NONLINEAR SUBSPACE CLUSTERING

Wencheng Zhu[1,2,3], *Jiwen Lu*[1,2,3,*], *and Jie Zhou*[1,2,3]

[1]Department of Automation, Tsinghua University, Beijing, China
[2]State Key Lab of Intelligent Technologies and Systems, Beijing, China
[3]Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China
zhu1992719@foxmail.com; {lujiwen,jzhou}@tsinghua.edu.cn

ABSTRACT

This paper presents a nonlinear subspace clustering (NSC) method for image clustering. Unlike most existing subspace clustering methods which only exploit the linear relationship of samples to learn the affine matrix, our NSC reveals the multi-cluster nonlinear structure of samples via a nonlinear neural network. While kernel-based clustering methods can also address the nonlinear issue of samples, this type of methods suffers from the scalability issue. Differently, our NSC employs a feed-forward neural network to map samples into a nonlinear space and performs subspace clustering at the top layer of the network, so that the mapping functions and the clustering issues are iteratively learned. Experimental results illustrate that our NSC outperforms the state-of-the-arts.

Index Terms— Subspace clustering, neural network, nonlinear transformation, local similarity

1. INTRODUCTION

Subspace clustering has been one important visual analysis task and has many potential applications such as image and motion segmentation [1, 2], face clustering [3], and so on. The objective of subspace clustering is to partition samples into different subspaces and seek the multi-cluster structure of data [4, 5]. Over the past decade, a number of subspace clustering methods have been proposed in the literature.

Existing subspace clustering methods can be mainly divided into four categories including algebraic based, iterative based, statistical based and spectral clustering based methods [4]. Methods in the first category apply the linear algebra or polynomial algebra theories to split the data into different subspaces, where the typical methods are matrix factorization based methods and generalized PCA [6]. Matrix factorization based methods segment data by factorizing the data matrix into a low-rank matrix and a bases matrix [7]. Generalized PCA assumes that a set of polynomials of degree n can fit a union of n subspaces [6]. Methods in the second category iteratively update the clusters of the subspaces to partition the data. For example, the K-subspace method assigns a new sample to the nearest subspaces and then updates the subspaces [8]. Statistical based methods assume that the data obeys some distributions and estimate the subspaces with the distributions. Two representative methods in this category include random sample consensus [9] and agglomerative lossy compression [10]. Methods in the last category [11–15] first utilize the self-representation property which reflects the similarity structure of data to reconstruct the original data and then impose sparse, low-rank or the grouping effect constraints on the self-representation matrix [4]. Representative methods in this category include sparse subspace clustering [11], low-rank representation based subspace clustering [12], least squares regression based subspace clustering [14] and smooth representation clustering [15].

Most existing subspace clustering methods only facilitate

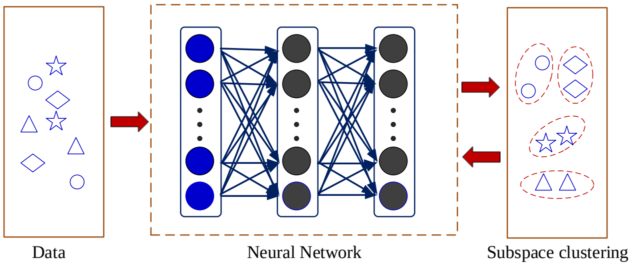


Fig. 1. The basic idea of our NSC method. We employ a feed-forward neural network to map samples into a nonlinear space and learn the self-representation matrix to perform subspace clustering at the top layer of the network. The parameters of our model are iteratively learned.

[*] Corresponding author.

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001001, the National Natural Science Foundation of China under Grants 61672306, 61572271, 61527808, 61373074 and 61373090, the National 1000 Young Talents Plan Program, the National Basic Research Program of China under Grant 2014CB349304, the Ministry of Education of China under Grant 20120002110033, and the Tsinghua University Initiative Scientific Research Program.

the linear relationship of samples to learn the affine matrix, which are not powerful enough to model the nonlinear relationship of samples, especially when images are captured in wild conditions. While kernel-based clustering methods [16, 17] can also address the nonlinear issue of samples, this type of methods suffers from the scalability issue. To address this, we propose a nonlinear subspace clustering (NSC) method for image clustering. Specifically, we employ a feed-forward neural network to map samples into a nonlinear space and learn the self-representation matrix to perform subspace clustering at the top layer of the network. The parameters of our model are iteratively learned. Fig. 1 shows the basic idea of the proposed NSC method. Experimental results illustrate that our NSC outperforms the state-of-the-arts.

2. NONLINEAR SUBSPACE CLUSTERING

In this section, we first describe the proposed model NSC and then details the optimization procedure.

2.1. Model

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ denote the data matrix, where $\mathbf{x}_i$ is the i^{th} sample of $\mathbf{X}$, the dimension of data matrix is d and the number of samples is n. $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n] \in \mathbb{R}^{n\times n}$ is the self-representation matrix. We utilize a multi-layer feed-forward neural network to map each sample $\mathbf{x}_i$ into a nonlinear feature space so that the nonlinear relationship of samples can be well discovered. Assume that there are $M+1$ layers in our NSC model, which conducts M times nonlinear transformations. To give a clear description of our model, we make some definitions. The input sample $\mathbf{x}_i$ is denoted as $\mathbf{h}_i^{(0)} = \mathbf{x}_i \in \mathbb{R}^d$ and the output of m^{th} layer is denoted as

$$\mathbf{h}_i{}^{(m)} = g\left(\mathbf{W}^{(m)}\mathbf{h}_i{}^{(m-1)} + \mathbf{b}^{(m)}\right) \in \mathbb{R}^{d_m}, \qquad (1)$$

where $m = 1, 2, ..., M$ is the number of the layer in the network, $g(\cdot)$ denotes the activation function, d_m is the dimension of the outputs in the m^{th} layer. $\mathbf{W}^{(m)} \in \mathbb{R}^{d_m\times d_{m-1}}$ and $\mathbf{b}^{(m)} \in \mathbb{R}^{d_m}$ are the weight and bias matrixes in the m^{th} layer respectively [18].

Given the data matrix $\mathbf{X}$, the output $\mathbf{H}^{(M)}$ of the top layer in the neural network is defined as

$$\mathbf{H}^{(M)} = [\mathbf{h}_1{}^{(M)}, \mathbf{h}_2{}^{(M)}, ..., \mathbf{h}_n{}^{(M)}]. \qquad (2)$$

NSC first transforms the data matrix $\mathbf{X}$ into a nonlinear space by a multi-layers feed-forward neural network to obtain $\mathbf{H}^{(M)}$ and then conducts the subspace clustering iteratively. The objective function J of NSC can be formulated as

$$\min_{\left\{\mathbf{W}^{(m)},\mathbf{b}^{(m)}\right\}_{m=1}^{M},\mathbf{C}} J = J_1 + \alpha J_2 + \beta J_3, \qquad (3)$$

where J_1 is the loss function and guarantees the rebuilding ability of the self-representation matrix in the nonlinear space, which is defined as

$$J_1 = \frac{1}{2}\sum_{i=1}^{n}\left\|\mathbf{h}_i{}^{(M)} - \mathbf{H}^{(M)}\mathbf{c}_i\right\|_F^2. \qquad (4)$$

J_2 utilizes the grouping effect and the effectiveness of the grouping effect is proved in [15], which is formulated as

$$J_2 = \frac{1}{2}tr\left(\mathbf{C}\mathbf{L}\mathbf{C}^T\right), \qquad (5)$$

where $\mathbf{L}$ is the Laplacian matrix and $\mathbf{L} = \mathbf{D} - \mathbf{S}$, $\mathbf{S}$ measures the similarity of data, $\mathbf{D}$ is the diagonal matrix with the element $D_{ii} = \sum_{j=1}^{n} S_{ij}$. J_3 is the regularization term and aims to avoid the model over-fitting, which is designed as

$$J_3 = \frac{1}{2}\sum_{m=1}^{M}\left(\left\|\mathbf{W}^{(m)}\right\|_F^2 + \left\|\mathbf{b}^{(m)}\right\|_2^2\right). \qquad (6)$$

The corresponding α and β are the positive parameters.

Then, NSC can be expressed as

$$\min_{\left\{\mathbf{W}^{(m)},\mathbf{b}^{(m)}\right\}_{m=1}^{M},\mathbf{C}} J = \left\{\begin{array}{l}\underbrace{\frac{1}{2}\sum_{i=1}^{n}\left\|\mathbf{h}_i{}^{(M)} - \mathbf{H}^{(M)}\mathbf{c}_i\right\|_F^2}_{J_1} + \\ \underbrace{\frac{\alpha}{2}tr\left(\mathbf{C}\mathbf{L}\mathbf{C}^T\right)}_{J_2} + \\ \underbrace{\frac{\beta}{2}\sum_{m=1}^{M}\left(\left\|\mathbf{W}^{(m)}\right\|_F^2 + \left\|\mathbf{b}^{(m)}\right\|_2^2\right)}_{J3}\end{array}\right\} \qquad (7)$$

2.2. Optimization

In this subsection, we present the detailed procedures of the optimization problem in (7). We update $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$ and $\mathbf{C}$ iteratively.

Update $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$: To update $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$, we fix $\mathbf{C}$, $\mathbf{H}^{(M)}$ and remove the irrelevant term to obtain the following optimization problem:

$$\min_{\left\{\mathbf{W}^{(m)},\mathbf{b}^{(m)}\right\}_{m=1}^{M}} \left\{\begin{array}{l}\frac{1}{2}\sum_{i=1}^{n}\left\|\mathbf{h}_i{}^{(M)} - \mathbf{H}^{(M)}\mathbf{c}_i\right\|_F^2 + \\ \frac{\beta}{2}\sum_{m=1}^{M}\left(\left\|\mathbf{W}^{(m)}\right\|_F^2 + \left\|\mathbf{b}^{(m)}\right\|_2^2\right)\end{array}\right\} \qquad (8)$$

The optimization problem in (8) can be solved by the subgradient descent algorithm.

We take the derivative of the objective in (8) with the parameters $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$ to zero and apply the chain rule [18–22] to acquire the following equations:

$$\frac{\partial J}{\partial \mathbf{W}^{(m)}} = \Delta^{(m)}\left(\mathbf{h}_i{}^{(m-1)}\right)^T + \beta\mathbf{W}^{(m)}, \qquad (9)$$

Algorithm 1 : NSC

```
Input:
    The data matrix $\mathbf{X} = [\mathbf{x}_1,\ \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{d \times n}$;
    The parameters $\lambda_1$ and $\lambda_2$;
Output:
    The neural network $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$ $m = 1, 2..., M$;
    The self-representation matrix $\mathbf{C}$;
    Initialize $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$, $\mathbf{H}^{(M)}$ and $\mathbf{C}$.
    Compute the Laplacian matrix $\mathbf{L}$;
    while
        Update $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ by (8);
        Compute $\mathbf{H}^{(M)}$ by (1);
        Update $\mathbf{C}$ by (14);
    end
```

$$\frac{\partial J}{\partial \mathbf{b}^{(m)}} = \Delta^{(m)} + \beta \mathbf{b}^{(m)}, \tag{10}$$

where $\Delta^{(m)}$ has the following form:

$$\Delta^{(m)} = \begin{cases} \left(\mathbf{W}^{(m+1)}\right)^T \Delta^{(m+1)} \odot g'\left(\mathbf{z}_i^{(m)}\right), m = 1, ..., M-1 \\ \left(\mathbf{h}_i^{(M)} - \mathbf{H}^{(M)}\mathbf{c}_i\right) \odot g'\left(\mathbf{z}_i^{(M)}\right), m = M \end{cases} \tag{11}$$

where $\mathbf{z}_i^{(m)} = \mathbf{W}^{(m)}\mathbf{h}_i^{(m-1)} + \mathbf{b}^{(m)}$, $g(\cdot)$ is the activation function whose derivative is $g'(\cdot)$. The operator $\odot$ means the element-wise multiplication.

Thus, the neural network can be updated by the following paradigm:

$$\begin{cases} \mathbf{W}^{(m)} = \mathbf{W}^{(m)} - \tau \frac{\partial J}{\partial \mathbf{W}^{(m)}} \\ \mathbf{b}^{(m)} = \mathbf{b}^{(m)} - \tau \frac{\partial J}{\partial \mathbf{b}^{(m)}} \end{cases} \tag{12}$$

where τ is the step size (we set $\tau = 10^{-4}$ in our experiment).

Update C: To update $\mathbf{C}$, we fix $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$ and omit unrelated items, then we get the following optimization problem:

$$\underset{\mathbf{C}}{min} \left\|\mathbf{H}^{(M)} - \mathbf{H}^{(M)}\mathbf{C}\right\|_F^2 + \alpha tr\left(\mathbf{C}\mathbf{L}\mathbf{C}^T\right), \tag{13}$$

we set the derivative of (13) with $\mathbf{C}$ to zero and have:

$$\left(\mathbf{H}^{(M)}\right)^T \left(\mathbf{H}^{(M)}\right) \mathbf{C} + \alpha \mathbf{C}\mathbf{L} = \left(\mathbf{H}^{(M)}\right)^T \left(\mathbf{H}^{(M)}\right), \tag{14}$$

the equation in (14) is the continuous Lyapunov equation, which can be solved using the MATLAB “lyap” function.

We alternately update $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$ and $\mathbf{C}$ until the objective function converges. Then, the self-representation matrix $\mathbf{C}$ is earned and we build the graph. Finally, we perform spectral clustering on the graph. The detailed algorithm of our NSC method is summarized in **Algorithm 1**.

3. EXPERIMENTS

In this section, we will present the implementation details and experimental results in our experiment.

3.1. Data Sets and Settings

We conducted experiments on two famous benchmark datasets: Extended Yale Face B [23] and USPS [24].

The Extended Yale Face B dataset [23] is a face dataset containing 38 individuals with different pose and illumination conditions and the original size is 192 × 168 pixels. The first 10 persons are used, each person has 64 frontal face images and all images are resized to 48 × 42 pixels. We used PCA to reduce the dimension of data into 170 dimensions.

The USPS dataset [24] has 9298 handwritten digit images and the size of each image is 16 × 16 pixels. For each digit, we selected the first 100 images.

We employed two evaluation criteria [15,18] including the clustering error (ce) and normalized mutual information (NMI) to evaluate the performances of different subspace clustering methods.

3.2. Parameter Settings

Following the work in [15], we built the similarity matrix $\mathbf{S}$ by using the k-nearest neighbor graph with 0-1 weights. Moreover, we set the neighbor size as 4. A small diagonal matrix $\theta\mathbf{I}$ is added to the Laplacian matrix $\mathbf{L}$ for the propose of numerical stability. $\mathbf{I}$ is the identity matrix and $0 < \theta \ll 1$ ($\theta = 0.001$ for the USPS dataset, $\theta = 0.01$ for the Extended Yale Face B dataset in our experiment) [15].

For a fair comparison, we used a ’grid-search’ approach to tune parameters in the range of $\{10^{-3}, 10^{-2}, ..., 10^{3}\}$. The *tanh* function is used as the nonlinear activation function in NSC and has the following form:

$$g(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{15}$$

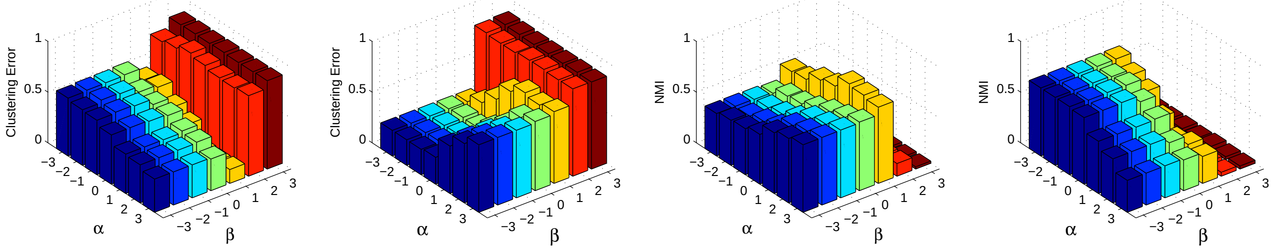
and the derivative is denoted as:

$$g'(x) = \tanh'(x) = 1 - \tanh^2(x). \tag{16}$$

where $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ are initialized as the identity matrix and the zero matrix respectively. The neural network for the Extended Yale Face B dataset has three layers with 168-100-70 neurons, α and β are set as 0.1 and 10^{-3}. We trained the neural network on the USPS dataset with three layers and the number of each layer is 256, 256 and 256, and α and β are set as 10^3 and 10.

3.3. Results and Analysis

We compared NSC with four state-of-the-art methods: SSC [11, 25], LRR [12], LSR [14] and SMR [15]. The codes of comparison algorithms are acquired from the original authors. LSR has two different forms named LSR1 and LSR2 separately. In this subsection, we conduct the experiments on two datasets and then present and investigate the experimental results clearly.



(a) USPS (b) Extended Yale B (c) USPS (d) Extended Yale B

Fig. 2. The clustering error and NMI of NSC on USPS and Extended Yale B datasets with different values of α and β: (a) the clustering error on USPS, (b) the clustering error on Extended Yale B, (c) the NMI on USPS, (d) the NMI on Extended Yale B.

Table 1. Experimental results on the Extended Yale Face B dataset (%).

Method	SSC	LRR	LSR1	LSR2	SMR	NSC
CE	33.1	38.6	30.3	26.9	26.1	**25.0**
NMI	58.4	54.5	57.8	65.3	66.1	**67.1**

Table 2. Experimental results on the USPS dataset (%).

Method	SSC	LRR	LSR1	LSR2	SMR	NSC
CE	41.5	29.3	29.4	31.1	29.5	**12.4**
NMI	59.6	67.6	68.0	66.6	69.6	**79.0**

Talbe 1 presents the clustering error and NMI of different subspace clustering methods on the Extended Yale Face B dataset. We see that NSC and SMR can hold the local similarity relationship and have good performances. Our NSC mines the nonlinear structure among data and outperforms SMR about 1.1% and 1.0% in clustering error and NMI.

Table 2 shows the clustering error and NMI of different subspace clustering methods on USPS dataset. The best results are marked in bold. As can be seen, NSC achieves best performance in clustering error and NMI. As to clustering error, NSC outperforms SSC method by 29.1% and the best comparison method by 16.9%. For NMI, NSC improves SSC method by 19.4% and the best comparison method by 9.4%.

The sensitiveness of parameters α and β is investigated. Fig. 2 presents the clustering error and NMI of NSC on the USPS and Extended Yale B datasets with different α and β. The X-axis is the parameter β, the Y-axis is the parameter α and the Z-axis represents the clustering error or NMI. For the ease of representation, we take the logarithms (base 10) of parameters. We see that NSC is not sensitive to the parameter α in clustering error and NMI. As we know, the regularization term is crucial to avoid the overfitting of models. Thus, an appropriate β can lead to a good performance and the accuracies of clustering error and NMI change sharply with the variation of β. When the β is so large or small, the model has no effect. The grouping effect is applied to guide the learning of the affine matrix and only provides the tendency. For a given β, the clustering error and NMI change slightly with different α.

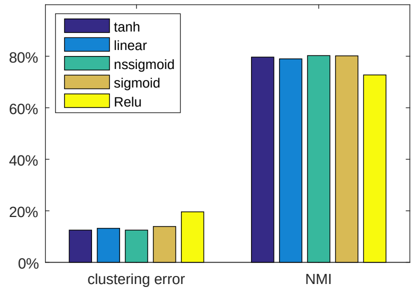


Fig. 3. The clustering error and NMI of NSC on the USPS dataset with different activation functions.

We also evaluated the performances of different nonlinear activation functions used in our NSC model such as the *tanh*, *linear*, *nssigmod* [26], *sigmod* and *ReLu* [27] functions on the USPS dataset, where the clustering performances of different methods are shown in Fig. 3 (the best results are recorded). We see that the *ReLu* activation function has the worst performance in clustering error and NMI, The activation functions *tanh* and *nssigmod* achieve the lowest clustering error and the activation function *nssigmod* acquires highest value in NMI. The difference between the activation functions *tanh* and *nssigmod* is limited.

4. CONCLUSION

In this paper, we have proposed a nonlinear subspace clustering method (NSC) for image clustering. NSC simultaneously transforms the original feature space into a nonlinear space. Experimental results have clearly shown that our NSC achieve superior results than four state-of-the-art subspace clustering methods. In the future, we are going to enforce more constraints to discover the geometrical information of samples to further improve the clusering performance.

5. REFERENCES

[1] Fabien Lauer and Christoph Schnörr, "Spectral clustering of linear subspaces for motion segmentation," in *ICCV*. IEEE, 2009, pp. 678–685.

[2] Shankar Rao, Roberto Tron, Rene Vidal, and Yi Ma, "Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories," *TPAMI*, vol. 32, no. 10, pp. 1832–1845, 2010.

[3] Shijie Xiao, Mingkui Tan, and Dong Xu, "Weighted block-sparse low rank representation for face clustering in videos," in *ECCV*, 2014, pp. 123–138.

[4] René Vidal, "Subspace clustering," *IEEESPM*, vol. 28, no. 2, pp. 52–68, 2011.

[5] Jiwen Lu, Gang Wang, and Pierre Moulin, "Localized multifeature metric learning for image-set-based face recognition," *TCSVT*, vol. 26, no. 3, pp. 529–540, 2016.

[6] Rene Vidal, Yi Ma, and Shankar Sastry, "Generalized principal component analysis (gpca)," *TPAMI*, vol. 27, no. 12, pp. 1945–1959, 2005.

[7] João Paulo Costeira and Takeo Kanade, "A multibody factorization method for independently moving objects," *IJCV*, vol. 29, no. 3, pp. 159–179, 1998.

[8] Pankaj K Agarwal and Nabil H Mustafa, "K-means projective clustering," in *ACM SIGMOD/PODS*. ACM, 2004, pp. 155–165.

[9] Martin A Fischler and Robert C Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *COMMUN ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] Yi Ma, Harm Derksen, Wei Hong, and John Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *TPAMI*, vol. 29, no. 9, 2007.

[11] Ehsan Elhamifar and René Vidal, "Sparse subspace clustering," in *CVPR*. IEEE, 2009, pp. 2790–2797.

[12] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma, "Robust recovery of subspace structures by low-rank representation," *TPAMI*, vol. 35, no. 1, pp. 171–184, 2013.

[13] Dijun Luo, Feiping Nie, Chris Ding, and Heng Huang, "Multi-subspace representation and discovery," in *ECML-PKDD*. Springer, 2011, pp. 405–420.

[14] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan, "Robust and efficient subspace segmentation via least squares regression," in *ECCV*. Springer, 2012, pp. 347–360.

[15] Han Hu, Zhouchen Lin, Jianjiang Feng, and Jie Zhou, "Smooth representation clustering," in *CVPR*, 2014, pp. 3834–3841.

[16] Vishal M Patel and René Vidal, "Kernel sparse subspace clustering," in *ICIP*. IEEE, 2014, pp. 2849–2853.

[17] Shijie Xiao, Mingkui Tan, Dong Xu, and Zhao Yang Dong, "Robust kernel low-rank representation," *TNNLS*, vol. 27, no. 11, pp. 2268–2281, 2016.

[18] Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Zhang Yi, "Deep subspace clustering with sparsity prior," in *IJCAI*, 2016, pp. 1925–1931.

[19] Junlin Hu, Jiwen Lu, and Yap-Peng Tan, "Discriminative deep metric learning for face verification in the wild," in *CVPR*, 2014, pp. 1875–1882.

[20] Jiwen Lu, Venice Erin Liong, and Jie Zhou, "Deep hashing for scalable image search," *TIP*, vol. 26, no. 5, pp. 2352–2367, 2017.

[21] Zhixiang Chen, Jiwen Lu, Jianjiang Feng, and Jie Zhou, "Nonlinear structural hashing for scalable video search," *TCSVT*, 2017.

[22] Zhixiang Chen, Jiwen Lu, Jianjiang Feng, and Jie Zhou, "Nonlinear discrete hashing," *TMM*, vol. 19, no. 1, pp. 123–135, 2017.

[23] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *TPAMI*, vol. 23, no. 6, pp. 643–660, 2001.

[24] Jonathan J. Hull, "A database for handwritten text recognition research," *TPAMI*, vol. 16, no. 5, pp. 550–554, 1994.

[25] Ehsan Elhamifar and Rene Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *TPAMI*, vol. 35, no. 11, pp. 2765–2781, 2013.

[26] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010, pp. 807–814.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.