1622.

[19] J. Chen, H. Mao, Y. Sang, Z. Yi, Subspace clustering using a symmetric low-rank representation, Knowledge-Based Systems 127 (2017) 46–57.

[20] Y. Zheng, X. Zhang, S. Yang, L. Jiao, Low-rank representation with local constraint for graph construction, Neurocomputing 122 (2013) 398–405.

[21] M. Soltanolkotabi, E. Elhamifar, E. J. Candes, et al., Robust subspace clustering, The Annals of Statistics 42 (2) (2014) 669–699.

[22] C.-G. Li, R. Vidal, Structured sparse subspace clustering: A unified optimization framework, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 277–286.

[23] Y. Yang, J. Feng, N. Jojic, J. Yang, T. S. Huang, \ell^{0}-sparse subspace clustering, in: European Conference on Computer Vision, Springer, 2016, pp. 731–747.

[24] C. You, V. Palade, X. Wu, Robust structured low-rank representation for image segmentation, International Journal on Artificial Intelligence Tools 27 (05) (2018) 1850020.

[25] F. Nie, H. Huang, Subspace clustering via new low-rank model with discrete group structure constraint., in: IJCAI, 2016, pp. 1874–1880.

[26] A. Cuzzocrea, Accuracy control in compressed multidimensional data cubes for quality of answer-based olap tools, in: Scientific and Statistical Database Management, 2006. 18th International Conference on, IEEE, 2006, pp. 301–310.

[27] A. Cuzzocrea, F. Furfaro, D. Saccà, Enabling olap in mobile environments via intelligent data cube compression techniques, Journal of Intelligent Information Systems 33 (2) (2009) 95–143.

[28] J. Mairal, F. Bach, J. Ponce, Task-driven dictionary learning, IEEE transactions on pattern analysis and machine intelligence 34 (4) (2012) 791–804.

[29] L. Zhang, M. Yang, Z. Feng, D. Zhang, On the dimensionality reduction for sparse representation based face recognition, in: Pattern Recognition (ICPR), 2010 20th International Conference on, IEEE, 2010, pp. 1237–1240.

[30] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, R. Chellappa, Sparse embedding: A framework for sparsity promoting dimensionality reduction, in: European conference on Computer vision, Springer, 2012, pp. 414–427.

[31] V. M. Patel, H. Van Nguyen, R. Vidal, Latent space sparse subspace clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 225–232.

[32] L. Ma, C. Wang, B. Xiao, W. Zhou, Sparse representation for face recognition based on discriminative low-rank dictionary learning, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2586–2593.

[33] M. Yang, L. Zhang, X. Feng, D. Zhang, Fisher discrimination dictionary learning for sparse representation, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 543–550.

[34] I. Ramirez, P. Sprechmann, G. Sapiro, Classification and clustering via dictionary learning with structured incoherence and shared features, in:

Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 3501–3508.

[35] Z. Lin, R. Liu, Z. Su, Linearized alternating direction method with adaptive penalty for low-rank representation, in: Advances in neural information processing systems, 2011, pp. 612–620.

[36] J.-F. Cai, E. J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization 20 (4) (2010) 1956–1982.

[37] C. Peng, Z. Kang, Q. Cheng, Integrating feature and graph learning with low-rank representation, Neurocomputing 249 (2017) 106–116.

[38] J. Yan, M. Pollefeys, A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate, in: European conference on computer vision, Springer, 2006, pp. 94–106.

[39] 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.