Pattern Recognition Letters
journal homepage: www.elsevier.com

Nonlinear Subspace Clustering for Face Clustering

Wencheng Zhu[a,b,c], Jiwen Lu[a,b,c,**], Jie Zhou[a,b,c]

[a]*Department of Automation, Tsinghua University, Beijing, 100084, China.*
[b]*State Key Lab of Intelligent Technologies and Systems, Beijing, 100084, China.*
[c]*Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, 100084, China.*

ABSTRACT

We present in this paper a nonlinear subspace clustering (NSC) method for face 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. Specifically, 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. Otherwise, our NSC applys a similarity measure based on the grouping effect to capture the local structure of data. Experimental results illustrate that our NSC outperforms the state-of-the-arts.

Keywords: subspace clustering; neural network; nonlinear transformation; local similarity
© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Subspace clustering has been one important visual analysis task and has many potential applications such as image and motion segmentation (Lauer and Schnörr, 2009; Rao et al., 2010), face clustering (Xiao et al., 2014) and so on. The objective of subspace clustering is to partition samples into different subspaces and seeks the multi-cluster structure of data (Vidal, 2011). 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 (Vidal, 2011). 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 (Costeira and Kanade, 1998; Kanatani, 2001) and generalized PCA (Vidal et al., 2005). Matrix factorization based methods segment data by factorizing the data matrix into a low-rank matrix and a bases matrix (Costeira and Kanade, 1998). These methods easily fail when the subspaces are connected or influenced by noise. Generalized PCA assumes that a set of polynomials of degree n can fit a union of n subspaces (Vidal et al., 2005). GPCA can be considered as polynomial fitting, thus GPCA can handle subspaces with different dimensions and is sensitive to noise. Methods in the second category first assign the samples to subspaces and then iteratively update the clusters of the subspaces to partition the data. For example, the K-subspace method appoints a new sample to the nearest subspaces and updates the subspaces in the following (Agarwal and Mustafa, 2004). Statistical based methods suppose that the data obey Gaussian distribution and apply Expectation Maximization (EM) to estimate the probabilities of subspaces. Two representative methods in this category include random sample consensus (Fischler and Bolles, 1981) and agglomerative lossy compression (Ma et al., 2007). Methods in the last category (Elhamifar and Vidal, 2009; Luo et al., 2011; Lu et al., 2012; Hu et al., 2014; Liu et al., 2013) 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 (Vidal, 2011). Representative methods in this category include sparse subspace clustering (SSC) (Elhamifar and Vidal, 2009), low-rank representation based sub-

**Corresponding author.
e-mail: lujiwen@tsinghua.edu.cn (Jiwen Lu)

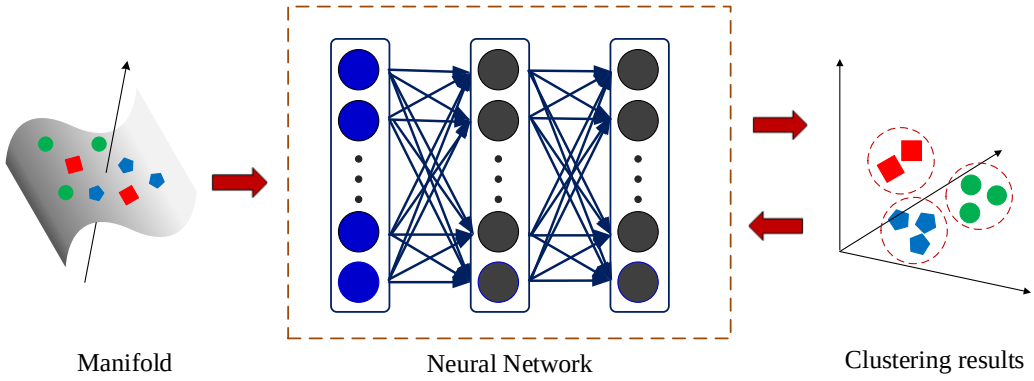


Fig. 1. The basic idea of our NSC method. We employ a feed-forward neural network to map samples into a nonlinear space and learn the self-representation matrix to perform subspace clustering at the top layer of the network. The parameters of our model are iteratively learned.

space clustering (LRR) (Liu et al., 2013), least squares regression based subspace clustering(LSR) (Lu et al., 2012) and smooth representation clustering (SMR) (Hu et al., 2014). Sparse subspace clustering (SSC) holds the assumption that each subspaces can be linear and sparse combination of other points. SSC seeks for the points in the same subspace with the sparsest representation. SSC can deal with noise and outliers (Elhamifar and Vidal, 2009). Unlike SSC obtaining the sparsest representations of candidate points, LRR finds the lowest-rank representation of each data and uncovers the correlation structure of data. Otherwise, an effective way is provided to obatin robust data segmentation (Liu et al., 2013). The block diagonal property of data between-cluster is used in SSC and LRR, however least squares regression based subspace clustering (LSR) utilizes the within-cluster correlations and segments data by using the group effect. The group effect describes that the close data have the close representations and the correlations in the real data tend to be dense and highly correlated, the F-norm should be used to group data (Lu et al., 2012). Smooth representation clustering (SMR) also applys the group effect of representation and proves the effectiveness for subpsace clustering in the self-representation model. Further, SMR proposes a new form of the group effect which is more superior than the old one (Hu et al., 2014). Spectral clustering based methods have achieved good performance, but these methods can not deal with the points between the intersection of subspaces and only utilize the linear structure of data.

Most existing subspace clustering methods tend to exploit 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. Thus, many kernel-based clustering methods appear (Patel and Vidal, 2014; Xiao et al., 2016). Kernel-based clustering methods embed the nonlinear data into a high-dimension space by a nonliner mapping, However, this type of methods suffers from the scalability issue. To address this, we propose a nonlinear subspace clustering (NSC) method for face clustering. Specifically, we employ a feed-forward neural network to transform 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. Figure 1 shows the basic idea of the proposed NSC method.

We summary our contributions as follows:

1. We propose a nonlinear subspace clustering method (NSC), which utilizes a feed-forward neural network to embed samples into a nonlinear space.
2. NSC can capture the local structure of data via the grouping effect.
3. Experimental results illustrate that our NSC outperforms the state-of-the-arts.

2. Related Work

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ denote the data matrix, $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n] \in \mathbb{R}^{n\times n}$ is the self-representation matrix, where $\mathbf{x}_i$ is the i^{th} sample of $\mathbf{X}$, the dimension of data is d and the number of data is n.

2.1. Subspace Clustering

Recently, representation based subspace clustering methods (Elhamifar and Vidal, 2009; Liu and Yan, 2011; Lu et al., 2012; Liu et al., 2013; Hu et al., 2014) have attracted many attentions and achieved superior results. The self-representation approach is used to seek the block diagonal property of data between-cluster and then conduct spectral clustering on the self-representation matrix. Generally, these methods can be formulated as:

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_l + \lambda R(\mathbf{C}) \tag{1}$$

where $\|\cdot\|_l$ defines the suitable norm of loss function, $R(\mathbf{C})$ is the regularization term and λ is a trade-off parameter.

SSC aims to find the sparse representation of data and has the fllowing form:

$$\min_{\mathbf{C}} \|\mathbf{C}\|_1 \quad s.t.\ \mathbf{X} = \mathbf{XC}\ , diag(\mathbf{C}) = 0 \tag{2}$$

The sample can be spare represented by other samples in the same subspace. SSC has difficulty dealing with the highly relevant samples as the only sample is selected. Otherwise, the l_1-norm minimization problem is time-consuming to solve (Elhamifar and Vidal, 2009).

LRR assumes that the representation of data is low-rank as well as sparse which captures the global structure of data accurately. LRR solves the following problem:

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{XC}\|_{2,1} + \lambda\|\mathbf{C}\|_* \tag{3}$$

where $\|\cdot\|_*$ is the nuclear norm defined by the sum of singular values of $\mathbf{C}$. LRR is not sensitive to noise and outliers and can be effectively solved. However, whether the affine matrix forced to be low-rank leads to the good segmentation need to be analyzed futher (Liu et al., 2013).

LSR has proved that if the objective function meets Enforced Block Diagonal (EBD) condition, the affine matrix is block diagnoal. Thus, we can take simple norm like $\|\cdot\|_F$ to restrict the objective function.

SMR is a special case of EBD condition and applies the grouping effect to explore the local structure of data. SMR considers the following minimization problem:

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{XC}\|_F^2 + \lambda tr\left(\mathbf{CLC}^T\right) \tag{4}$$

Different from these subspace clustering methods, our proposed NSC embeds the samples into a nonlinear space by nonlinear transformation. Thus, NSC can utilize the nonlinearity of data. Otherwise, the grouping effect is applied to acquire the local structure of data.

2.2. Neural Network

Due to the nonlinear transformantion of neural network, neural network especially deep neural network has achieved great success in many applications, e.g. face recognition (Ding and Tao, 2016; Ding et al., 2015; Ding and Tao, 2015, 2017) and image classification (He et al., 2016). Neural network is composed of many neural units. Each neural unit is linked to other units and has inputs, outputs, a summation function (Graupe, 2013). Multi-layers feed-forward neural network is a kind of neural network whose adjacent two layers are fully connected in the networks. In the feed-forward neural network, there is no connection in the same layer and cross-layer. The input of a neural unit is the outputs of the upper layer. The output of a neural unit is the result of the summation function with the weighted inputs. The properly adjusted multi-layers feed-forward neural network can provide the suitable nonlinear transformation of input data. Since the complex neural network tends to over-fit, we use a shallow feed-forward neural network to embed the data into a nonlinear space and then conduct subspace clustering (Schmidhuber, 2015).

3. Nonlinear Subspace Clustering

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

3.1. Model

As described above, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ denotes the data matrix, $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n] \in \mathbb{R}^{n\times n}$ is the self-representation matrix, where $\mathbf{x}_i$ is the i^{th} sample of $\mathbf{X}$, the dimension of data is d and the number of data is n. 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

$$\mathbf{h}_i{}^{(m)} = g\left(\mathbf{W}^{(m)}\mathbf{h}_i{}^{(m-1)} + \mathbf{b}^{(m)}\right) \in \mathbb{R}^{d_m}, \tag{5}$$

where $m = 1, 2, ..., M$ is the number of the layer in the network, $g(\cdot)$ denotes the activation function e.g. *tanh*, *linear*, *nssigmod* (Nair and Hinton, 2010), *sigmod* and *ReLu* (Krizhevsky et al., 2012), 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 (Peng et al., 2016).

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

$$\mathbf{H}^{(M)} = [\mathbf{h}_1{}^{(M)}, \mathbf{h}_2{}^{(M)}, ..., \mathbf{h}_n{}^{(M)}]. \tag{6}$$

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

$$\min_{\{\mathbf{W}^{(m)},\mathbf{b}^{(m)}\}_{m=1}^{M},\mathbf{C}} J = J_1 + \alpha J_2 + \beta J_3, \tag{7}$$

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

$$J_1 = \frac{1}{2}\sum_{i=1}^{n}\left\|\mathbf{h}_i{}^{(M)} - \mathbf{H}^{(M)}\mathbf{c}_i\right\|_F^2. \tag{8}$$

J_2 utilizes the grouping effect to capture the local structure of data and the effectiveness of the grouping effect is proved in (Hu et al., 2014), which is formulated as

$$J_2 = \frac{1}{2}tr\left(\mathbf{CLC}^T\right), \tag{9}$$

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

$$J_3 = \frac{1}{2}\sum_{m=1}^{M}\left(\left\|\mathbf{W}^{(m)}\right\|_F^2 + \left\|\mathbf{b}^{(m)}\right\|_2^2\right). \tag{10}$$

The corresponding α and β are the positive trade-off parameters.

Then, NSC can be expressed as

$$\min_{\{\mathbf{W}^{(m)},\mathbf{b}^{(m)}\}_{m=1}^{M},\mathbf{C}} J = \left\{ \begin{array}{l} \underbrace{\frac{1}{2}\sum_{i=1}^{n}\left\|\mathbf{h}_i{}^{(M)} - \mathbf{H}^{(M)}\mathbf{c}_i\right\|_F^2 +}_{J_1} \\ \underbrace{\frac{\alpha}{2}tr\left(\mathbf{C}\mathbf{L}\mathbf{C}^T\right)}_{J_2} + \\ \underbrace{\frac{\beta}{2}\sum_{m=1}^{M}\left(\left\|\mathbf{W}^{(m)}\right\|_F^2 + \left\|\mathbf{b}^{(m)}\right\|_2^2\right)}_{J3} \end{array} \right\}. \quad (11)$$

The proposed neural network model NSC facilitates the self-representation idea to learn the affine matrix at the top layer of network. The nonlinearity of data is exploited through neural network and the local structure is manipulated by the grouping effect.

3.2. Optimization

In this subsection, we present the detailed procedures of the optimization problem in (11). 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:

$$\min_{\{\mathbf{W}^{(m)},\mathbf{b}^{(m)}\}_{m=1}^{M}} \left\{ \begin{array}{l} \frac{1}{2}\sum_{i=1}^{n}\left\|\mathbf{h}_i{}^{(M)} - \mathbf{H}^{(M)}\mathbf{c}_i\right\|_F^2 + \\ \frac{\beta}{2}\sum_{m=1}^{M}\left(\left\|\mathbf{W}^{(m)}\right\|_F^2 + \left\|\mathbf{b}^{(m)}\right\|_2^2\right) \end{array} \right\}. \quad (12)$$

The optimization problem in (12) can be solved by the sub-gradient descent algorithm.

We take the derivative of the objective in (12) with the parameters $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$ to zero and employ the chain rule (Peng et al., 2016) to obtain the following equations:

$$\frac{\partial J}{\partial \mathbf{W}^{(m)}} = \Delta^{(m)}\left(\mathbf{h}_i{}^{(m-1)}\right)^T + \beta\mathbf{W}^{(m)}, \quad (13)$$

$$\frac{\partial J}{\partial \mathbf{b}^{(m)}} = \Delta^{(m)} + \beta\mathbf{b}^{(m)}, \quad (14)$$

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

$$\Delta^{(m)} = \left\{ \begin{array}{l} \left(\mathbf{W}^{(m+1)}\right)^T\Delta^{(m+1)} \odot g'\left(\mathbf{z}_i^{(m)}\right), m = 1, ..., M-1 \\ \left(\mathbf{h}_i{}^{(M)} - \mathbf{H}^{(M)}\mathbf{c}_i\right) \odot g'\left(\mathbf{z}_i^{(M)}\right), m = M \end{array} \right. \quad (15)$$

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:

$$\left\{ \begin{array}{l} \mathbf{W}^{(m)} = \mathbf{W}^{(m)} - \tau\frac{\partial J}{\partial \mathbf{W}^{(m)}} \\ \mathbf{b}^{(m)} = \mathbf{b}^{(m)} - \tau\frac{\partial J}{\partial \mathbf{b}^{(m)}} \end{array} \right. \quad (16)$$

where $\tau > 0$ is the step size (we set $\tau = 10^{-4}$ in our experiment).

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 α and β;
Output:
The neural network $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$ $m = 1, 2..., M$;
The self-representation matrix $\mathbf{C}$;
The clustering results;

```
Initialize W^(m), b^(m), H^(M) and C.;
Compute the Laplacian matrix L;
while not converge do
    for i = 1, 2, ...., n do
        Randomly select a sample x_i and let h_i^(0) = x_i;
        Update W^(m) and b^(m) by (12);
        Compute H^(M) by (5);
        Update C by (18);
    end
end
Build the graph G = |C| + |C^T|;
Acquire the clustering results via spectral clustering;
```

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

$$\min_{\mathbf{C}} \left\|\mathbf{H}^{(M)} - \mathbf{H}^{(M)}\mathbf{C}\right\|_F^2 + \alpha tr\left(\mathbf{C}\mathbf{L}\mathbf{C}^T\right). \quad (17)$$

We set the derivative of (17) with $\mathbf{C}$ to zero and have:

$$\left(\mathbf{H}^{(M)}\right)^T\left(\mathbf{H}^{(M)}\right)\mathbf{C} + \alpha\mathbf{C}\mathbf{L} = \left(\mathbf{H}^{(M)}\right)^T\left(\mathbf{H}^{(M)}\right), \quad (18)$$

the equation in (18) is the continuous Lyapunov equation which can be settled using the MATLAB "lyap" function.

We iteratively update $\mathbf{W}^{(m)}$, $\mathbf{b}^{(m)}$ and $\mathbf{C}$ until the objective function converges. Figure 2 shows the convergence curve on the Extended Yale Face B and AR datasets with parameters $\alpha = 1$ and $\beta = 1$. Then, the self-representation matrix $\mathbf{C}$ is gotten and we build the graph $\mathbf{G} = |\mathbf{C}| + \left|\mathbf{C}^T\right|$. Finally, we perform spectral clustering on the graph $\mathbf{G}$. The detailed algorithm of our NSC method is summarized in **Algorithm 1**.

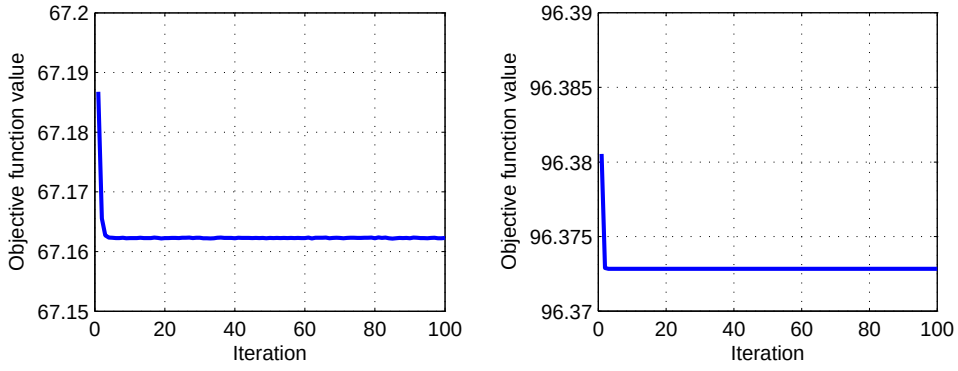
4. Experiments

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

4.1. Data Sets and Settings

We conducted experiments on two popular benchmark datasets: the Extended Yale Face B (Hu et al., 2014) and the AR (Zhang et al., 2011).

The Extended Yale Face B dataset (Georghiades et al., 2001) 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



(a) Extended Yale B (b) AR

Fig. 2. The convergence curves on the Extended Yale Face B and AR datasets with parameters $\alpha = 1$ and $\beta = 1$. (a) shows the curve on Extended Yale Face B dataset and (b) shows curve on AR dataset

Fig. 3. The Extended Yale Face B dataset. For each individual, the first 10 images are shown with different illumination conditions.

used PCA to reduce the dimension of data into 168 dimensions. Figure 3 shows the first 10 images for each individial with different illumination conditions.

The AR dataset (Zhang et al., 2011) is a famous dataset in computer vision. A subset of AR dataset is used containg 50 subjects with variation in illumination and expression. Each subject has 7 images and the size of image is 128×128 pixels. We used VGG-16 (facenet) to extract the feature with 4096 dimensions and then applied PCA to reduce the dimension to 200 dimensions.

We employed two evaluation criteria (Hu et al., 2014; Peng et al., 2016) including the clustering error (CE) and normalized mutual information (NMI) to evaluate the performances of different subspace clustering methods. The clustering error is defined as

$$CE = 1 - \frac{1}{N}\sum_{i=1}^{N}\delta\left(p_i, q_i\right), \tag{19}$$

where p_i and q_i denote predicted label and the ground truth label for i^{th} sample respectively, $\delta(\cdot)$ is the mapping function which matchs the predicted labels to ground truth labels. Following, NMI is stated as

$$NMI\left(\mathbf{X}, \mathbf{Y}\right) = \frac{1}{2}\frac{I\left(\mathbf{X}, \mathbf{Y}\right)}{H\left(\mathbf{X}\right) + H\left(\mathbf{Y}\right)}, \tag{20}$$

where $I\left(\mathbf{X}; \mathbf{Y}\right)$ and $H\left(\mathbf{X}\right)$ are denoted as Equation (21) and (22) respectively.

$$I\left(\mathbf{X}; \mathbf{Y}\right) = \sum_{y\in\mathbf{Y}}\sum_{x\in\mathbf{X}} p\left(x, y\right)\log\left(\frac{p\left(x, y\right)}{p\left(x\right)p\left(y\right)}\right) \tag{21}$$

$$H\left(\mathbf{X}\right) = -\sum_{i=1}^{n} p\left(x_i\right)\log\left(p\left(x_i\right)\right) \tag{22}$$

4.2. Parameter Settings

Following the work in (Hu et al., 2014), 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 $0 < \theta \ll 1$ (Hu et al., 2014). For a fair comparison, we used a 'grid-search' approach to tune parameters in the range

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 AR dataset (%).

Method	SSC	LRR	LSR1	LSR2	SMR	NSC
CE	18.3	24.9	16.0	17.7	13.7	**9.1**
NMI	91.4	80.1	90.7	90.7	92.1	**93.6**

of $\{10^{-3}, 10^{-2}, ..., 10^{3}\}$. The *tanh* function is used as the nonlinear activation function in NSC and has the following form:

$$g(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}, \tag{23}$$

and the derivative is denoted as

$$g'(x) = \tanh'(x) = 1 - \tanh^{2}(x), \tag{24}$$

where $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ are initialized as the identity matrix and zero matrix respectively. The neural network for the Extended Yale Face B dataset has three hidden layers with 168-100-70 neurons, α and β are set as 0.1 and 10^{-3}. We trained the neural network on AR dataset with two hidden layers and the number of each layer is 200 and 200. α and β are set as 10^{3} and 0.1.

4.3. Results and Analysis

We compared NSC with four state-of-the-art methods: SSC (Elhamifar and Vidal, 2009, 2013), LRR (Liu et al., 2013), LSR (Lu et al., 2012) and SMR (Hu et al., 2014). 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.

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 0.01 in clustering error and NMI.

Table 2 shows the clustering error and NMI of different subspace clustering methods on AR 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 LRR method by 15.8% and the best comparison method by 4.6%. For NMI, NSC improves LRR method by 0.135 and the best comparison method by 0.015.

We also investigated the sensitiveness of parameters α and β. Figure 4 presents the clustering error and NMI of NSC on the AR 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.

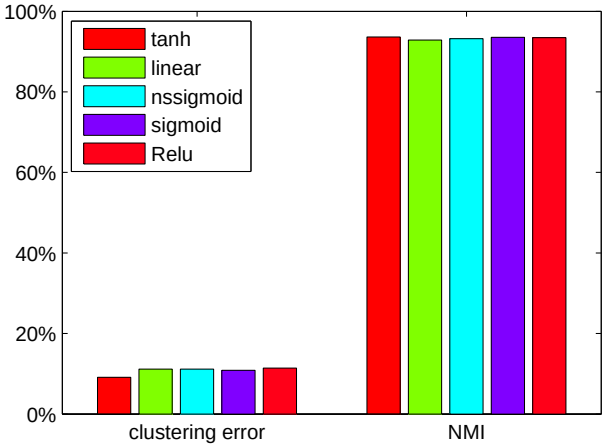


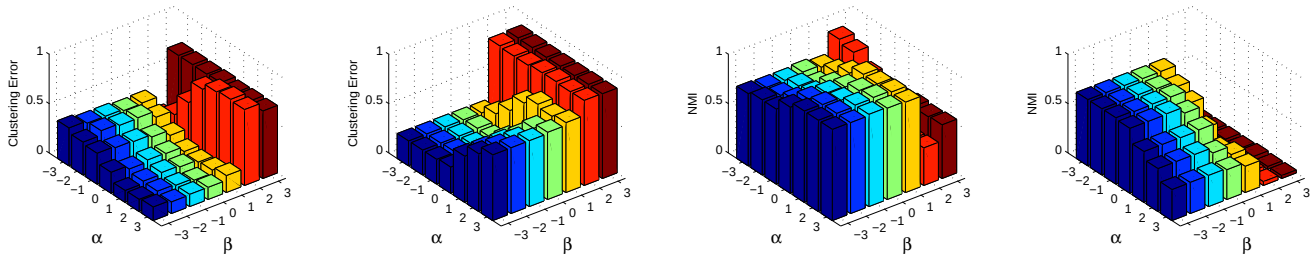
Fig. 5. The clustering error and NMI of NSC on the AR dataset with different activation functions.

We also evaluated the performances of different nonlinear activation functions used in our NSC model such as the *tanh*, *linear*, *nssigmod* (Nair and Hinton, 2010), *sigmod* and *ReLu* (Krizhevsky et al., 2012) functions on the AR dataset, where the clustering performances of different methods are shown in Figure 5 (the best results are recorded). We see that the *ReLu* activation function has the worst performance in clustering error and *linear* activation function obtains the worst performance in NMI. The activation function *tanh* achieves the lowest clustering error and highest value in NMI. The difference between the activation functions *linear* and *nssigmod* is limited.

4.4. Evaluation on the Digital Dataset

The USPS dataset (Hull, 1994) is used to evaluate the performance of NSC on digital dataset. The USPS dataset has 9298 handwritten digit images and the size of each image is 16×16 pixels. For each digit, we selected the first 100 images. The parameter settings of USPS dataset follows the parameter settings on Extended Yale Face B and AR datasets. 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 1000 and 10.

The clustering error and NMI results are shown in Table 3. As can be seen, NSC achieves best performance in clustering error and NMI. For clustering error, NSC outperforms SSC method by 29.1% and the best comparison method by 16.9%. As to NMI, NSC improves SSC method by 19.4% and the best comparison method by 9.4%. The digit dataset is utilized to validate that our proposed clustering method can handle the digit dataset as well as face datasets.



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

Fig. 4. The clustering error and NMI of NSC on AR and Extended Yale B datasets with different values of α and β: (a) the clustering error on AR, (b) the clustering error on Extended Yale B, (c) the NMI on AR, (d) the NMI on Extended Yale B.

Table 3. 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**

5. 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.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001001, in part by the National Natural Science Foundation of China under Grant 61672306, Grant 61225008, Grant 61572271, Grant 61527808, Grant 61373074, and Grant 61373090, in part by the National 1000 Young Talents Plan Program, the National Basic Research Program of China under Grant 2014CB349304, in part by the Ministry of Education of China under Grant 20120002110033, and in part by the Tsinghua University Initiative Scientific Research Program.

References

Agarwal, P.K., Mustafa, N.H., 2004. K-means projective clustering, in: ACM SIGMOD/PODS, ACM. pp. 155–165.

Costeira, J.P., Kanade, T., 1998. A multibody factorization method for independently moving objects. IJCV 29, 159–179.

Ding, C., Tao, D., 2015. Robust face recognition via multimodal deep face representation. TMM 17, 2049–2058.

Ding, C., Tao, D., 2016. Trunk-branch ensemble convolutional neural networks for video-based face recognition. arXiv preprint arXiv:1607.05427 .

Ding, C., Tao, D., 2017. Pose-invariant face recognition with homography-based normalization. PR 66, 144–152.

Ding, C., Xu, C., Tao, D., 2015. Multi-task pose-invariant face recognition. TIP 24, 980–993.

Elhamifar, E., Vidal, R., 2009. Sparse subspace clustering, in: CVPR, IEEE. pp. 2790–2797.

Elhamifar, E., Vidal, R., 2013. Sparse subspace clustering: Algorithm, theory, and applications. TPAMI 35, 2765–2781.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. COMMUN ACM 24, 381–395.

Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J., 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. TPAMI 23, 643–660.

Graupe, D., 2013. Principles of artificial neural networks. volume 7. World Scientific.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: CVPR, pp. 770–778.

Hu, H., Lin, Z., Feng, J., Zhou, J., 2014. Smooth representation clustering, in: CVPR, pp. 3834–3841.

Hull, J.J., 1994. A database for handwritten text recognition research. TPAMI 16, 550–554.

Kanatani, K.i., 2001. Motion segmentation by subspace separation and model selection, in: ICCV, IEEE. pp. 586–591.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: NIPS, pp. 1097–1105.

Lauer, F., Schnörr, C., 2009. Spectral clustering of linear subspaces for motion segmentation, in: ICCV, IEEE. pp. 678–685.

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y., 2013. Robust recovery of subspace structures by low-rank representation. TPAMI 35, 171–184.

Liu, G., Yan, S., 2011. Latent low-rank representation for subspace segmentation and feature extraction, in: ICCV, IEEE. pp. 1615–1622.

Lu, C.Y., Min, H., Zhao, Z.Q., Zhu, L., Huang, D.S., Yan, S., 2012. Robust and efficient subspace segmentation via least squares regression, in: ECCV, Springer. pp. 347–360.

Luo, D., Nie, F., Ding, C., Huang, H., 2011. Multi-subspace representation and discovery, in: ECML-PKDD, Springer. pp. 405–420.

Ma, Y., Derksen, H., Hong, W., Wright, J., 2007. Segmentation of multivariate mixed data via lossy data coding and compression. TPAMI 29.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines, in: ICML, pp. 807–814.

Patel, V.M., Vidal, R., 2014. Kernel sparse subspace clustering, in: ICIP, IEEE. pp. 2849–2853.

Peng, X., Xiao, S., Feng, J., Yau, W.Y., Yi, Z., 2016. Deep subspace clustering with sparsity prior, in: IJCAI, pp. 1925–1931.

Rao, S., Tron, R., Vidal, R., Ma, Y., 2010. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. TPAMI 32, 1832–1845.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. NEURAL NETWORKS 61, 85–117.

Vidal, R., 2011. Subspace clustering. IEEE SPM 28, 52–68.

Vidal, R., Ma, Y., Sastry, S., 2005. Generalized principal component analysis (gpca). TPAMI 27, 1945–1959.

Xiao, S., Tan, M., Xu, D., 2014. Weighted block-sparse low rank representation for face clustering in videos, in: ECCV, pp. 123–138.

Xiao, S., Tan, M., Xu, D., Dong, Z.Y., 2016. Robust kernel low-rank representation. TNNLS 27, 2268–2281.

Zhang, L., Yang, M., Feng, X., 2011. Sparse representation or collaborative representation: Which helps face recognition?, in: ICCV, IEEE. pp. 471–478.