Robust Subspace Clustering by Cauchy Loss Function

Xuelong Li, *Fellow, IEEE*, Quanmao Lu, Yongsheng Dong, *Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— Subspace clustering is a problem of exploring the low-dimensional subspaces of high-dimensional data. State-of-the-art approaches are designed by following the model of spectral clustering-based method. These methods pay much attention to learn the representation matrix to construct a suitable similarity matrix and overlook the influence of the noise term on subspace clustering. However, the real data are always contaminated by the noise and the noise usually has a complicated statistical distribution. To alleviate this problem, in this paper, we propose a subspace clustering method based on Cauchy loss function (CLF). Particularly, it uses CLF to penalize the noise term for suppressing the large noise mixed in the real data. This is due to that the CLF's influence function has an upper bound that can alleviate the influence of a single sample, especially the sample with a large noise, on estimating the residuals. Furthermore, we theoretically prove the grouping effect of our proposed method, which means that highly correlated data can be grouped together. Finally, experimental results on five real data sets reveal that our proposed method outperforms several representative clustering methods.**

***Index Terms*— Cauchy loss function (CLF), grouping effect, noise suppression, similarity matrix, subspace clustering.**

I. Introduction

SUBSPACE clustering, as an important clustering analysis technique, has gained much attention in recent years and has numerous applications in image processing and computer vision, e.g., image representation [1], motion segmentation [2], saliency detection [3], and image clustering [4], [5]. It aims to explore the low-dimensional structure lying in the high-dimensional data. Particularly, conventional [6] can be regarded as a special subspace clustering method which finds a single low-dimensional subspace of the high-dimensional data. However, in practice, data are always drawn from multiple low-dimensional subspaces and each subspace has a different dimension. For example, the trajectories of different motion objects usually belong to different affine subspaces, or face images of individuals under varying pose may lie in different linear subspaces. Motivated by these, subspace clustering is designed for seeking the low-dimensional subspace of the raw data and clustering the data into groups with each group fitting a subspace. Furthermore, subspace clustering problem is formally defined as follows:

Definition 1 (Subspace Clustering): Given a set of sufficiently sampled data vectors $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_k] = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, where d represents the feature dimension and n is the number of data. Assume that the data are drawn from a union of k subspaces $\{S_i\}_{i=1}^{k}$, and X_i be a collection of n_i points drawn from the subspace S_i, $n = \sum_{i=1}^{k} n_i$. The task of subspace clustering is to segment the data according to the underlying subspaces they are drawn from.

In the past two decades, many advances have been done to improve the performance of subspace clustering [7]–[13]. They can be roughly divided into four categories, including algebraic methods [14], [15], iterative methods [16], [17], statistical methods [18], [19], and spectral clustering-based methods [20]–[24]. Most recently, spectral clustering-based methods have shown its excellent performance in many applications. In general, spectral clustering-based methods consist of two main steps. First, a similarity or affinity matrix is constructed to represent the similarity between the samples in the raw data. Second, a spectral clustering algorithm is employed to divide the raw data into k groups based on the learned similarity matrix. Note that how to build a proper similarity matrix plays a decisive role in the process of subspace clustering. Therefore, most spectral clustering-based models were proposed to construct a more efficient similarity matrix.

Reviewing the existing methods, a similarity matrix is generally constructed using a self-expression model which regards the data itself as a dictionary to learn a representation matrix [25], [26]. Such a self-expression model assumes that the samples can be well represented using the points in the same subspace and the learned representation matrix can capture the similarity between the samples in the raw data.

Manuscript received December 31, 2016; revised November 23, 2017; accepted October 9, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1107400, in part by the National Natural Science Foundation of China under Grant 61871470, Grant 61761130079, Grant U1604153, and Grant 61301230, in part by the Program for Science and Technology Innovation Talents in Universities of Henan Province under Grant 19HASTIT026, and in part by the Training Program for the Young-Backbone Teachers in Universities of Henan Province under Grant 2017GGJS065. *(Corresponding author: Yongsheng Dong.)*

X. Li, Q. Lu, and D. Tao are with the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@opt.ac.cn; quanmao.lu.opt@gmail.com; dacheng.tao@gmail.com).

Y. Dong is with the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China, and also with the School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China (e-mail: dongyongsheng98@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2876327

Ideally, the learned representation matrix should be block diagonal [27], [28], which means the affinities of samples between cluster are all zeros. Considering that the real data usually contain noise, a loss function is employed to deal with the noise. Then, the general model of spectral clustering based methods can be formulated as

$$\begin{aligned}&\min_{\mathbf{Z},\mathbf{E}}\ \varphi(\mathbf{E})+\delta(\mathbf{Z})\\&\text{s.t. } \mathbf{X}=\mathbf{X}\mathbf{Z}+\mathbf{E}\end{aligned}\tag{1}$$

where $\mathbf{X}$ is the original data matrix,$\mathbf{Z}$ is the representation matrix, and $\mathbf{E}$ represents the noise matrix. The functions of $\varphi(\mathbf{E})$ and $\delta(\mathbf{Z})$ are designed for restricting $\mathbf{E}$ and $\mathbf{Z}$, respectively. In many works, $\varphi(\mathbf{E})$ and $\delta(\mathbf{Z})$ are two properly norms. For example, sparse subspace clustering (SSC) [22] uses ℓ_1-norm to regularize the matrix $\mathbf{Z}$ for seeking the most sparsest representation of each point and chooses Frobenius norm to deal with the noise term $\mathbf{E}$. Different with SSC, low-rank representation (LRR) [29] employs the nuclear norm to regularize the matrix $\mathbf{Z}$ for capturing the correlation structure of the data and uses ℓ_{21}-norm to describe the matrix $\mathbf{E}$. Based on SSC and LRR, many works were proposed to design different regularizations for the representation matrix $\mathbf{Z}$ and choose a simple norm on the noise matrix $\mathbf{E}$ [30]–[35].

Note that the previous works mainly focus on choosing a proper norm to regularize the representation matrix and ignore the influence of the noise term on subspace clustering. However, the real data are always contaminated by the unknown noise, and the noise usually has a complicated statistical distribution [29], [36], [37]. If we cannot adopt a proper model to deal with the noise, the learned representation matrix may fail to capture the similarity between samples which can result in an unreliable subspace clustering result. Therefore, how to handle the noise is a difficult task and has a significant influence on subspace clustering. Although the existing methods choose the different norm to handle the noise, they can only deal with the specific noise. For example, ℓ_1-norm is suitable for entrywise corruptions, ℓ_{21}-norm is for sample-specific corruptions, and Frobenius norm is to tackle Gaussian noise. In addition, Li *et al.* [36] tried to describe the noise using mixture of Gaussian (MoG) regression. Although it has shown its superiority through the comparison experiments, it is sensitive to the number of Gaussian and has high computational cost.

To alleviate the noise's effect on subspace clustering, in this paper, we propose a subspace clustering method by using Cauchy loss function (CLF) to suppress the noise term. Compared with the conventional ℓ_1 or ℓ_2 loss, the influence function of CLF has an upper bound. Therefore, it can alleviate the influence of a single sample, especially the sample with a large noise, on estimating the residuals. Therefore, CLF has less dependence on the distribution of the noise and is more robust to the noise. Because our work mainly focuses on the noise term, we simply use the Frobenius norm to regularize the representation matrix. Furthermore, we prove the grouping effect of our method, which means that highly correlated data can be grouped together. Experimental results on the real data sets show the effectiveness of our proposed method.

A. Paper Contributions and Organization

Our work has the following three main contributions.

1) We propose a robust subspace clustering method based on CLF. Specifically, CLF is able to penalize the point with large noise rather than giving a specific assumption on the distribution of the noise. Therefore, our method is more robust to different kinds of noise in the real data.
2) The grouping effect of our method is theoretically proved, which can preserve the local structure in the raw data. Therefore, highly correlated point can be grouped together in the low-dimensional subspace.
3) We verify our method on different real applications, including motion segmentation and image clustering. The experimental results show that our method achieves better performance than several representative methods.

The rest of this paper is arranged as follows. The related work is introduced in Section II. Section III gives the problem formulation and the whole framework of our subspace clustering algorithm. In Section IV, we prove the grouping effect of our method which is a very useful property for subspace clustering and then analyze the convergence of our optimization algorithm. The experimental results on real databases are presented in Section V. Finally, the paper is briefly concluded in Section VI.

II. Related Work

Considering that our proposed method is a kind of spectral clustering-based method, we mainly review the most recent and related works. Throughout the paper, we use the nonbold letters, bold lower case letters and bold upper case letters to represent scalars, vectors, and matrices, respectively.

SSC [22], as a first proposed spectral clustering-based method, aims to find the sparsest representation for each point with all other points in a union of subspaces by solving the following problem:

$$\begin{aligned}&\min_{\mathbf{Z},\mathbf{E}}\ \|\mathbf{E}\|_{\mathbf{F}}^{\mathbf{2}}+\lambda\|\mathbf{Z}\|_{\mathbf{0}}\\&\text{s.t. } \mathbf{X}=\mathbf{X}\mathbf{Z}+\mathbf{E},\quad \text{diag}(\mathbf{Z})=\mathbf{0}\end{aligned}\tag{2}$$

where $\lambda > 0$ is a weighting factor to balance two terms. $\text{diag}(\mathbf{Z})=\mathbf{0}$ is used to avoid the solution $\mathbf{Z}$ being an identity matrix, which means that one point cannot be reconstructed using itself. As we all know, solving such sparse representation is an NP-hard problem. Therefore, SSC uses ℓ_1-norm to approximate the ℓ_0-norm. The final objective function is given as

$$\begin{aligned}&\min_{\mathbf{Z},\mathbf{E}}\ \|\mathbf{E}\|_{\mathbf{F}}^{\mathbf{2}}+\lambda\|\mathbf{Z}\|_{\mathbf{1}}\\&\text{s.t. } \mathbf{X}=\mathbf{X}\mathbf{Z}+\mathbf{E},\quad \text{diag}(\mathbf{Z})=\mathbf{0}.\end{aligned}\tag{3}$$

SSC assumes that one point can be reconstructed only using few points in the same subspace. When the data are drawn from independent subspaces, SSC can divide the points into their subspaces. However, for the real data, the representation matrix of SSC may be too sparse to capture the relationship between points in the same subspace. Based on SSC, Wang and Xu [38] proposed a modified version, named Noisy SSC (NSSC), to deal with noisy data.

LRR [29] was proposed to capture the correlation structure of the data by finding an LRR of the samples instead of a sparse one. The original problem of LRR is formulated as

$$\min_{\mathbf{Z}} \ \operatorname{rank}(\mathbf{Z}) \quad \text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z}. \tag{4}$$

The above-mentioned optimization problem is hard to be solved due to the discrete nature of the rank function. Therefore, LRR adopts the nuclear norm as a surrogate of the rank function. Furthermore, LRR uses ℓ_{21}-norm to deal with the noise term for improving its robustness to the noise and outliers. The subspace clustering problem becomes

$$\min_{\mathbf{Z}} \ \|\mathbf{E}\|_{\mathbf{21}} + \lambda\|\mathbf{Z}\|_* \quad \text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}. \tag{5}$$

However, there is no theoretical analysis about the importance of low rank property of the representation matrix $\mathbf{Z}$ for subspace clustering. In addition, the solution $\mathbf{Z}^*$ may be very dense and far from block diagonal.

Least Squares Regression (LSR) [27] employs the Frobenius norm to handle the representation matrix and the noise matrix simultaneously. The corresponding optimization problem is defined as

$$\min_{\mathbf{Z}} \ \|\mathbf{E}\|_{\mathbf{F}}^{\mathbf{2}} + \lambda\|\mathbf{Z}\|_{\mathbf{F}}^{\mathbf{2}} \quad \text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}. \tag{6}$$

Note that the above-mentioned problem can be efficiently solved. The main contribution of LSR is that it encourages grouping effect which can group highly correlated data together.

In order to balance the sparsity and low-rank property of the representation matrix, Correlation Adaptive Subspace Segmentation (CASS) [30] was proposed to optimize the problem

$$\min_{\mathbf{Z},\mathbf{E}} \ \|\mathbf{E}\|_F^2 + \lambda \sum_{i=1}^{n} \|\mathbf{X}\operatorname{diag}(\mathbf{z}_i)\|_* \quad \text{s.t. } \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E} \tag{7}$$

where $\|\mathbf{X}\operatorname{diag}(\mathbf{z}_i)\|_*$ is trace lasso and its definition can be found in [30]. Due to taking the data correlation into account, it can adaptively interpolate SSC and LSR.

MoG Regression [36], as a most related method to our work, uses the mixture of Gaussian model to describe the noise term and tries to solve the following problem:

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{E},\pi,\Sigma} \ & -\sum_{i=1}^{n} \ln\left(\sum_{k=1}^{K} \pi_k N(\mathbf{e}_i|0,\Sigma_k)\right) + \lambda\|\mathbf{Z}\|_F^2 \\ \text{s.t. } & \mathbf{X} = \mathbf{X}\mathbf{Z} + \mathbf{E}, \quad \operatorname{diag}(\mathbf{Z}) = \mathbf{0} \\ & \pi_k \geq 0, \Sigma_k \in S^+, \quad \sum_{k=1}^{K} \pi_k = 1 \end{aligned} \tag{8}$$

where π_k is the mixing weight, $\mathbf{e}_n$ is mean vector, Σ_k is the covariance matrix, and K denotes the number of Gaussian. Although MoG regression has a better performance than the single Gaussian model, it is only an extended version of single Gaussian and is sensitive to the number of Gaussian. In addition, solving the above-mentioned problem needs high computation cost.

III. Subspace Clustering by CLF

In this paper, we propose a new spectral clustering-based method to alleviate the influence of the noise on subspace clustering. Particularly, we employ CLF to suppress the noise. Next, we give the details of our optimization objection function and the framework of our subspace clustering method.

A. Problem Formulation

In statistics, M-estimator is a broad class of estimators, which is used to represent the minima of sum of functions. Let r_i denote the residual of the ith data with its estimated value and $\rho(r_i)$ be a symmetric and positive define function which has a unique minimum at zero. M-estimator aims to optimize the following problem:

$$\min \sum_i \rho(r_i). \tag{9}$$

The influence function of ρ function is defined as

$$\psi(x) = \frac{\partial\rho(x)}{\partial x} \tag{10}$$

which is used to measure the effect of changing the point of the sample on the value of the parameter estimation.

We demonstrate different estimators and their influence functions in Fig. 1. For the l_2 estimator (least squares) with $\rho(x) = x^2$, its influence function is $\psi(x) = x$. From Fig. 1, we can see that the influence of a sample on the estimate grows linearly as the error increases. This means the l_2 estimator is not robust to the noise. Although the l_1 estimator (least-absolute deviation) with $\rho(x) = |x|$ can alleviate the effect of the large error, its influence function has no cutoff [39], [40]. For a robust estimator, its influence function should not be sensitive to the increase of the error. CLF gives good characteristic on this aspect, and its definition is shown as follows:

$$\rho(x) = \log(1 + (x/c)^2) \tag{11}$$

with influence function

$$\psi(x) = \frac{2x}{x^2 + c^2} \tag{12}$$

where c is a constant. Note that CLF's influence function has the upper bound and its value tends to zero with the increase of the error.

Considering CLF is robust to the noise, we use CLF to penalize the noise term which is defined as

$$\sum_{i=1}^{n} \log\left(1 + \frac{\|\mathbf{x}_i - \mathbf{X}\mathbf{z}_i\|_2^2}{c^2}\right) \tag{13}$$

where $\mathbf{X}$ is the data matrix, and $\mathbf{z}_i$ denotes the representation vector of the ith data $\mathbf{x}_i$. As stated before, we simply use the Frobenius norm to regularize the representation matrix for verifying the influence of the noise model on subspace clustering

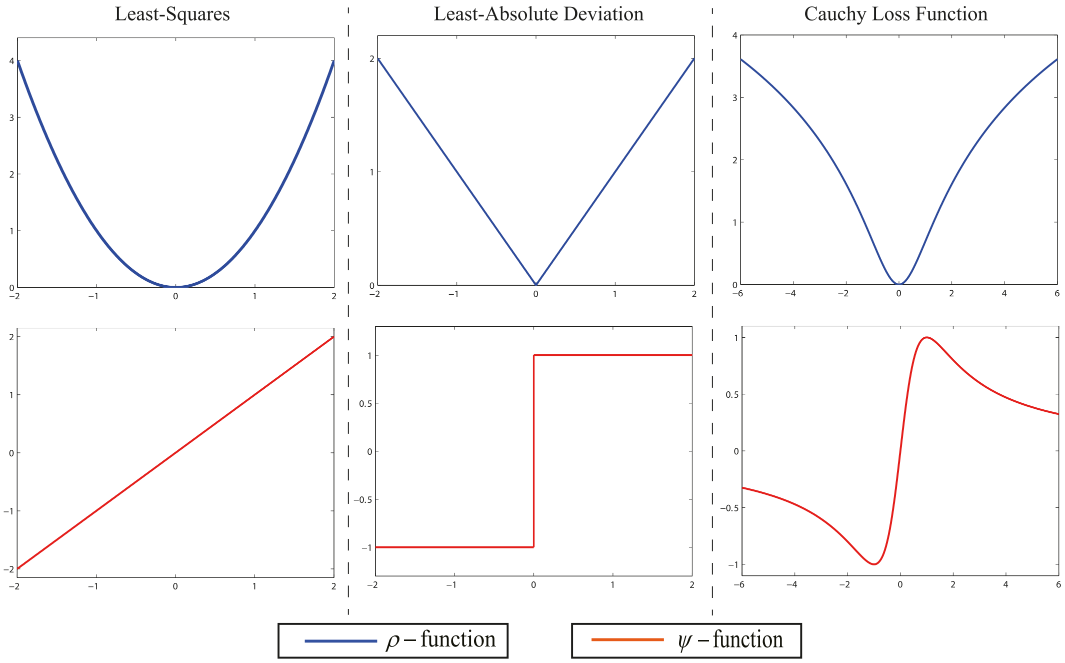


Fig. 1. Illustration of different estimators. Right: least-squares. Middle: least-absolute deviation. Left: CLF.

and facilitating the problem solving. The corresponding model can be formulated as

$$\min_{\mathbf{Z}} \sum_{i=1}^{n} \log\left(1+\frac{\|\mathbf{x}_i-\mathbf{X}\mathbf{z}_i\|_2^2}{c^2}\right)+\lambda\|\mathbf{Z}\|_F^2 \tag{14}$$

where λ is a weight factor to balance the effect of two terms. For the formula (14), an iterative algorithm can be employed to find the solution for each data point, but it is not a high-efficiency way to obtain the representation matrix. In order to reduce the time complexity and keep the valuable property, we revise (14) and give the final objective function

$$\min_{\mathbf{Z}} \log\left(1+\frac{\|\mathbf{X}-\mathbf{X}\mathbf{Z}\|_F^2}{c^2}\right)+\lambda\|\mathbf{Z}\|_F^2. \tag{15}$$

Note that it takes the representation matrix $\mathbf{Z}$ as an integrate to learn. Therefore, we can directly to optimize the representation matrix by using an iteration process.

B. Optimization

For the problem (15), we adopt iteratively reweighted residuals (IRR) method to find the solution. Given the data matrix X, (15) can be rewritten as

$$\min_{\mathbf{Z}} \mathcal{J} = \log\left(1+\frac{\|\mathbf{X}-\mathbf{X}\mathbf{Z}\|_F^2}{c^2}\right)+\lambda\|\mathbf{Z}\|_F^2. \tag{16}$$

Setting the derivative of $\mathcal{J}$ with respect to $\mathbf{Z}$ to zero, we have

$$\frac{-2\mathbf{X}^T(\mathbf{X}-\mathbf{X}\mathbf{Z})}{c^2+\|\mathbf{X}-\mathbf{X}\mathbf{Z}\|_F^2}+2\lambda\mathbf{Z}=0 \tag{17}$$

which is equivalent to

$$\left(\frac{\mathbf{X}^T\mathbf{X}}{c^2+\|\mathbf{X}-\mathbf{X}\mathbf{Z}\|_F^2}+\lambda\mathbf{I}\right)\mathbf{Z}=\frac{\mathbf{X}^T\mathbf{X}}{c^2+\|\mathbf{X}-\mathbf{X}\mathbf{Z}\|_F^2}. \tag{18}$$

Then, we can obtain the solution

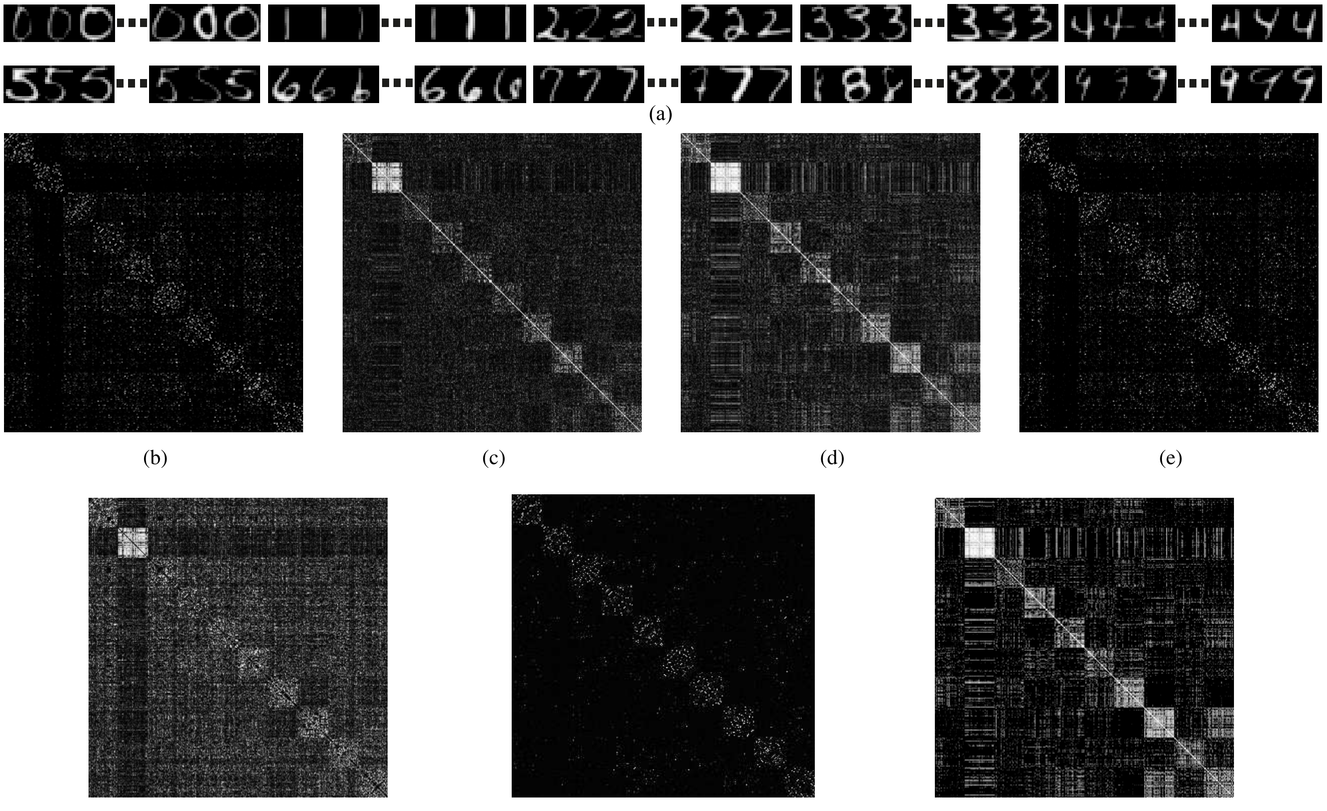
$$\begin{cases}\mathbf{Z}=Q(Q\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\\ Q=\dfrac{1}{c^2+\|\mathbf{R}\|_F^2}\\ \mathbf{R}\!=\!\mathbf{X}\!-\!\mathbf{X}\mathbf{Z}\end{cases} \tag{19}$$

where $\mathbf{R}$ is the residual of the data matrix with the corrected matrix, and Q is the weight function which is used to reduce the effect of the noise. Note that Q should be calculated using the representation matrix $\mathbf{Z}$. Then, an iterative way is adopted to update $\mathbf{Z}$ until convergence. The whole procedure for solving (15) is described in Algorithm 1.

```
Algorithm 1 IRR
Input: data matrix $\mathbf{X}$, parameters $\lambda$ and $c$, initial representation matrix $\mathbf{Z}^0$, $t = 0$.
Output: $\mathbf{Z}^*$.
  while not converge do
    1) $\mathbf{R}^{t+1} \leftarrow \mathbf{X} - \mathbf{X}\mathbf{Z}^t$
    2) $Q^{t+1} \leftarrow 1/(c^2 + \|\mathbf{R}^{t+1}\|_F^2)$
    3) $\mathbf{Z}^{t+1} \leftarrow Q^{t+1}\left(Q^{t+1}\mathbf{X}^T\mathbf{X} + 2\lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{X}$
  end while
```

C. Subspace Clustering Algorithm via CLF

In this section, we give the framework of our proposed subspace clustering algorithm which is outlined in Algorithm 2. Note that we first use Algorithm 1 to find the representation matrix $\mathbf{Z}^*$. Then, the similarity matrix is defined as $\mathbf{W} = (|\mathbf{Z}^*| + |(\mathbf{Z}^*)^T|)/2$, where $(\mathbf{Z}^*)^T$ is the transposition of $\mathbf{Z}^*$. Finally, Normalized Cuts [41], a kind of spectral clustering



(a)

(b) (c) (d) (e)

(f) (g) (h)

Fig. 2. Example of the similarity matrix W of 10 classes derived by different methods on the USPS database. (a) Some images of the USPS database. (b) SSC. (c) LRR. (d) LSR. (e) CASS. (f) MoG. (g) NSSC. (h) Ours.

```
Algorithm 2 Subspace Clustering Algorithm via CLF
Require: data matrix $\mathbf{X}$, number of subspaces $k$ leftmargin=1pt
    1) Solve the problem (15) and obtain the final representation matrix $\mathbf{Z}^*$.
    2) Construct similarity matrix W using $(|\mathbf{Z}^*| + |(\mathbf{Z}^*)^T|)/2$.
    3) Group the data points into $k$ clusters by Normalized Cuts.
```

algorithm [42], is employed to group the data points into k clusters based on the similarity matrix.

In order to demonstrate the structure of the learned similarity matrix, we show the similarity matrices of 10 subjects derived by SSC, LRR, LSR, CASS, MoG Regression, NSSC, and our proposed method on the USPS data set in Fig. 2. For simplicity, we use MoG to denote MoG Regression. USPS is a popular handwritten digit database for clustering analysis. From Fig. 2, we can see that all the methods can give an approximate block-diagonal matrix. The similarity matrices obtained by SSC and CASS are sparse and similar which means that CASS gives a large weight for the sparsity of the representation matrix. In addition, NSSC also gives a very sparse similarity matrix. However, the points in the same cluster have no high correlation which can degenerate the performance of subspace clustering. In contrast, the similarity matrices learned by LRR, LSR, MoG Regression, and our method are very dense, which give high similarity for the samples within the same cluster. Furthermore, we define a Contrast Index (CI) to quantitatively measure the difference between diagonal blocks and nondiagonal blocks of the similarity matrix. The corresponding formulation is

TABLE I

CI (%) OF THE SIMILARITY MATRICES OBTAINED BY DIFFERENT METHODS

Method	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
CI	30.39	23.89	25.46	30.30	16.80	32.39	**38.12**

$$CI = \frac{S_D}{S_D + S_{ND}} = \frac{S_D}{\|W\|_1} \tag{20}$$

where S_D and S_{ND} denote the sum of elements in diagonal and nondiagonal blocks, respectively. Table I gives the CI of the similarity matrices obtained by different methods. Note that MoG gives the lowest CI which can be seen from Fig. 2. Obviously, our method gives a higher CI than other methods, which means that our proposed model has greater ability to group correlated data together.

IV. THEORETICAL ANALYSIS

In this section, we prove that our proposed method has the grouping effect which can group highly correlated data together, and then analyze the convergence of our optimization algorithm.

A. Grouping Effect

Theorem 1: Given a data point $\mathbf{x} \in \mathbb{R}^d$, the normalized data matrix $\mathbf{X}$, and a parameter λ. Let $\hat{\mathbf{z}}$ be the optimal solution to the following problem (in vector form):

$$\min_{\mathbf{z}} \log\left(1 + \frac{\|\mathbf{x} - \mathbf{X}\mathbf{z}\|_2^2}{c^2}\right) + \lambda\|\mathbf{z}\|_2^2. \tag{21}$$

Then, we have

$$\frac{|\hat{z}^i - \hat{z}^j|}{\|\mathbf{x}\|_2} \le \frac{1}{\lambda c^2}\sqrt{2(1-r)} \tag{22}$$

where $r = \mathbf{x}_i^T\mathbf{x}_j$ is the sample correlation. $\hat{z}^i$ and $\hat{z}^j$ are the ith and jth entries of vector $\hat{\mathbf{z}}$. $\mathbf{x}_i$ and $\mathbf{x}_j$ are the ith and jth columns of $\mathbf{X}$.

Proof: Let

$$L(\mathbf{z}) = \log\left(1 + \frac{\|\mathbf{x} - \mathbf{X}\mathbf{z}\|_2^2}{c^2}\right) + \lambda\|\mathbf{z}\|_2^2. \tag{23}$$

Since $\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} L(\mathbf{z})$, we have

$$\left.\frac{\partial L(\mathbf{z})}{\partial \mathbf{z}}\right|_{\mathbf{z}=\hat{\mathbf{z}}} = 0. \tag{24}$$

This gives

$$\frac{-2\mathbf{x}_i{}^T(\mathbf{x} - \mathbf{X}\hat{\mathbf{z}})}{c^2 + \|\mathbf{x} - \mathbf{X}\hat{\mathbf{z}}\|_2^2} + 2\lambda\hat{z}^i = 0 \tag{25}$$

$$\frac{-2\mathbf{x}_j{}^T(\mathbf{x} - \mathbf{X}\hat{\mathbf{z}})}{c^2 + \|\mathbf{x} - \mathbf{X}\hat{\mathbf{z}}\|_2^2} + 2\lambda\hat{z}^j = 0 \tag{26}$$

Equations (25) and (26) give

$$\hat{z}^i - \hat{z}^j = \frac{\left(\mathbf{x}_i{}^T - \mathbf{x}_j{}^T\right)\left(\mathbf{x} - \mathbf{X}\hat{\mathbf{z}}\right)}{\lambda\left(c^2 + \|\mathbf{x} - \mathbf{X}\hat{\mathbf{z}}\|_2^2\right)} \le \frac{\left(\mathbf{x}_i{}^T - \mathbf{x}_j{}^T\right)\left(\mathbf{x} - \mathbf{X}\hat{\mathbf{z}}\right)}{\lambda c^2}. \tag{27}$$

Since each column of $\mathbf{X}$ is normalized, $\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{2(1-r)}$, where $r = \mathbf{x}_i{}^T\mathbf{x}_j$. Note that $\hat{\mathbf{z}}$ is the optimal to the problem (21), and we deduce

$$\log\left(1 + \frac{\|\mathbf{x} - \mathbf{X}\hat{\mathbf{z}}\|_2^2}{c^2}\right) \le \log\left(1 + \frac{\|\mathbf{x} - \mathbf{X}\hat{\mathbf{z}}\|_2^2}{c^2}\right) + \lambda\|\hat{\mathbf{z}}\|_2^2 = L(\hat{\mathbf{z}}) \le L(0) = \log\left(1 + \frac{\|\mathbf{x}\|_2^2}{c^2}\right). \tag{28}$$

Thus, $\|\mathbf{x} - \mathbf{X}\hat{\mathbf{z}}\|_2 \le \|\mathbf{x}\|_2$. Finally, we obtain

$$\frac{|\hat{z}^i - \hat{z}^j|}{\|\mathbf{x}\|_2} \le \frac{1}{\lambda c^2}\sqrt{2(1-r)}. \tag{29}$$

□

As stated in Theorem 1, if $\mathbf{x}_i$ and $\mathbf{x}_j$ are highly correlated, the value of r is close to 1, which means that the difference between $\hat{z}^i$ and $\hat{z}^j$ is almost 0. Then, $\mathbf{x}_i$ and $\mathbf{x}_j$ can be grouped into the same cluster. Note that Theorem 1 gives the grouping effect for one point (vector form). For the matrix form, the corresponding grouping effect can still be proved using the similar proof procedure of Theorem 1.

B. Convergence Analysis

We employ the Weiszfeld's method [43] to analyze the convergence of Algorithm 1. The formula (16) is equivalent to

$$\min_{\mathbf{z}_1,\ldots,\mathbf{z}_n} \mathcal{J}(\mathbf{Z}) = \log\left(1 + \frac{\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{X}\mathbf{z}_i\|_2^2}{c^2}\right) + \lambda\sum_{i=1}^{n}\|\mathbf{z}_i\|_2^2 \tag{30}$$

where $\mathbf{z}_i$ is the representation vector of $\mathbf{x}_i$. The solution $\mathbf{Z}$ in (19) can be rewritten as

$$\mathbf{z}_i = Q(Q\mathbf{X}^T\mathbf{X} + 2\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{x}_i, \quad i = 1, 2, \ldots, n. \tag{31}$$

The main idea of the Weiszfelds method is to globally approximate $\mathcal{J}$ using a sequence of quadratic function [44]. After obtaining the solution $\mathbf{Z}^k$, we can define an upper bound of $\mathcal{J}(\mathbf{z}_i)$ as $\phi(\mathbf{z}_i; \mathbf{z}_i^k)$, where $\mathcal{J}(\mathbf{z}_i)$ is obtained by fixing the other variables in $\mathcal{J}(\mathbf{Z})$. $\phi(\mathbf{z}_i; \mathbf{z}_i^k)$ should satisfy the following conditions:

$$\begin{aligned}\phi\left(\mathbf{z}_i^k; \mathbf{z}_i^k\right) &= \mathcal{J}\left(\mathbf{z}_i^k\right)\\ \phi'\left(\mathbf{z}_i^k; \mathbf{z}_i^k\right) &= \mathcal{J}'\left(\mathbf{z}_i^k\right)\end{aligned} \tag{32}$$

Then, $\phi(\mathbf{z}_i; \mathbf{z}_i^k)$ has the form

$$\begin{aligned}\phi\left(\mathbf{z}_i; \mathbf{z}_i^k\right) = \mathcal{J}\left(\mathbf{z}_i^k\right) + \left(\mathbf{z}_i - \mathbf{z}_i^k\right)^T\mathcal{J}'\left(\mathbf{z}_i^k\right)\\ + \left(\mathbf{z}_i - \mathbf{z}_i^k\right)^T C\left(\mathbf{z}_i^k\right)\left(\mathbf{z}_i - \mathbf{z}_i^k\right)\end{aligned} \tag{33}$$

with symmetric matrix $C(\mathbf{z}_i^k)$

$$C(\mathbf{z}_i^k) = \frac{\mathbf{X}^T\mathbf{X}}{c^2 + \|\mathbf{X} - \mathbf{X}\mathbf{Z}^k\|_F^2} + \lambda\mathbf{I}. \tag{34}$$

Then, the convergence of Algorithm 1 can be guaranteed by the following theorem.

Theorem 2: The IRR algorithm proposed in Algorithm 1 guarantees that the objective function value of (16) is monotone decreasing in iterations, i.e., $\mathcal{J}(\mathbf{Z}^{k+1}) \le \mathcal{J}(\mathbf{Z}^k)$, until it converges.

Proof: Suppose that $\phi(\mathbf{z}_i; \mathbf{z}_i^k)$ is locally convex with respect to $\mathbf{z}_i$ and has a local minimizer. Let $\mathbf{z}_i^{k+1}$ be the minimizer, we get

$$\phi'\left(\mathbf{z}_i{}^{k+1}; \mathbf{z}_i^k\right) = \mathcal{J}'\left(\mathbf{z}_i^k\right) + 2C\left(\mathbf{z}_i^k\right)\left(\mathbf{z}_i{}^{k+1} - \mathbf{z}_i^k\right) = 0. \tag{35}$$

Substituting for $\mathcal{J}'(\mathbf{z}_i^k)$, we can obtain the update rule in (31).

By appropriately choose $\mathbf{z}_i^k$ near $\mathbf{z}_i$, we have $\mathcal{J}(\mathbf{z}_i) \le \phi(\mathbf{z}_i; \mathbf{z}_i^k)$ which implies that

$$\begin{aligned}\mathcal{J}\left(\mathbf{z}_i^{k+1}\right) &\le \phi\left(\mathbf{z}_i{}^{k+1}; \mathbf{z}_i^k\right)\\ &= \mathcal{J}\left(\mathbf{z}_i^k\right) + \left(\mathbf{z}_i{}^{k+1} - \mathbf{z}_i^k\right)^T\mathcal{J}'\left(\mathbf{z}_i^k\right)\\ &\quad + \left(\mathbf{z}_i{}^{k+1} - \mathbf{z}_i^k\right)^T C\left(\mathbf{z}_i^k\right)\left(\mathbf{z}_i{}^{k+1} - \mathbf{z}_i^k\right).\end{aligned} \tag{36}$$

Equations (35) and (36) give

$$\begin{aligned}\mathcal{J}\left(\mathbf{z}_i^{k+1}\right) - \mathcal{J}\left(\mathbf{z}_i^k\right) &\le -\left(\mathbf{z}_i{}^{k+1} - \mathbf{z}_i^k\right)^T C\left(\mathbf{z}_i^k\right)\left(\mathbf{z}_i{}^{k+1} - \mathbf{z}_i^k\right)\\ &\le -\lambda\|\mathbf{z}_i{}^{k+1} - \mathbf{z}_i^k\|^2 \le 0.\end{aligned} \tag{37}$$

Therefore, we have $\mathcal{J}(\mathbf{z}_i^{k+1}) \leq \mathcal{J}(\mathbf{z}_i^{k})$. Based on (30), we can easily deduce

$$\mathcal{J}(\mathbf{Z}^{k+1}) \leq \mathcal{J}(\mathbf{Z}^{k}). \tag{38}$$

□

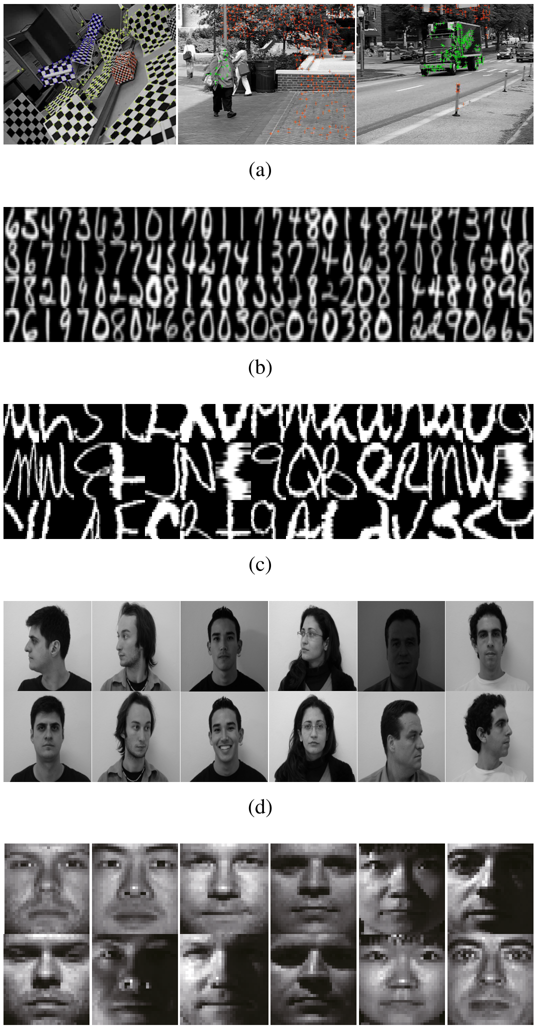
V. Experimental Verification and Analysis

In this section, we verify the effectiveness of our proposed method on five real databases: Hopkins 155 motion segmentation database [45], USPS [46], C-Cube [47], [48], Fundacao Educacional Inaciana (FEI) and Extended Yale B database [49]. Our method is compared with the traditional Kmeans, SSC [22], LRR [29], LSR [27], CASS [30], MoG regression [36], and NSSC [38]. SSC, LRR, LSR, CASS, MoG regression, and NSSC are representative subspace clustering methods which are introduced in Section II. For fair comparison with the previous methods, we adopt the same preprocessing for the whole databases: use principle component analysis to reduce the dimension of the original data and keep nearly 98% energy. In addition, the parameters of each method are manually tuned to achieve their best performance. Finally, we employ the clustering accuracy (CA) [50], [51] and the normalized mutual information (NMI) metric [52], [53] to evaluate the subspace clustering results. From the experimental results, we can see that our method achieves better performance than other state-of-the-art methods.

A. Data Sets

We initially give the detailed description about five real data sets used in the experiments.

1) The first data set is the Hopkins 155 motion segmentation database. It consists of 155 video sequences, where 120 of the videos have 2 motions and 35 of the videos contain 3 motions (a motion corresponding to a subspace). For each video, feature trajectories have been extracted for clustering. The number of feature trajectories of each video ranges from 39 to 550. Each video can be regarded as a subspace clustering task, and so there are 155 subspace segmentation tasks totally.
2) The second is the USPS database which is one of the standard data sets for handwritten digit recognition [54]. It contains 9298 images of hand-written digits from 0 to 9. The size of each image is 16×16. To reduce the memory consumption in our experiments, we randomly select 30 images for each digit to construct a subset with 300 samples.
3) The third is the C-Cube cursive character data set which contains both the upper and lower case of 26 letters. It has 57 646 character images and the average dimension of all images is about 3120. For each subject, we randomly select 20 images to form a subset for our experiments. Then, each image is normalized to 24×24 pixel array and reshaped to a vector.
4) The forth data set is the FEI part 1 database. This database is the subset of the whole FEI database. It contains 700 images with 50 subjects, and each subject has 14 images captured from a large range of views.



(a)

(b)

(c)

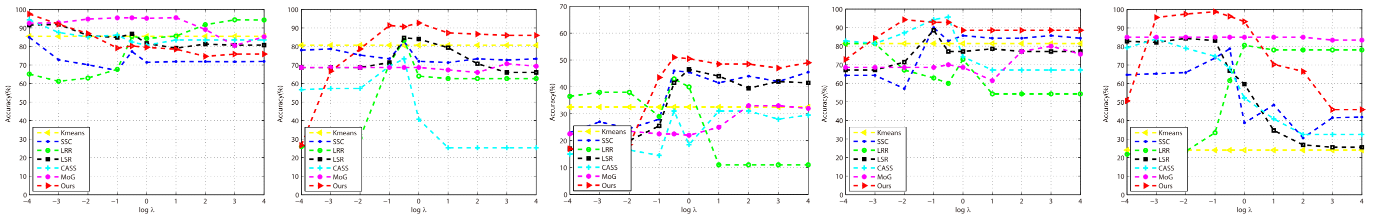
(d)

(e)

Fig. 3. Examples of different data sets. For the Hopkins 155 database, we simply choose some frames in the videos. The motion objects in these three frames are checkerboard, people, and truck, respectively. For the FEI and Extended Yale B database, each column represents a single subject. (a) Hopkins 155 motion segmentation database. (b) USPS database. (c) C-Cube database. FEI database. (d) Extended Yale B database.

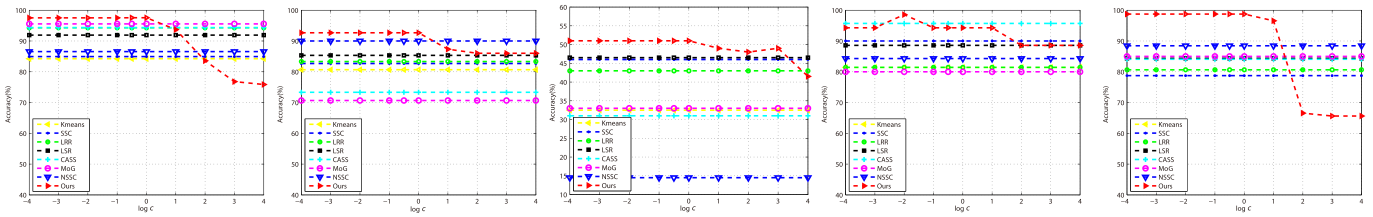
5) The fifth data set is the Extended Yale B Database which is a popular data set for image clustering [55]–[57]. It consists of 2414 frontal face images of 38 subjects, and each subject has about 64 frontal face images with different pose, angle, and illumination conditions. In our experiment, we construct three subspace clustering tasks based on the first 5, 8, and 10 subjects, and each subject has 64 face images.

Fig. 3 gives some samples of these five data sets. From Fig. 3(e), we can see that Extended Yale B is a tough database for subspace clustering due to its large noise. Therefore, we can further verify the effectiveness of our method in handling the noise. Table II gives the statistics of these databases. For the Hopkins 155 database, the values of size and dimensionality represent the average of the whole videos, and the class of each video is 2 or 3.



(a) (b) (c) (d) (e)

Fig. 4. Performance of different methods versus parameter λ. (a) Hopkins 155. (b) USPS. (c) C-Cube. (d) FEI. (e) Extended Yale B.



(a) (b) (c) (d) (e)

Fig. 5. Performance of different methods versus parameter c. (a) Hopkins 155. (b) USPS. (c) C-Cube. (d) FEI. (e) Extended Yale B.

TABLE II
STATISTICS OF FIVE DATA SETS

dataset	size	dimensionality	# of classes
Hopkins 155	59	296	2 or 3
USPS	9298	256	10
C-Cube	57646	3120	52
FEI	700	768	50
Extended Yale B	2414	1024	68

B. Evaluation Criterion

The clustering results are evaluated by comparing the obtained label of each subspace clustering method with the ground truth. The CA and the NMI, as two popular metrics, are employed to measure the clustering performance.

Given an obtained label vector $\mathbf{o}_i$ and a corresponding ground truth label vector $\mathbf{g}_i$. The CA is calculated by

$$\text{CA} = \frac{\sum_{j=1}^{n} \delta\big(\mathbf{g}_i(j), \mathbf{o}_i{}'(j)\big)}{n} \tag{39}$$

$$\delta(x, y) = \begin{cases} 1, & \text{if} x = y \\ 0, & \text{else} \end{cases} \tag{40}$$

where $\mathbf{o}_i{}' = \text{map}(\mathbf{o}_i)$. $\text{map}(\mathbf{o}_i)$ is the permutation mapping function that chooses $\mathbf{g}_i$ as a reference vector and maps each element in $\mathbf{o}_i$ to the equivalent label in $\mathbf{g}_i$. Therefore, $\text{map}(\mathbf{o}_i)$ is designed for solving the problem of correspondence between two label vectors. Kuhn–Munkres algorithm can be utilized to find the best mapping.

Mutual Information (MI), as a symmetric measure to quantify the information shared between two statistical distributions, provides a degree of agreement between two clustering results. Let c_p be the cluster obtained from the ground truth $\mathbf{g}_i$ and c'_q obtained from our clustering result $\mathbf{o}_i$. Then, the corresponding MI is defined as follows:

$$\text{MI}(\mathbf{g}_i, \mathbf{o}_i) = \sum_{p=1}^{k} \sum_{q=1}^{k'} \frac{n_{pq}}{n} \log \left(\frac{\frac{n_{pq}}{n}}{\frac{n_p}{n} \cdot \frac{n'_q}{n}} \right) \tag{41}$$

where k and k' denote the number of clusters in ground truth and our clustering result, respectively. n_p is the number of points in cluster c_p, n'_q is the number of points in cluster c'_q, and n_{pq} denotes the number of shared points between c_p and c'_q. In order to obtain a normalized version of MI that ranges from 0 to 1, we use the NMI metric as

$$\text{NMI}(\mathbf{g}_i, \mathbf{o}_i) = \frac{2 \cdot MI(\mathbf{g}_i, \mathbf{o}_i)}{H(\mathbf{g}_i) + H(\mathbf{o}_i)} \tag{42}$$

where $H(\cdot)$ denotes the entropy function.

C. Parameter Selection

Our proposed method has two essential parameters, namely, the weight factor λ and a constant c. Then, we conduct the corresponding comparison experiments to choose the best parameter for each method on the whole databases. To reduce the memory consumption in our experiments, we only use the first five videos of the Hopkins 155 database to choose the appropriate parameters. For the USPS, NSSC, FEI, and Extended Yale B databases, we use the first five subjects to select the parameters, respectively. In addition, we set the range of λ and c as $[10^{-4}, 10^{4}]$.

Fig. 4 gives the performance of different methods with the parameter λ. For NSSC, when $\lambda < 1$, its optimization method usually fails to give a locally optimal solution, and it can give a good performance when $\lambda = 10$. Therefore, we fix $\lambda = 10$ for NSSC on the whole data sets. Note that our method can give the best performance when $\lambda = 10^{-4}$ on Hopkins 155 database. Hence, we fix $\lambda = 10^{-4}$ for our method on the Hopkins 155. For the USPS database, our method obtains a better performance than other methods when λ is larger than 0.01 and gives the largest CI when $\lambda = 1$. For C-Cube, our proposed method gives the best clustering result when $\lambda = 0.5$. For FEI and Extended Yale B databases, our method shows its effectiveness when λ is around 0.01. Compared with other methods, MoG can give a stable performance on

TABLE III
BEST PARAMETER FOR EACH METHOD ON DIFFERENT DATABASES

dataset	SSC (λ)	LRR (λ)	LSR (λ)	CASS (λ)	MoG (λ)	NSSC (λ)	Ours (λ, c)
Hopkins 155	0.0001	1000	0.001	0.0001	10	10	(0.0001, 0.5)
USPS	0.5	0.5	0.5	0.5	1000	10	(1,0.1)
C-Cube	0.5	0.5	1	0.5	1000	10	(0.5,0.1)
FEI	0.1	0.001	0.1	0.5	1000	10	(0.01,0.01)
Extended Yale B	0.5	1	0.01	0.001	10	10	(0.1,0.1)

TABLE IV
CLUSTERING RESULTS OF DIFFERENT ALGORITHMS ON THE HOPKINS 155 DATABASE. THE BEST RESULTS ARE IN BOLD FONT

k		Accuracy							
		Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
2 motions	Average	87.80	83.40	96.47	96.14	92.01	**98.03**	88.76	97.81
Acc.(%)	Median	88.10	83.83	99.67	99.54	99.64	**100.00**	90.23	**100.00**
3 motions	Average	77.22	74.88	90.38	90.66	89.67	94.25	78.46	**95.03**
Acc.(%)	Median	80.42	75.45	94.57	92.34	91.43	97.66	79.90	**99.17**
Total	Average	85.55	81.48	95.08	94.96	91.55	**97.21**	86.37	**97.21**
Acc.(%)	Median	85.86	80.84	99.41	99.06	97.76	99.71	88.50	**100.00**
k		Normalized Mutual Information							
		Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
2 motions	Average	53.96	40.09	86.53	79.60	70.42	86.10	57.37	**87.24**
Acc.(%)	Median	44.11	27.96	96.43	94.92	96.19	**100.00**	57.88	**100.00**
3 motions	Average	49.69	43.33	80.19	76.01	77.67	83.21	50.22	**86.61**
Acc.(%)	Median	47.93	46.90	80.14	76.86	79.47	89.17	47.34	**95.41**
Total	Average	53.26	40.96	85.17	78.85	72.14	85.50	55.78	**87.12**
Acc.(%)	Median	45.14	34.65	94.42	92.19	85.59	96.96	56.23	**100.00**

TABLE V
CLUSTERING RESULTS OF DIFFERENT ALGORITHMS ON THE USPS DATABASE. THE BEST RESULTS ARE IN BOLD FONT

k	Accuracy							
	Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
5 subjects	80.67	82.67	83.33	85.33	73.33	70.66	90.00	**92.67**
6 subjects	75.00	82.77	83.89	80.00	70.00	62.22	81.67	**87.78**
7 subjects	77.14	80.00	75.24	80.95	73.81	58.10	81.90	**83.33**
8 subjects	78.75	79.85	76.25	79.17	71.25	54.17	82.08	**86.25**
9 subjects	77.78	80.00	69.63	80.74	75.56	55.93	80.00	**85.56**
10 subjects	73.00	67.67	70.00	76.33	71.00	55.67	77.00	**81.33**
k	Normalized Mutual Information							
	Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
5 subjects	66.10	71.47	72.57	69.00	66.76	45.52	76.86	**82.86**
6 subjects	60.69	63.86	73.84	65.74	62.42	46.37	70.64	**77.56**
7 subjects	64.45	64.88	64.49	68.90	63.92	42.24	74.02	**74.63**
8 subjects	68.48	72.22	66.69	70.29	64.03	42.13	74.96	**80.07**
9 subjects	67.28	68.63	67.37	71.49	66.23	46.87	74.31	**78.87**
10 subjects	63.26	59.93	63.95	68.87	62.48	45.54	69.93	**74.86**

these five databases with respect to the parameter λ while it gives a bad CA on the USPS and FEI databases. For the USPS and Extended Yale B databases, the curve of LRR and CASS both give a bigger fluctuation. Because K-means has no parameter, its accuracy curve is a straight line. Note that the K-means algorithm gives a very low performance on the Extended Yale B database.

For the parameter c, we can see from Fig. 5 that the comparison methods have no parameter c and always give a straight line. Note that our method can give the best performance when c is smaller than 1 on the Hopkins 155, USPS, C-Cube, and Extended Yale B databases. In particular, for Extended Yale B, the accuracy of our method is almost 100%. For FEI, the accuracy of our method is highest when $c = 0.01$. Therefore, our method has the ability to achieve the best performance for the whole databases. Note that when the value of c is larger than 0 or 1, the performance of our method tends to decrease rapidly. From our objective function (15), we can see that when parameter c increases, the noise term can be very small for all situations, which directly reduces the ability of our objective function to suppress the large noise. Hence, using CLF to deal with the noise term is powerful to reduce the influence of the noise on subspace clustering. The best parameters of each method for the experiments on the whole databases are listed in Table III.

D. Experimental Results

Tables IV–VIII illustrate the experimental results of different methods on the Hopkins 155, USPS, C-Cube, FEI,

TABLE VI

CLUSTERING RESULTS OF DIFFERENT ALGORITHMS ON THE C-CUBE DATABASE. THE BEST RESULTS ARE IN BOLD FONT

k	Accuracy							
	Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
10 subjects	32.50	46.50	43.00	45.50	22.50	33.00	14.50	**51.00**
20 subjects	26.25	**44.00**	32.25	33.25	27.75	26.25	21.00	37.50
30 subjects	24.33	32.50	29.67	34.33	27.67	24.67	28.83	**35.17**
40 subjects	22.87	30.50	28.00	28.75	25.86	24.50	28.62	**32.37**
50 subjects	23.70	29.50	28.10	32.30	26.60	26.70	25.70	**32.40**
k	Normalized Mutual Information							
	Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
10 subjects	30.22	36.86	37.56	43.70	17.26	28.32	10.45	**46.61**
20 subjects	35.16	44.36	42.97	41.86	36.53	29.04	29.28	**45.19**
30 subjects	37.96	40.46	44.96	45.72	41.31	35.24	42.67	**48.51**
40 subjects	40.77	40.50	47.27	47.26	43.77	40.87	45.19	**48.79**
50 subjects	43.43	44.50	49.76	**51.80**	47.75	45.14	45.59	51.58

TABLE VII

CLUSTERING RESULTS OF DIFFERENT ALGORITHMS ON THE FEI DATABASE. THE BEST RESULTS ARE IN BOLD FONT

k	Accuracy							
	Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
5 subjects	81.43	90.00	81.43	88.57	95.71	80.00	84.29	**98.57**
10 subjects	65.00	71.43	70.71	72.14	80.00	66.43	70.00	**85.71**
15 subjects	68.57	80.00	69.05	65.23	78.57	62.38	71.90	**82.38**
20 subjects	66.79	73.93	71.43	70.00	**75.36**	65.36	71.01	72.50
30 subjects	64.52	**76.19**	59.29	65.48	67.86	66.67	66.43	69.29
40 subjects	61.07	**77.14**	57.86	64.46	65.36	63.75	66.07	66.07
k	Normalized Mutual Information							
	Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
5 subjects	80.51	82.33	77.65	81.95	93.24	69.24	76.57	**96.77**
10 subjects	70.03	70.21	76.47	74.20	77.58	63.70	73.02	**89.44**
15 subjects	79.85	79.40	77.90	69.72	83.74	66.26	78.59	**85.85**
20 subjects	79.49	77.74	81.34	74.72	**81.93**	71.80	75.14	80.64
30 subjects	77.37	**81.64**	76.76	75.27	79.99	75.65	78.59	81.22
40 subjects	78.29	**83.48**	76.54	76.08	0.7906	76.59	78.67	80.48

TABLE VIII

CLUSTERING RESULTS OF DIFFERENT ALGORITHMS ON THE EXTENDED YALE B DATABASE. THE BEST RESULTS ARE IN BOLD FONT

k	Accuracy							
	Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
5 subjects	24.06	78.75	80.63	84.36	84.06	85.00	88.44	**95.00**
8 subjects	15.63	60.74	60.55	75.78	72.46	**83.59**	58.01	**83.59**
10 subjects	13.59	60.47	60.62	66.09	75.00	62.78	49.69	**80.31**
k	Normalized Mutual Information							
	Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
5 subjects	1.24	69.51	64.39	73.10	73.17	69.22	78.20	**90.65**
8 subjects	0.69	56.78	55.68	69.27	66.90	76.78	52.72	**78.00**
10 subjects	1.20	58.66	56.15	57.81	72.50	62.78	47.69	**77.37**

and Extended Yale B databases, respectively. From Table IV, we can see that MoG and our method give the best performance on the average accuracy of the whole videos. However, the corresponding NMI of MoG is lower than our proposed method. For the three-motions situation, our method gives the best results both on the metrics CA and NMI. The medians of our method for 2 motions and total cases can reach 100%, which shows the superiority of our proposed method. Although the accuracy of MoG is slightly bigger than our method for two motions, our method gives better quality clustering results by balancing all the cases. For the USPS data set, our method outperforms other algorithms for the whole situations. In particular, for the case of 5 subjects, the accuracy of our method is more than 7% better than the second best result. Note that the K-means algorithm gives the better performance than CASS and MoG on the USPS database, which means the handwritten digit data perhaps lacks the subspace structure. Even so, our method still shows its effectiveness on this data set. For C-Cube, we can see that SSC shows good performance for 20 subjects based on CA, and LSR gives the highest NMI for 50 subjects. However, our method outperforms other methods in 8 out of 10 total cases. In particular, the CA value of our method is more than 4% higher than the second best result. From Table VII, we can see that SSC outperforms other methods for 30 and 40 subjects, and CASS gives the best performance with 20 subjects. These subspace clustering results can be attributed to the subspace preserving of sparseness. For the remaining cases, our method can achieve the best clustering results. In particular, the accuracy of our method is nearly 99% for the five subjects. Table VIII shows the

TABLE IX

COMPUTATION TIME OF DIFFERENT ALGORITHMS ON THE FEI DATA SET AS A FUNCTION OF THE NUMBER OF SUBJECTS

k	Kmeans	SSC	LRR	LSR	CASS	MoG	NSSC	Ours
5 subjects	0.03	38.27	1.09	0.04	2.27	11.38	0.13	0.32
10 subjects	0.09	84.68	1.22	0.11	10.77	72.01	0.22	0.51
15 subjects	0.18	149.33	1.71	0.17	31.13	292.68	0.36	0.84
20 subjects	0.29	239.44	2.39	0.26	60.34	728.12	0.56	1.40
30 subjects	0.63	495.24	3.87	0.57	159.61	3247.61	1.13	2.45
40 subjects	1.10	893.98	6.08	1.29	321.43	12637.33	2.22	5.25

clustering results on the Extended Yale B database. It shows that our method outperforms state-of-the-art methods for all these three clustering tasks, and MoG gives the same accuracy with our method for eight subjects. In particular, for the case of five subjects, the accuracy of our method is higher than the second best result by 10%, which is a significant improvement. Note that K-means gives a very bad performance on the Extended Yale B database which means that the performance of K-means algorithm is easily influenced by the noise in the data. As stated in Section V-A, the Extended Yale B database contains the large noise. Therefore, this experiment can further verify the effectiveness of our method in handling the noise.

In summary, our proposed method is more robust to the noise and outperforms other state-of-the-art methods on the whole databases. It is sufficient to verify that our method is capable of finding the underlying subspace structure and clustering the data points into their subspaces.

E. Computational Complexity Analysis

As shown in Algorithm 1, the computation cost of our iterative algorithm depends on the computation of $\mathbf{Z}$, Q and $\mathbf{R}$. The main computation cost of $\mathbf{Z}$ is the computation of $(Q^{t+1}\mathbf{X}^T\mathbf{X}+2\lambda\mathbf{I})^{-1}$ which is $\mathbf{O}(n^3)$. For Q, its time cost is the computation of $\|\mathbf{R}^{t+1}\|_F^2$ which is $\mathbf{O}(n^2)$. The computational cost for $\mathbf{R}$ is $\mathbf{O}(dn^2)$. Therefore, the overall time complexity of our optimization method is $\mathbf{O}(tn^3+tdn^2)$, where t denotes the number of iterations.

Furthermore, we give the computation time of different algorithms. Due to space limit, we only report the running time of all compared methods on the FEI data set, which is shown in Table IX. Note that the results are based on the codes implemented by their authors. The calculations are performed using an Intel Core i3-2130 CPU @ 3.40 GHz with 16.00-GB memory and 64-bit Windows 7 operating system. It can be seen that the computation time of LSR is lower than other subspace clustering methods. This comes from the fact that LSR can directly obtain a closed-form solution without using an iterative way. However, SSC, CASS, and MoG consume more time than other methods. In particular, for MoG, its computation time increases drastically with the number of subjects. As for LRR, NSSC, and our method, the computational cost of them is moderate for all situations.

VI. CONCLUSION

In this paper, we propose a robust subspace clustering method based on CLF. To this end, we use CLF to penalize the noise term for suppressing the large noise mixed in the real data. Due to that, the CLF's influence function has an upper bound, it can alleviate the influence of a single sample, especially the sample with a large noise, on estimating the residuals. Furthermore, we theoretically prove the grouping effect of our proposed method and present its convergence analysis. Finally, the experimental results on five real data sets reveal that our proposed method outperforms several representative methods.

REFERENCES

[1] W. Hong, J. Wright, K. Huang, and Y. Ma, "Multiscale hybrid linear models for lossy image representation," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3655–3671, Dec. 2006.

[2] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *Proc. Eur. Conf. Comput. Vis.*, Graz, Austria, May 2006, pp. 94–106.

[3] C. Lang, G. Liu, J. Yu, and S. Yan, "Saliency detection by multitask sparsity pursuit," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1327–1338, Mar. 2012.

[4] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Madison, WI, USA, Jun. 2003, pp. 11–18.

[5] G. Cui, X. Li, and Y. Dong, "Subspace clustering guided convex nonnegative matrix factorization," *Neurocomputing*, vol. 292, pp. 38–48, 31, May 2018.

[6] L. I. Smith, "A tutorial on principal components analysis," *Inform. Fusion*, vol. 51, no. 3, pp. 219–226, 2002.

[7] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, "A unified framework for representation-based subspace clustering of out-of-sample and large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2499–2512, Dec. 2016.

[8] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, no. 2, pp. 443–482, Feb. 1999.

[9] A. Adler, M. Elad, and Y. Hel-Or, "Linear-time subspace clustering via bipartite graph modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2234–2246, Oct. 2015.

[10] P. S. Bradley and O. L. Mangasarian, "K-plane clustering," *J. Global Optim.*, vol. 16, no. 1, pp. 23–32, Jan. 2000.

[11] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit.*, Colorado Springs, CO, USA, Jun. 2011, pp. 1801–1807.

[12] H. F. Bassani and A. F. R. Araujo, "Dimension selective self-organizing maps with time-varying structure for subspace and projected clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 3, pp. 458–471, Mar. 2015.

[13] Q. Lu, X. Li, Y. Dong, and D. Tao, "Subspace clustering by capped l_1 norm," in *Proc. Chin. Conf. Pattern Recognit.*, Chengdu, China, Nov. 2016, pp. 663–674.

[14] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, Dec. 2005.

[15] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 159–179, 1998.

[16] T. Zhang, A. Szlam, and G. Lerman, "Median k-flats for hybrid linear modeling with many outliers," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Sep. 2009, pp. 234–241.

[17] P. Tseng, "Nearest q-flat to m points," *J. Optim. Theory Appl.*, vol. 105, no. 1, pp. 249–252, Apr. 2000.

[18] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1546–1562, Sep. 2007.

[19] S. R. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *Proc. 21st IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[20] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2016, pp. 3918–3927.

[21] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, Jun. 2010, pp. 663–670.

[22] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. 22nd IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 2790–2797.

[23] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *Proc. 13nd IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 1615–1622.

[24] X. Li, Q. Lu, Y. Dong, and D. Tao, "SCE: A manifold regularized set-covering method for data partitioning,"*IEEE Trans. Neural Netw. Learn. Syst*, vol. 29, no. 5 pp. 1760–1773, May 2018.

[25] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: A review," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 90–105, Jun. 2004.

[26] Q. Lu, X. Li, and Y. Dong, "Structure preserving unsupervised feature selection," *Neurocomputing*, vol. 301, pp. 36–45, Aug. 2018.

[27] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *Proc. 12th Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 347–360.

[28] K. Tang, D. B. Dunson, Z. Su, R. Liu, J. Zhang, and J. Dong, "Subspace segmentation by dense block and sparse representation," *Neural Netw.*, vol. 75, pp. 66–76, Mar. 2016.

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

[30] C. Lu, J. Feng, Z. Lin, and S. Yan, "Correlation adaptive subspace segmentation by trace lasso," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 1345–1352.

[31] R. He, L. Wang, Z. Sun, Y. Zhang, and B. Li, "Information theoretic subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2643–2655, Dec. 2016.

[32] W. Jiang, J. Liu, H. Qi, and Q. Dai, "Robust subspace segmentation via nonconvex low rank representation," *Inf. Sci.*, vol. 340, pp. 144–158, May 2016.

[33] C.-G. Li and R. Vidal, "Structured sparse subspace clustering: A unified optimization framework," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Oct. 2015, pp. 277–286.

[34] Y. Fu, J. Gao, D. Tien, Z. Lin, and X. Hong, "Tensor LRR and sparse coding-based subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 2120–2133, Oct. 2016.

[35] P. Ji, M. Salzmann, and H. Li, "Efficient dense subspace clustering," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Steamboat Springs, CO, USA, Mar. 2014, pp. 461–468.

[36] B. Li, Y. Zhang, Z. Lin, and H. Lu, "Subspace clustering by mixture of Gaussian regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 2094–2102.

[37] M. Lee, J. Lee, H. Lee, and N. Kwak, "Membership representation for detecting block-diagonal structure in low-rank or sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1648–1656.

[38] Y.-X. Wang and H. Xu, "Noisy sparse subspace clustering," *J. Mach. Learn. Res.*, vol. 17, no. 12, pp. 1–41, 2016.

[39] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.

[40] X. He, D. G. Simpson, and G. Wang, "Breakdown points of t-type regression estimators," *Biometrika*, vol. 87, no. 3, pp. 675–687, Sep. 2000.

[41] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[42] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[43] H. Voß and U. Eckhardt, "Linear convergence of generalized weiszfeld's method," *Comput.*, vol. 25, no. 3, pp. 243–251, Sep. 1980.

[44] I. Singer, *Duality for Nonconvex Approximation and Optimization*. New York, NY, USA: Springer, 2007.

[45] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8.

[46] K. Zeng, J. Yu, C. Li, J. You, and T. Jin, "Image clustering by hypergraph regularized non-negative matrix factorization," *Neurocomputing*, vol. 138, pp. 209–217, Aug. 2014.

[47] F. Camastra, M. Spinetti, and A. sVinciarelli, "Offline cursive character challenge: A new benchmark for machine learning and pattern recognition algorithms," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)* , 2006, pp. 913–916.

[48] L. Fei, Y. Xu, X. Fang, and J. Yang, "Low rank representation with adaptive distance penalty for semi-supervised subspace classification," *Pattern Recognit.*, vol. 67, pp. 252–262, Jul. 2017.

[49] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[50] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[51] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang, "Diversity-induced multi-view subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 586–594.

[52] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Dec. 2002.

[53] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.

[54] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 5147–5156.

[55] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.

[56] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with ℓ^1-graph for image analysis," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 858–866, Apr. 2010.

[57] X. Li, G. Cui, and Y. Dong, "Graph regularized non-negative low-rank matrix factorization for image clustering," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3840–3853, Nov. 2017.

Xuelong Li (M'02–SM'07–F'12) is currently a Full Professor with the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.

Quanmao Lu is currently pursuing the Ph.D. degree with the Center for Optical Imagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.

His current research interests include machine learning and computer vision.

Yongsheng Dong (M'14) received the Ph. D. degree in applied mathematics from Peking University, Beijing, China, in 2012.

From 2013 to 2016, he was a Post-Doctoral Research Fellow with the Center for Optical Imagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China. From 2016 to 2017, he was a Visiting Research Fellow with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He is currently an Associate Professor with the School of Information Engineering, Henan University of Science and Technology, Luoyang, China. His current research interests include pattern recognition, machine learning, and computer vision.

Dacheng Tao (F'15) is currently a Full Professor with the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.