Applied Intelligence (2023) 53:22282–22296
https://doi.org/10.1007/s10489-023-04668-4

Laplacian regularized deep low-rank subspace clustering network

Yongyong Chen[1] · Lei Cheng[1] · Zhongyun Hua[1] · Shuang Yi[2]

Accepted: 25 April 2023 / Published online: 24 June 2023

Abstract
Self-expression-based deep subspace clustering, integrating traditional subspace clustering methods into deep learning paradigm to enhance the representative capacity, has become an important branch in unsupervised learning methods. However, most existing methods investigate to impose only sparse constraint on the coefficient matrix to sparsely and independently represent all data, yet omitting another essential global low-rank prior. Meanwhile, some non-linear geometric structures within data has not been well utilized for deep subspace clustering. To conquer these challenges, this paper proposes a novel deep subspace clustering method, named Laplacian Regularized Deep Low-Rank Subspace Clustering Network (LDLRSC), in which the low-rank prior and non-linear geometric information in data are captured simultaneously. Specifically, LDLRSC utilizes the nonconvex surrogate instead of sparsity to describe the global low-rankness of the self-representation matrix. Moreover, two types of Laplacian constraints are exploited to mine the geometric structure of the data samples. Extensive experiments on the several widely-used datasets have demonstrated the effectiveness of the proposed LDLRSC over existing state-of-the-arts.

Keywords Subspace clustering · Low-rank representation · Laplacian constraint · Deep clustering · Auto-encoder

1 Introduction

Subspace clustering [1] algorithms believes that many high-dimensional data are actually distributed in low-dimensional subspaces [2]. This is because high dimensionality and redundancy make some dimensions irrelevant and the intrinsic dimensions are often lower. Currently most mainstream subspace clustering methods are based on the self-expression property, which assumes that data from the same category are in the same linear subspace [3, 4]. Due to the theoretical soundness and robustness to noises, subspace clustering has been widely used in different applications, such as face clustering [5], motion segmentation [6], movie recommendation [7], image restoration [8], hyperspectral image band selection [9]. In addition, there are a large of works about multi-view subspace clustering in recent years [10–13]. Most existing subspace clustering methods in the literature can be broadly classified into two categories: shallow models and deep models.

Considering the self-expression property that each data could be linearly expressed by the other data in the same subspace, a large number of shallow models have been developed for subspace clustering. In general, most of them followed a three-step framework [14]: (1) learning a coefficient matrix C with specific characteristic by solving a certain optimization model; (2) constructing an affinity matrix S from the learned coefficient matrix C by $S = \frac{1}{2}(|C| + |C^T|)$ or other heuristic methods to describe the pair-wise correlation between two data points; (3) performing the spectral clustering algorithm on the above affinity matrix S to yield the clustering information. The research hot among existing methods is how to learn a coefficient matrix C to well discover the membership of data samples. Several popular studies include l_1-norm regularized sparse subspace clustering (SSC) [5], l_p-norm regularized nonconvex sparse coding [15], nuclear norm regularized low-rank representation (LRR) [6], least squares regression [16], tractable schatten-p regularized LRR [17], thresholding ridge regression [18], block diagonal representation [19], scaled simplex representation [14] and their combination [20]. *Unfortunately, one main limitation of these shallow models is that they may cause performance*

✉ Zhongyun Hua
huazhongyun@hit.edu.cn

[1] The School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China

[2] Criminal Investigation School, Southwest University of Political Science and Law, Chongqing 401120, China

degradation, especially when data points are sampled from nonlinear subspaces. Thus, the data does not always conform to the linear assumption used in above subspace clustering methods. Among existing methods, there are two kinds of methods to handle the nonlinearity, i.e., kernel-based methods and deep learning-based ones. To cluster the data drawn from a union of nonlinear subspaces, kernel-based methods assumed that all data points could be linearly expressed by other ones in the projected space instead of the original input space. Following this idea, many researchers have proposed kernel versions of above shallow models, such as kernel sparse subspace clustering (KSSC) [21], kernel low-rank representation [22, 23], kernel truncated regression representation [24]. *The major challenge of kernel-based shallow models is how to select perfect kernel functions to guarantee the clustering performance.*

Integrating traditional subspace clustering methods into deep learning paradigm, numerous deep subspace clustering methods have been developed to overcome the nonlinear challenge in real applications. Following the three-step framework of shallow models, they learn the coefficient matrix C in an end-to-end learning manner. For example, Peng et al. [25] developed Deep Subspace Clustering with Sparsity Prior, in which the coefficient matrix is first learned in the original space, and then applied into the stacked auto-encoder to get new potential features. Subsequently, Ji et al. [26] excavated the self-expression property in the latent space instead of the original noisy space, in which a fully connected layer is inserted between the encoder and decoder to simulate the nature of self-expression. Continuing along this vein, numerous researchers have combined adversarial learning [27], neural collaborative learning [28], multi-level representation [29], multi-modal learning [30] into [26].

Although the improvement of the representative capacity has been made by introducing the deep learning paradigm, we observe that they may suffer from the following limitations. First, most of existing deep subspace clustering methods investigate to impose only sparse prior on the coefficient matrix. That is, the sparse representation of each data vector is calculated sparsely and independently. This is because the sparse prior is imposed on each element in a local perspective not in a global perspective. Many studies have proven that the low-rank prior may be a better regularizer than the sparse prior in image denoising [31], dynamic MRI restoration [32], background subtraction [33], snapshot compressive sensing [34] and also in shallow subspace clustering [6, 35]. The underlying reason is that the sparse representation represents all samples individually, while the low-rank representation represents all samples jointly by finding their lowest-rank representation. Unfortunately, such powerful low-rank prior has not been well utilized for deep subspace clustering. To the best of our knowledge, the only work using low-rank constraint for deep subspace clustering is DLRSC [36], which decomposes the coefficient matrix into the product of two matrices whose number of rows is much larger than that of columns, such that the rank of the coefficient matrix would be limited by the minimum of rows and columns. Different from DLRSC [36], we explore the nuclear norm instead of the matrix factorization to characteristic the low-rank prior of the coefficient matrix. Meanwhile, some non-linear geometric structures within data has not been well utilized for deep subspace clustering. For clustering task, the global characteristic and local geometric structures embedding in high-dimensional data are both significant. In summary, there is an urgent need to design one unified model to exploit the low-rank prior and non-linear geometric information simultaneously to strengthen the ability of discovering latent patterns.

To overcome the above challenges, in this paper, we propose a novel deep subspace clustering method namely Laplacian Regularized Deep Low-Rank Subspace Clustering Network (LDLRSC), which exploits the global low-rank prior and non-linear geometric information simultaneously, such that the representative capacity of the similarity matrix can be further improved. The flowchart of our LDLRSC is shown in Fig. 1. Our LDLRSC is inspired by [26, 37]. It exploits the auto-encoder to solve the non-linearity of input features, a self-expression layer between the encoder and decoder to learn favorable deep subspace clustering representations, the nuclear norm to realize the low-rank prior, and two types of Laplacian constraints to capture non-linear geometric information. Besides, we map each singular value of the coefficient matrix to a smaller interval and then perform the summation to prevent the adverse effects of extreme singular values [31]. Extensive experiments show the superiority of the proposed LDLRSC method over 12 state-of-the-art subspace clustering methods. Our contributions are summarized as follows:

- We propose a new LDLRSC method to simultaneously take the global low-rank prior and local non-linear geometric information into consideration for deep subspace clustering.
- Unlike existing deep subspace clustering methods using the sparse prior to individually represent all sample, LDLRSC utilizes the nuclear norm to carry out the low-rank prior. To preserve the local non-linear geometric information, LDLRSC uses two Laplacian regularizers.
- Extensive experiments on five widely used data have demonstrated the effectiveness of the proposed LDLRSC method compared with state-of-the-art seven shallow subspace clustering methods and five deep subspace clustering ones.

The remainder of this article is organized as follows. Section 2 briefly introduces some related methods for sub-

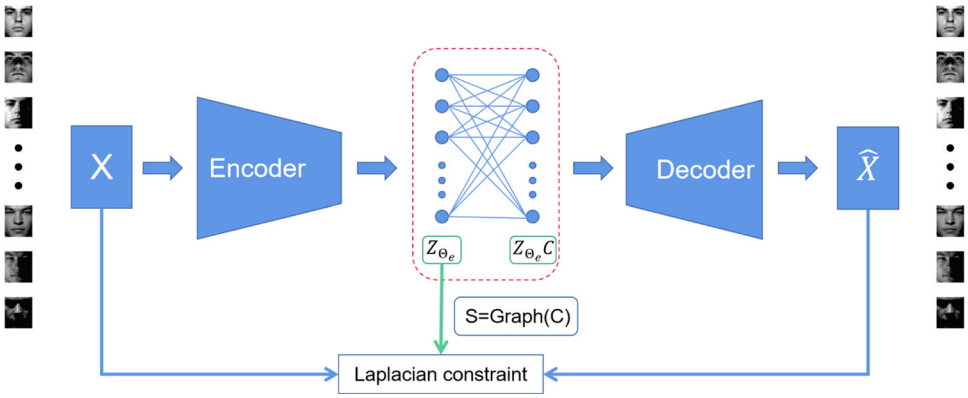

Fig. 1 Network architecture of the proposed LDLRSC network. It contains three modules: (1) convolutional encoder module: extracting deep convolutional feature Z_{Θ_e} to overcome the non-linearity challenge; (2) self-expression module: learning the self-expressive coefficient matrix C which is imposed by the low-rank prior instead of the sparse prior; (3) convolutional decoder module: recovering the latent feature Z_{Θ_e} into $\hat{X}$. To preserve the local geometric structures, LDLRSC introduces two types of Laplacian constraints. The first type Laplacian constraint is imposed on the latent features Z_{Θ_e} while the second type is imposed on the original feature X and the recovered feature $\hat{X}$

space clustering. We present the proposed LDLRSC method, conduct extensive experiments to investigate the performance of LDLRSC, and make conclusions in Sections 3, 4, and 5, respectively.

2 Related work

This section briefly reviews some related works, including shallow subspace clustering and deep subspace clustering. A review of current methods is shown in Table 1.

2.1 Shallow subspace clustering

Given a set of n data points $[x_1, x_2, \cdots, x_n] \in \mathbb{R}^{d\times n}$, we usually assume that all data points are drawn from K linear subspaces. The subspace clustering problem refers to finding these K subspaces and which subspace each data point lies in. In general, the existing shallow subspace clustering method could be divided into the following categories: algebraic methods [42], iterative methods [43], statistical methods [44], and spectral clustering-based methods [6]. Iterative method needs to alternate between assigning points to subspaces and fit a subspace to each cluster. However, this type of methods is more sensitive to initialization and typically requires to know the dimensions of each subspace in advance. Algebraic methods need to factorize the data matrix firstly, which exhibits much sensitivity to noise points and outliers. Among them, the spectral clustering-based methods have taken a dominant position in recent years. Essentially, they aim to build a similarity matrix of data vectors in a certain way to construct an affinity matrix and have stronger robustness to noise points and outliers. For example, Sparse Subspace Clustering (SSC) [5] discovered the self-expression property, that is, each data point can be linearly represented by other data points in its local subspace. Let $X = [x_1, x_2, \cdots, x_n]$ be the data matrix, self-expression property can be expressed by the following formula:

$$X = XC, \tag{1}$$

where C is the coefficient matrix (also known as self-representation matrix), which actually encodes the pairwise relationship between two samples. It has been proven in [45] that, under the condition that the subspaces are independent of each other, the coefficient matrix C has a diagonal structure (up to a certain permutations) and the block-structure of coefficient matrix C can be obtained by minimizing a certain norm of C. Then Eq. (1) can be rewritten as:

$$\min_C \|C\|_p \quad \text{s.t.} \quad X = XC, \quad diag(C) = 0, \tag{2}$$

where $\|\cdot\|_p$ represents an arbitrary matrix norm. Various norms for C have been proposed in the past few years, such as the ℓ_1 norm in Sparse Subspace Clustering (SSC) [5] and the Frobenius norm in Least-Squares Regression (LSR) [16]. In addition, the studies in [46] proposed lowest-rank representation to capture the global structures of the data X. Yang et al. further proposed a low-rank variation dictionary to fully mine the characteristics of low-rank decomposition of microarray data [47]. Mathematically, its model is formalized as the following optimization problem:

$$\min_C rank(C) \quad \text{s.t.} \quad X = XC. \tag{3}$$

Model (3) is a discrete optimization problem, which is NP-hard, and it is usually approximated as a convex optimization

Table 1 A brief review of current methods for subspace clustering

Category	Representative Work	Characteristic
Shallow method	SSC [5]	attempts to impose sparse constraint on subspace representation and proposes an efficient algorithm to solve the sparse presentation model; can handle data points near the intersection of subspaces and deal directly data nuisances
	LRR [6]	seeks the lowest-rank representation among all candidates linearly representing all data points; better captures the global structure of data
	LRLER [38]	constructs a low-rank local embedding representation model in which the local as well as global manifold structures of a dataset are concerned
	S^3C [39]	learns the affinity between data pairs and the segmentation simultaneously
Deep method	PARTY [25]	is the first attempt to apply deep learning into subspace clustering; learns to progressively map input data into nonlinear latent space with sparse subspace clustering prior
	DSC [26]	is the first attempt to directly learn the affinities between all data points within one neural network by designing a novel self-expressive layer
	S^2ConvSCN [40]	uses the current segmentation results to self-supervise the network learning
	RGRL [41]	presents a novel representation learning network guided by sample relations learned by the network itself
	DLRSC [36]	proposes to insert a fully-connected linear layer and its transpose in the network to implicitly seek a rank constraint on the learned representations

problem. Specifically, the rank function is replaced by the nuclear norm, transforming Eq. (3) into:

$$\min_C \|C\|_* \quad \text{s.t.} \quad X = XC, \tag{4}$$

where $\|\cdot\|_*$ denotes the nuclear norm [48] of a matrix, i.e., the sum of the singular values of the matrix. Generally, the equality constraint in Eq. (4) is relaxed into the following model:

$$\min_C \|C\|_* + \|X - XC\|_F^2. \tag{5}$$

By solving the optimization problem Eq. (5), a graph encoded by coefficient matrix is obtained, and finally the spectral clustering algorithm is applied to obtain the clustering result.

As aforementioned, unfortunately, the real data may not ensure the linear assumption, which may degrade the effectiveness of the above models. Though several kernel-based methods [22, 49] have been proposed to tackle the nonlinear problem, how to select perfect kernel functions to guarantee the clustering performance is still a challenging problem.

2.2 Deep subspace clustering

Deep learning has been introduced into the subspace clustering problem in recent years. Initially, the entire algorithm architecture is to learn effective latent representations for clustering. With its natural non-linear mapping characteristics, deep learning has yielded promising representation ability, and the nonlinear problem in real data can be effectively solved with deep learning paradigm. In various classic network architectures, auto-encoders are frequently adopted by subspace clustering approaches to learn latent representations or deep self-expressive representations from the input data. The deep embedding network (DEN) [50] introduces two constraints, i.e., locality-preserving and group sparsity, in its network. Among these two constraints, locality-preserving property is based on a simple fact that the similarity between two latent representations should hold if they are similar in the original space. The deep embedded clustering (DEC) [51] method proposes to adjust the encoder by minimizing KL divergence between soft assignment and target distribution.

In these early works, however, the two steps, i.e., feature learning and clustering were separated, which may led to the failure of the process of learning representation with the feedback on the clustering result. The lack of the combination of the two steps may degrade the final clustering performance. To conquer this limitation, deep subspace clustering (DSC) [26] proposed a novel network architecture in which a fully connected layer is cleverly inserted between the encoder and decoder to simulate the self-expression property, such

that the latent representation learning of the auto-encoder and the coefficient matrix learning are integrated into one unified model. The two-step learning process iterates constantly during the continuous gradient descent of the entire neural network. Continuing along this vein, several variants of DSC [26] have been developed such as [28, 40, 41]. To preserve the local structure, RGRL [41] uses the weighted reconstruction instead of self-reconstruction which is beneficial to characterize the neighborhood relations of samples. Later, generative adversarial network (which is also based on convolutional auto-encoder) has been introduced into the subspace clustering problem. Under the supervision of adversarial learning, sample representation learning and subspace clustering can further improve the clustering performance.

However, most of the deep subspace clustering algorithms only impose sparse constraints on the coefficient matrix. Since sparse representation may not capture the global structures of the data, these methods are more sensitive to noise points and outliers. On the contrary, LRR [6, 46] requires the rank of coefficient matrix to be as small as possible, which better captures the global structure of data and provides a more effective tool for robust subspace segmentation from corrupted data. To the best of our knowledge, the only deep subspace clusteringwork that uses low-rank constraints is DLRSC [36]. As mentioned above, it is implicit to impose low-rank constraint on the coefficient matrix and the number of columns is unknown in real applications.

3 The Proposed LDLRSC method

In this section, we present the proposed Laplacian Regularized Deep Low-Rank Subspace Clustering Network (LDLRSC) in details, which contains two important modules, i.e., the low-rank prior and the Laplacian terms, to capture the low-rank prior of the coefficient matrix and local non-linear geometric structures embedding in high-dimensional data.

3.1 Deep low-rank subspace clustering

Given a set of n data points $[x_1, x_2, \cdots, x_n] \in \mathbb{R}^{d\times n}$ drawn from K linear subspaces $\{S_i\}_{i=1}^{K}$. The key to self-expression-based subspace clustering methods is to construct a graph with a suitable similarity matrix to measure the pair-wise correlation between two samples. Existing deep subspace clustering methods including DSC [26] and its variants [28, 40, 41] consider the sparse prior of C without making full use of the low-rank prior. In our method, we exploit the low-rank prior by the following basic low-rank minimization model:

$$\begin{aligned} &\min_{C} rank(C) \approx \|C\|_*, \\ &\text{s.t.} \quad X = XC, \end{aligned} \tag{6}$$

where σ_i is the i-th singular value of C and $\|C\|_*$ is defined as the sum of singular values, i.e., $\|C\|_* = \sum_{i=1}^{n} \sigma_i$.

As pointed out in [31], the nuclear norm deviates significantly from the rank function when the singular values are greater than 1, indicating that they overshrink the rank component. Inspired by this observation, we choose to use the powerful γ-norm proposed in [31] instead of the traditional nuclear norm $\|C\|_*$:

$$\begin{aligned} &\min_{C} rank(C) \approx \|C\|_\gamma, \\ &\text{s.t.} \quad X = XC, \end{aligned} \tag{7}$$

where $\|C\|_\gamma = \sum_{i=1}^{n}(1 - e^{-\sigma_i(C)/\gamma})$, $e^{(\cdot)}$ denotes the exponential function, and γ is a super parameter greater than 0.

In terms of network architecture selection, we follow the convolutional auto-encoder as DSC [26] did. Motivated by the huge success of the convolutional auto-encoder [26], we adopt a deep neural network in our LDLRSC method. Specifically, as shown in Fig. 1, we integrate a self-expression layer into a deep auto-encoder between the encoder and the decoder. In our model, input data is mapped onto a non-linear latent space, self-expressed in this space, and, again, mapped onto the original space. The encoder can be regarded as a function that has the ability of dimensionality reduction and nonlinear conversion at the same time, denoted by F. The decoder is used to reconstruct the input features, denoted by G. This means that the encoder is used to map the original data onto a non-linear latent space and thus the nonlinearity of high-dimensional data could be solved. Suppose their parameters are represented by Θ_e and Θ_d respectively, then the basic model of the auto-encoder can be formalized as follows:

$$\min_{\Theta_e,\Theta_d} \sum_{i=1}^{n} ||X_i - G_{\Theta_d}(F_{\Theta_e}(X_i))||_2^2. \tag{8}$$

Combining the relaxed form of Eq. (7) and convolutional auto-encoder loss Eq. (8), our tentative loss function can be written as follows:

$$\begin{aligned} \min_{\Theta_e,\Theta_d,C} &\sum_{i=1}^{n} ||X_i - G_{\Theta_d}(F_{\Theta_e}(X_i))||_2^2 + \|C\|_\gamma \\ &+\|Z_{\Theta_e} - Z_{\Theta_e}C\|_F^2, \end{aligned} \tag{9}$$

where variable C denotes the coefficient matrix which is learned based on the discriminative features Z_{Θ_e} learned by deep encoder instead of the original (noisy) data. Unifying parameters Θ_e and Θ_d as Θ, we have:

$$\min_{\Theta,C} \sum_{i=1}^{n} ||X_i - \hat{X}_i||_2^2 + \alpha_1 \|C\|_\gamma + \alpha_2 \|Z_{\Theta_e} - Z_{\Theta_e} C\|_F^2, \quad (10)$$

where $\hat{X}_i = G_{\Theta_d}(F_{\Theta_e}(X_i))$. The first term of Eq. (10) aims to measure the differences between the input feature X and the recovered feature $\hat{X}$.

3.2 Laplacian constraint

However, most existing deep subspace clustering methods and our tentative model in Eq. (10) explore some specific characteristics such as sparsity and low-rankness, ignoring the geometric structures within data. Inspired by the manifold learning [52], we may further improve the clustering performance by fully mining the geometric structure of input data. That is, we aim to incorporate the Laplacian regularization into our tentative loss function in Eq. (10) so that similar data points have similar representation coefficients. Here, we consider two types of Laplacian constraints.

The first one is based on the manifold hypothesis [37], that is, if two data points x_i and x_j are close in the inner geometry of the data distribution, their embedding/mapping in the new space is also close to each other. Therefore, when the vectors in the low-dimensional latent space are used to describe each vertex in the graph, these vectors should maintain the affinity between the vertices. In mathematics, this idea can be expressed as:

$$\min \sum_{i,j} S_{ij} \|Z_{\Theta_e,i} - Z_{\Theta_e,j}\|_2^2, \quad (11)$$

where $Z_{\Theta_e,i}$ and $Z_{\Theta_e,j}$ are the i-th and j-th columns of Z_{Θ_e}, or the mapppings of x_i and x_j under some transformation (Here, we adopt the encoder in our proposed architecture, and thus z_i and z_j), respectively. S_{ij} is the similarity between samples $Z_{\Theta_e,i}$ and $Z_{\Theta_e,j}$.

For the second type, we noticed that the auto-encoder is a process of mapping the input to the latent feature space and then reconstructing it back. In the usual practice, each output data point is reconstructed by itself, which may not be enough to reflect the local structure between the data points. Therefore, we propose to use the weighted reconstruction strategy [41], i.e.,

$$\sum S_{ij} \|X_i - \hat{X}_j\|_2^2. \quad (12)$$

It is easy to derive that Eq. (11) can be rewritten as

$$\min Tr(Z_{\Theta_e} L Z_{\Theta_e}^T) \quad (13)$$

and Eq. (12) can be transformed as

$$\begin{aligned} \sum S_{ij} \|X_i - \hat{X}_j\|_2^2 &= \sum S_{ij} \left(\|X_i\|_2^2 - 2X_i^T \hat{X}_j + \|\hat{X}_j\|_2^2 \right) \\ &= \sum S_{ij} \Big(\left(\|X_i\|_2^2 - 2X_i^T \hat{X}_i + \|\hat{X}_j\|_2^2 \right) \\ &\qquad + 2 \left(X_i^T \hat{X}_i - X_i^T \hat{X}_j \right) \Big) \\ &= Tr((X - \hat{X}) D (X - \hat{X})^T) \\ &\quad + 2Tr(XL\hat{X}^T), \end{aligned} \quad (14)$$

where $Tr(\cdot)$ denotes the trace of a matrix; diagonal matrix $D = Diag(\sum_{j=1}^{n} S_{ij})$ and $L = D - S$. The similarity matrix S can be obtained simply by $S = \frac{1}{2}(|C| + |C^T|)$, which is dynamic, rather than using pre-defined values like many existing works. Using normalized degree matrix $D_n = I$ and normalized Laplacian matrix $L_n = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$, we have $Tr[(X - \hat{X}) D_n (X - \hat{X})^T] = ||X - \hat{X}||_F^2$. In summary, it can be seen clearly that the first type preserves the local non-linear geometric structures by imposing the Laplacian constraint on the encoder features Z_{Θ_e} while the second type is through imposing the Laplacian constraint on the original feature X and the recovered feature $\hat{X}$.

3.3 The proposed LDLRSC

Combining the above tentative loss function in Eq. (10) and two Laplacian constraints in Eqs. (13) and (14), we can obtain the final loss function:

$$\begin{aligned} \mathcal{L}(\Theta, C) = &\sum_{i=1}^{n} ||X_i - \hat{X}_i||_2^2 + \alpha_1 \|C\|_\gamma + \alpha_2 \|Z_{\Theta_e} - Z_{\Theta_e} C\|_F^2 \\ &+ \alpha_3 Tr(Z_{\Theta_e} L_n Z_{\Theta_e}^T) + \alpha_4 Tr(X L_n \hat{X}^T), \end{aligned} \quad (15)$$

where Θ denotes the network parameters including encoder parameters Θ_e and decoder parameters Θ_d; C denotes self-expression layer parameters; $\alpha_k (k = 1, 2, 3, 4)$ is the balance parameter; diagonal matrix $D = Diag(\sum_{j=1}^{n} S_{ij})$ and $L_n = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ is the normalized Laplacian matrix. For the sake of convenience, we introduce a few sysbomls: $\mathcal{L}_1 = \sum_{i=1}^{n} ||X_i - \hat{X}_i||_2^2 + \alpha_1 \|C\|_\gamma + \alpha_2 \|Z_{\Theta_e} - Z_{\Theta_e} C\|_F^2$, $\mathcal{L}_2 = Tr(Z_{\Theta_e} L_n Z_{\Theta_e}^T)$, $\mathcal{L}_3 = Tr(X L_n \hat{X}^T)$. Then loss function Eq. (15) can be rewritten as:

$$\mathcal{L}(\Theta, C) = \mathcal{L}_1 + \alpha_3 \mathcal{L}_2 + \alpha_4 \mathcal{L}_3. \quad (16)$$

Note that each of the above equations can be solved for differentiation, so it can be optimized by gradient descent algorithm. Once the self-expression matrix C is obtained using Eq. (15), it will be used to construct the affinity matrix A for the spectral clustering algorithm [53]. The simple way is the average strategy as in [6, 35]. That is, the affinity matrix A is yielded by $A = \frac{1}{2}(|C| + |C^T|)$. To further enhance the block structure of the similarity matrix A, in this paper, we adopt the heuristic (shown in Algorithm 1) employed by SSC [5] and EDSC [45], which has been proved beneficial for clustering. Finally, we obtain the clustering results by the popular spectral clustering algorithm. Note that in Algorithm 1, β is empirically selected according to the level of noise and d is the maximal intrinsic dimension of subspaces [41]. The whole process of the proposed LDLRSC is summarized in Algorithm 2.

Algorithm 1 Heuristic for affinity matrix construction.

Require: The relation matrix, C;
The number of cluster, k;
The intrinsic dimension of subspaces, d;
Ensure: The affinity matrix, A;
Let $S = \frac{1}{2}(|C| + |C|^T)$;
Compute the SVD of S, $S = U\Sigma V^T$;
Let $Z = U_m \Sigma_m^{\frac{1}{2}}$, where $m = k * d + 1$;
Compute affinity matrix $A = [ZZ^T]^\beta$;

Algorithm 2 LDLRSC for deep subspace clustering.

Require: Input data X;
Maximum iteration T_{max};
Trade-off parameters $\alpha_1, \alpha_2, \alpha_3, \alpha_4$;
The number of cluster K;
Ensure: Clustering result L;
Pre-train the convolutional auto-encoder;
Initialize the self-expression layer;
while $t \leq T_{max}$ **do**
  Calculate similarity matrix $S = \frac{1}{2}(|C| + |C^T|)$ and normalized Laplacian matrix $L_n = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$;
  Calculate the loss (15) and its gradient;
  Do forward propagation;
end while
Run Algorithm 1 to obtain affinity maxtrix A;
Run spectral clustering to get the clustering results.

3.4 Training strategy

For a fair comparison, we follow the pre-training and fine-tuning strategy applied by DSC in all experiments. Specifically, during the pre-training phase, the deep auto-encoder is pre-trained without using a self-expression layer for all data, then the trained parameters will be used to initialize the network. In the fine-tuning stage, we use all the data as a batch to minimize the loss function (15) with a gradient descent method for each epoch. After training, we use the coefficient matrix C (i.e., the parameters of the self-expression layer) to construct the affinity matrix through the heuristics as shown in Algorithm 1. The final clustering results are then obtained by the spectral clustering algorithm.

4 Experiment

We implement our LDLRSC method on TensorFlow-2 in python and evaluate its performance on five widely used benchmark datasets. In this section, we first provide more details for each testing dataset separately, then report the clustering performance of all competing methods, and finally conduct ablation experiments to investigate the contributions of the low-rank prior and the two Laplacian constraints. The source code will be released on the author's webpage[1].

4.1 Datasets

We conduct experiments on three benchmark face datasets: Extended YaleB (EYaleB)[2], ORL[3], Umist[4]; and two object datasets: COIL20[5] and MNIST[6]. EYaleB contains 38 subjects, each subject containing 64 images taken under different lighting. ORL consists of 40 subjects, each of which has 10 images taken in different poses and facial expressions. Umist contains only 20 persons, each person with 24 images taken in different poses. COIL20 has 20 classes of toys, and each class has 72 images. MNIST is a classic handwritten digit data set. Here we only take the first 100 pictures for each digit. Figure 2 shows several examples of the above five datasets whose main statistics are summarized in Table 2.

4.2 Evaluation metrics

We adopt three widely used metrics to evaluate the clustering performance: accuracy (ACC), normalized mutual information (NMI), and purity (PUR). The details of ACC, NMI, and PUR are described as follows.

Accuracy is the most common evaluation metric, which is defined as:

$$ACC = \max_m \frac{\sum_{i=1}^n \mathbf{1}\{\mathbf{y_i} = \mathbf{m}(\mathbf{l_i})\}}{n}, \tag{17}$$

[1] https://github.com/csanshi/LDLRSC
[2] http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html
[3] http://www.uk.research.att.com/facedatabase.html
[4] https://paperswithcode.com/dataset/umist-1
[5] http://www.cs.columbia.edu/CAVE/software/softlib/
[6] http://yann.lecun.com/exdb/mnist/

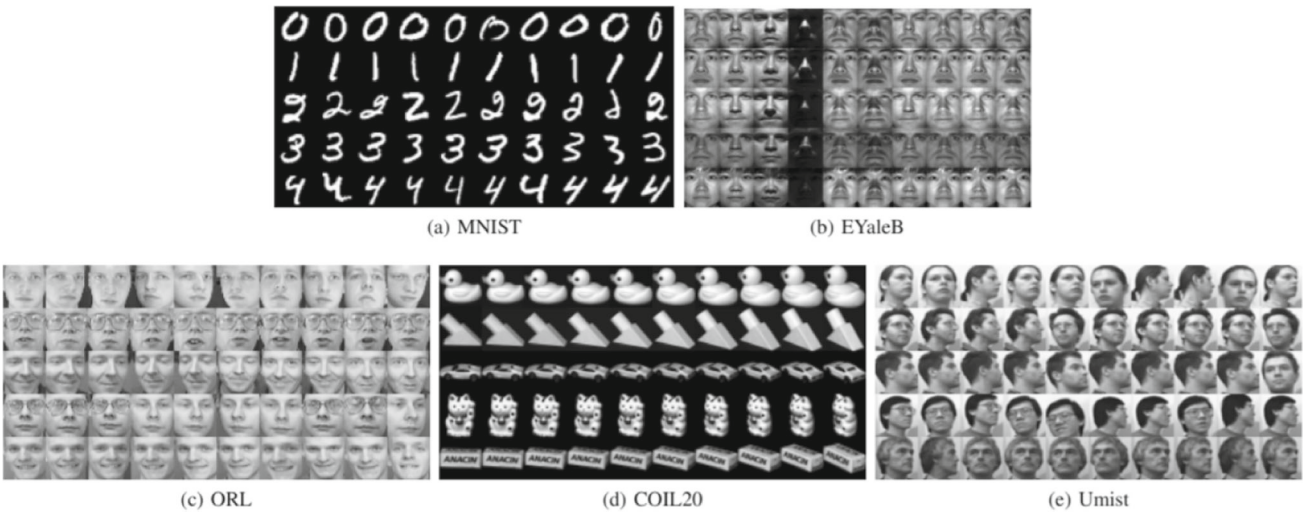

Fig. 2 Several examples of the five benchmark datasets

where y_i and l_i are the ground-truth label and the predicted label for sample x_i, respectively. $\mathbf{1}$ is a indicator function. m is the one-to-one mapping function of the label. The best mapping can be obtained by Kuhn-Munkres algorithm [54].

Normalized mutual information (NMI) is the second widely used evaluation metric for clustering task, which is defined as follows:

$$\mathrm{NMI}(y, l) = \frac{I(y, l)}{\sqrt{H(y)H(l)}}, \tag{18}$$

where y is the ground truth label and l denotes clustering label. H is the entropy and I is the mutual information measuring the information gain towards the true partition after knowing the segmentation. $\sqrt{H(y)H(l)}$ is the normalized mutual information.

Purity is a measure of the extent to which clusters contain a single class [55]. Formally, let y and l be some set of clusters and some set of classes, respectively, purity is defined as:

$$\mathrm{PUR}(y, l) = \frac{\sum_{i=1}^{k} \max_j |l_i \bigcap y_j|}{N}. \tag{19}$$

4.3 Experimental settings

Following DSC [26], our LDLRSC adopts the similar network architecture in all experiments. Specifically, for the input data X, these high-dimensional data are transformed into a latent representation Z_{Θ_e} by the encoder, and then passed through a fully connected layer without activation function and bias. Thus, there exists a fully connected layer between the encoder and the decoder to simulate the nature of self-expression, i.e., $Z_{\Theta_e} = Z_{\Theta_e}C$. Next, $Z_{\Theta_e}C$ reconstructs the input through a decoder that is symmetrical to the encoder. Among them, the encoder and decoder are both stacked convolutional layers, and different datasets have different configurations, e.g. different network depths and number of channels. We use Adam [56] as the optimizer. The learning rate of the network is set to $1e-3$, and ReLU is used for the activation function in each layer except for the self-expression layer. In the experiment within two Laplace constraints, the hyperparameter includes the penalty factor of each part of the network loss function, and there are 5 items in total. However, in the actual experiment, any one of them can be fixed, and only the other 4 parameters need to be adjusted. In this way, there are only 4 parameters for adjustment. We tune these parameters by grid search to the achieve best results on each dataset. The parameter values for all datasets are summarized in Table 3. More details of pre-training, please refer to Section 3.4.

Experimental details for each dataset are given as follows: **MNIST**: The MNIST is a classic handwritten digit image dataset. In our experiment, we select the first 100 samples o each digit from MNIST. For parameter setting, we set

Table 2 Statistics of the datasets

Dataset	MNIST	EYaleB	ORL	COIL20	Umist
# of samples	1,000	2432	400	1,440	480
# of subjects	10	38	40	20	20
# of dimensions	28×28	48×42	32×32	32×32	32×32

Table 3 Hyperparameter values for all datasets

	α_1	α_2	α_3	α_4
MNIST	0.8	1.2	0.001	0.005
EYaleB	7	18	0.000008	0.00005
ORL	6	1	0.00005	0.002
COIL20	100	25	0.04	0.005
Umist	0.5	15	0.0025	0.75

$\alpha_1 = 0.8, \alpha_2 = 1.2, \alpha_3 = 0.001, \alpha_4 = 0.005$. And we set $\alpha_3 = 0.001, \alpha_4 = 0.2$ for ablation experiment. γ is set to 0.1. **EYaleB**: The EYaleB dataset is a popular dataset for clustering task which contains 2432 images of size 192×168 from 38 different subjects ($K = 38$). Each subject is composed of 64 samples taken under various illuminations and poses. Following the experimental setting of DSC [26], we down-sampled the original face images to 48×42 pixels. We set the trade-off parameters to $\alpha_1 = 7, \alpha_2 = 18, \alpha_3 = 0.000008, \alpha_4 = 0.00005$. In ablation experiment, we set $\alpha_3 = 0.001$, $\alpha_4 = 0.2$. γ is set to 0.5. **ORL**: The ORL dataset consists of 400 human face images with 40 subjects. Each subject is composed of 10 images which are taken under varying illuminations with different facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses) [57]. The original images are down-sampled to 32×32. Compared to EYaleB, the ORL dataset is more challenging for subspace clustering because of more non-linearity in it due to various facial expressions. Besides, the small number of the samples will increase the training difficulty for deep learning. For fairness considerations, we adopt the same network configuration as DSC [26]. The specification of network is shown in Table 4. For the trade-off parameters, we set $\alpha_1 = 6, \alpha_2 = 1, \alpha_3 = 0.00005, \alpha_4 = 0.002$. In ablation experiment, we set $\alpha_3 = 0.001, \alpha_4 = 0.2$. γ is set to 0.01. **COIL20**: To further verify the superiority of our proposed method, we conduct experiments on dataset COIL20 which contains 1440 gray-scale images of 20 objects. Each image was down-sampled to 32×32. For the trade-off parameters, we set $\alpha_1 = 100, \alpha_2 = 25, \alpha_3 = 0.04, \alpha_4 = 0.005$. In ablation experiment, we set $\alpha_3 = 0.001, \alpha_4 = 0.2$. γ is set to 0.4. **Umist**: For the Umist dataset, it contains 20 subjects and each one only consists of 24 samples. Same to ORL, the small number of the samples makes it more challenging to find a proper division. Meanwhile, images in Umist are taken under different poses, which further increase the difficulty of clustering. For the trade-off parameters, we set $\alpha_1 = 0.5$, $\alpha_2 = 15$, $\alpha_3 = 0.0025$, $\alpha_4 = 0.75$. In ablation experiment, we set $\alpha_3 = 0.001$, $\alpha_3 = 0.2$. γ is set to 0.01.

The statistics of the dataset are summarized in Table 2. All network settings on these five datasets are shown in Table 4.

4.4 Experimental results

We compare the proposed LDLRSC with seven shallow subspace clustering methods: SSC [5], ENSC [58], KSSC [21], SSC-OMP [59], EDSC [45], LRR [46], LRSC [20], and six deep subspace clustering methods: AE+SSC, DEC [51], DSC [26], RGRL [41], DLRSC [36], GSA [60]. Note that both DSC and RGRL have two variants, they are Deep Subspace Clustering Networks with l_1-norm (DSC-L1), Deep Subspace Clustering Networks with l_2-norm (DSC-L2), Relation-Guided Representation Learning with l_1-norm (RGRL-L1), and Relation-Guided Representation Learning with l_2-norm (RGRL-L2). The reasons of selecting these competitors are as follows: (1) these shallow and deep subspace clustering methods are state-of-the-art clustering ones; (2) DSC, RGRL, DLRSC and our LDLRSC follow the similar network configuration; (3) DSC and RGRL are the representative sparse prior-based deep subspace clustering while DLRSC is the representative low-rank-based deep subspace clustering one. DLRSC uses the matrix factorization to carry out the low-rank prior while our LDLRSC uses the γ-norm. For Deep Low-Rank Subspace Clustering (DLRSC) [36], we use the source codes released by the author and tune its parameters by grid search to the achieve best results on MNIST, ORL and Umist. For the competitor methods, we directly collect the evaluation metrics from the corresponding papers and some existing literature [26, 41].

Table 4 Network settings, including the "kernel size@channels" and size of C

	MNIST	EYaleB	ORL	COIL20	Umist
encoder	5×5@15	5×5@10	5×5@5	3×3@15	5×5@20
	3×3@10	3×3@20	3×3@3	-	3×3@10
	3×3@5	3×3@30	3×3@3	-	3×3@5
C	1000×1000	2432×2432	400×400	1440×1440	480×480
decoder	3×3@5	3×3@30	3×3@3	3×3@15	3×3@5
	3×3@10	3×3@20	3×3@3	-	3×3@10
	5×5@15	5×5@10	5×5@5	-	5×5@20

Table 5 Clustering results of our method and compared methods on all datasets

Dataset	Metric	SSC	ENSC	KSSC	SSC-OMP	EDSC	LRR	LRSC	AE+SSC	DEC	DSC-L1	DSC-L2	RGRL-L1	RGRL-L2	DLRSC	GSA	ours(LDLRSC)
	ACC	0.4530	0.4983	0.5220	0.3400	0.5650	0.5386	0.5140	0.4840	0.6120	0.7280	0.7500	0.8130	**0.8140**	0.7410	0.6100	0.7770
MNIST	NMI	0.4709	0.5495	0.5623	0.3272	0.5752	0.5632	0.5576	0.5337	0.5743	0.7217	0.7319	0.7534	**0.7552**	0.6523	0.5808	0.6901
	PUR	0.4940	0.5483	0.5810	0.3560	0.6120	0.5684	0.5550	0.5290	0.6320	0.7890	0.7991	0.8150	**0.8160**	0.7450	0.6290	0.8040
	ACC	0.7354	0.7537	0.6921	0.7372	0.8814	0.8499	0.7931	0.7480	0.2303	0.9681	0.9733	0.9757	0.9753	0.9753	0.9428	**0.9786**
EYaleB	NMI	0.7796	0.7915	0.7359	0.7803	0.8835	0.8636	0.8264	0.7833	0.4258	0.9687	0.9703	0.9668	0.9661	0.9603	0.9519	**0.9711**
	PUR	0.7467	0.7654	0.7183	0.7542	0.8800	0.8623	0.8013	0.7597	0.2373	0.9711	0.9731	0.9757	0.9753	0.9704	0.9449	**0.9786**
	ACC	0.7425	0.7525	0.7143	0.7100	0.7038	0.8100	0.7200	0.7563	0.5175	0.8550	0.8600	0.8650	0.8700	0.8500	0.8200	**0.8750**
ORL	NMI	0.8459	0.8540	0.8070	0.7952	0.7799	0.8603	0.8156	0.8555	0.7449	0.9023	0.9034	0.9169	0.9215	0.9141	0.9039	**0.9320**
	PUR	0.7875	0.7950	0.7513	0.7463	0.7138	0.8225	0.7542	0.7950	0.5400	0.8585	0.8625	0.8775	0.8850	0.8675	0.8450	**0.8900**
	ACC	0.8631	0.8760	0.7087	0.6410	0.8371	0.8118	0.7416	0.8711	0.7215	0.9314	0.9368	0.9694	0.9701	0.9708	0.8548	**0.9778**
COIL20	NMI	0.8892	0.8952	0.8243	0.7412	0.8828	0.8747	0.8452	0.8990	0.8007	0.9353	0.9408	0.9748	0.9762	0.9694	0.9556	**0.9758**
	PUR	0.8747	0.8892	0.7497	0.6667	0.8585	0.8361	0.7937	0.8901	0.6931	0.9306	0.9397	0.9694	0.9701	0.9701	0.8916	**0.9778**
	ACC	0.6904	0.6931	0.6531	0.6438	0.6937	0.6979	0.6729	0.7042	0.5521	0.7242	0.7312	0.8104	0.8104	0.8479	0.7354	**0.9333**
Umist	NMI	0.7489	0.7569	0.7377	0.7068	0.7522	0.7630	0.7498	0.7515	0.7125	0.7556	0.7662	0.8812	0.8812	0.9288	0.8419	**0.9463**
	PUR	0.6554	0.6628	0.6256	0.6171	0.6683	0.6670	0.6562	0.6785	0.5917	0.7204	0.7276	0.8354	0.8354	0.8625	0.7833	**0.9333**

Clustering results of all methods on these five real-world datasets are recorded in Table 5, in which the best results are highlighted in bold. As observed, our LDLRSC method achieves the best performance among all competing methods in most cases. More specifically, we have the following observations:

- On EYaleB, ORL, COIL20 and Umist datasets, our proposed LDLRSC is the best method while on MNIST dataset, it is the second-best method among all competitors. Meanwhile, our proposed LDLRSC performs better than DSC on all datasets for all metrics. Specifically, our proposed method performs better than DSC on MNIST, EYaleB, ORL, COIL20 by 2.7%, 0.53%, 1.5% and 4.1% in terms of ACC, respectively. It benefits from the preservation of global low-rank prior and local geometric information.
- It is worth pointing out that our improvement on the Umist dataset is quite impressive. Our LDLRSC achieves 8.54%, 1.75%, 7.08% improvement over the second-best DLRSC method on Umist dataset in terms of ACC, NMI, and Purity, respectively. DLRSC [36] also imposes low-rank constraint on the presentation matrix while the solving process is indirect. These show the importance of local geometric structures of high-dimensional data.
- The recently proposed RGRL achieves the best performance on MNIST dataset while is relatively worse than our LDLRSC on the remaining four testing datasets especially on the Umist dataset. The improvement of ACC is around 10%. This directly demonstrates the effectiveness and superiority of the low-rank prior than the sparse prior since RGRL also adopted the Laplacian terms for preserving the local geometric structure.
- Most deep subspace clustering methods show better performance than shallow subspace clustering methods. For example, DSC, RGRL, DLRSC and our LDLRSC outperform all shallow subspace clustering methods on all datasets. This is due to the powerful feature representation and learning capabilities of deep learning.

4.5 Parameter analysis

Our LDLRSC method involves four free parameter, i.e., α_1, α_2, α_3, α_4. Figure 3 shows the ACC values with different combinations of those four parameters. We can see that our LDLRSC method achieves satisfactory performance within a large range of those four parameters. Specifically,

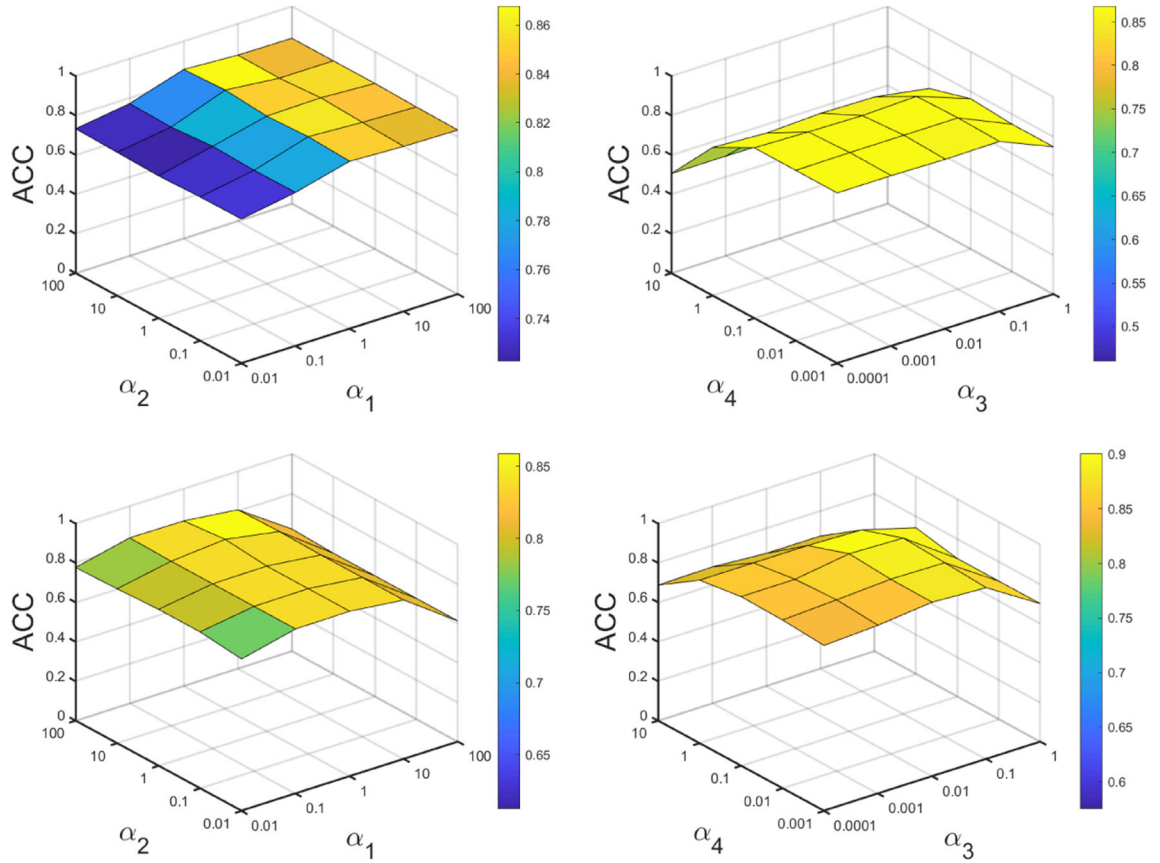

Fig. 3 Parameter selection with respect to α_1 and α_2 (the first column), α_3 and α_4 (the last column) on ORL (the first row) and Umist (the last row)

Table 6 Clustering results of our method and compared methods on all datasets for ablation study

Dataset	Metric	ours(L_1)	ours($L_1 + L_2$)	ours($L_1 + L_3$)	ours($L_1 + L_2 + L_3$)
MNIST	ACC	0.7650	0.7670	0.7650	**0.7770**
	NMI	0.6759	0.6773	**0.6924**	0.6901
	PUR	0.7650	0.7950	0.7660	**0.8040**
EYaleB	ACC	0.9782	0.9782	0.9782	**0.9786**
	NMI	0.9697	0.9694	0.9692	**0.9711**
	PUR	0.9786	**0.9786**	**0.9786**	**0.9786**
ORL	ACC	0.8700	**0.8750**	0.8725	**0.8750**
	NMI	0.9262	0.9317	0.9286	**0.9320**
	PUR	0.8875	**0.8900**	**0.8900**	**0.8900**
COIL20	ACC	0.9722	0.9743	0.9736	**0.9778**
	NMI	0.9719	0.9737	0.9734	**0.9758**
	PUR	0.9722	0.9743	0.9736	**0.9778**
Umist	ACC	0.8541	0.9041	0.8937	**0.9333**
	NMI	0.9127	0.9340	0.9372	**0.9463**
	PUR	0.8645	0.9041	0.9146	**0.9333**

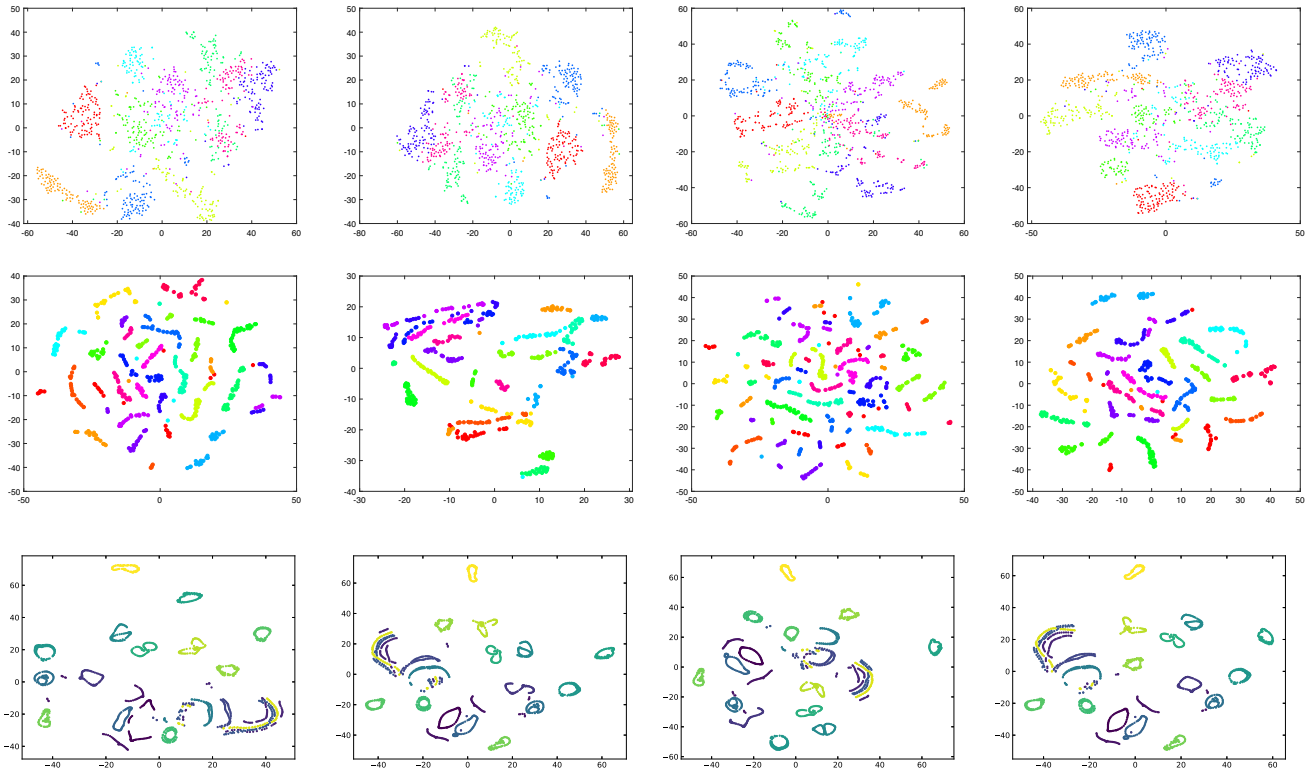

Fig. 4 2D visualization of the embedding results on MNIST (the first row), Umist (the second row) and COIL20 (the last row) datasets. The columns from left to right denote DSC, RGRL, DLRSC, and our LDLRSC, respectively

LDLRSC is insensitive to parameter α_2 as shown in lower right subfigure of Fig. 3. On Umist dataset, the proposed LDLRSC could achieve higher ACC values without tuning parameter α_1 and α_2 against all competitors as shown in top right subfigure of Fig. 3.

4.6 Computation complexity

It is intractable to analyze the theoretical complexity of network, since the time complexity of the neural network (here an autoencoder) is related to the number of the batch, the size and the dimensionality of the dataset, and computation of the error in back propagation process which depends on specified optimizer. The practical complexity time may differ in different computation frameworks for deep learning. However, it is clear that the computation cost of the proposed Algorithm 2 is dominated by computing the loss function (15). Specifically, let m denote the original latent dimension, d denote the latent dimension of a sample after the transformation of the encoder and N denote the size of the dataset. For the forward propagation, the Laplacian constraint costs $O(N^2 \times d^2)$, the $\|C\|_\gamma$ costs $O(N^3)$ and the other terms, i.e. reconstruction loss and self-expression loss cost $O(N \times m)$ and $O(N \times d)$ respectively. For the back propagation, it is intractable to analyze the theoretical complexity. But in our experiment, the time consumption of the experiment will be significantly reduced without γ-norm constraint, so we speculate that the singular value decomposition has high complexity in the back propagation in the practical experiment.

4.7 Ablation study

To show the contributions of low-rank prior and two types of Laplacian constrints, we also conduct several experiments on the variants of our LDLRSC method, (using loss $\mathcal{L}_1$, using $\mathcal{L}_1 + \mathcal{L}_2$, using loss $\mathcal{L}_1 + \mathcal{L}_3$ and using loss $\mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$). Here, losses $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ are defined in Eq. (16). The experimental results for the various possible loss combinations are shown in the Table 6. We observe that (1) Even under the condition without two Laplacian constaints, i.e., using L_1, the performance of our LDLRSC method on all datasets is also better than DSC [26], directly indicating the superiority of low-rank prior. (2) These two types of Laplacian constraints have similar performance on all datasets while combining Laplacian constraints and low-rank prior can further improve the clustering performance.

4.8 Visualization

We also show the visualization of the learned affinity matrix by the t-SNE algorithm in Fig. 4 to further investigate the clustering performance. Here, we use MNIST, Umist and COIL20 as the testing datasets and compare our method with three popular deep subspace clustering methods DSC, RGRL, and DLRSC.

As observed, our LDLRSC method attempt to cluster all samples by inheriting from the superiority of low-rank prior and local geometric structures, such that sample belonging to the same cluster are usually clustering together and easily separated.

5 Conclusion

In this paper, we developed a novel deep subspace clustering method named Laplacian regularized Deep Low-Rank Subspace Clustering Network (LDLRSC) to explore the global low-rank prior and local non-linear geometric structures simultaneously by the nonconvex γ-norm and Laplacian constraint, respectively. In our network architecture, LDLRSC transforms the original data into a latent low-dimensional representation by an encoder, passes through a fully connected layer that simulates the nature of self-expression, and finally reconstructs features by the decoder. To the best of our knowledge, this is the first attempt to directly use neural networks to learn low-rank representation. In addition, we try to exploit two kinds of Laplacian constraints in our proposed loss function, which can mine the geometric structure of the data. Extensive experiments show the effectiveness of our LDLRSC method compared to several shallow subspace clustering methods and deep subspace clustering methods. In the future, we aim to extend the proposed method to handle the multi-view clustering to explore the consistent and supplementary information embedding multiple features.

Funding This work was supported in part by the National Natural Science Foundation of China under Grants 62106063, 62071142, 62002301, by the Guangdong Natural Science Foundation under Grant 2022A1515010819, by the Shenzhen College Stability Support Plan under Grants GXWD20201230155427003-20200824113231001 and GXWD20201230155427003-20200824210638001, by the Shenzhen Science and Technology Program under Grant RCBS20210609103708013, by the University Innovation Research Group project of Chongqing (Evidence Science Technology Innovation and Application) under Grant CXQT21033, and by Humanities and Social Sciences Foundation of the Ministry of Education of China (22YJC630129).

Data Availability The datasets we used in this paper are public without private protection.

References

1. Vidal R (2011) Subspace clustering. IEEE Signal Process Mag 28(2):52–68
2. Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsl 6(1):90–105
3. Chen G, Lerman G (2009) Spectral curvature clustering (SCC). Int J Comput Vision 81(3):317–330

4. Soltanolkotabi M, Candes EJ (2012) A geometric analysis of subspace clustering with outliers. Ann Stat 40(4):2195–2238
5. Elhamifar E, Vidal R (2013) Sparse subspace clustering: algorithm, theory, and applications. IEEE Trans Pattern Anal Mach Intell 35(11):2765–2781
6. Liu G, Lin Z, Yan S, Sun J, Yu Y, Ma Y (2012) Robust recovery of subspace structures by low-rank representation. IEEE Trans Pattern Anal Mach Intell 35(1):171–184
7. Ntoutsi E, Stefanidis K, Rausch K, Kriegel H-P (2014) "Strength lies in differences" diversifying friends for recommendations through subspace clustering. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp 729–738
8. Lin J, Huang T-Z, Zhao X-L, Jiang T-X, Zhuang L (2020) A tensor subspace representation-based method for hyperspectral image denoising. IEEE Trans Geosci Remote Sens 59(9):7739–7757
9. Zhai H, Zhang H, Zhang L, Li P (2018) Laplacian-regularized low-rank subspace clustering for hyperspectral image band selection. IEEE Trans Geosci Remote Sens 57(3):1723–1740
10. Chen Y, Xiao X, Peng C, Lu G, Zhou Y (2021) Low-rank tensor graph learning for multi-view subspace clustering. IEEE Trans Circuits Syst Video Technol 32(1):92–104
11. Wang S, Chen Y, Cen Y, Zhang L, Wang H, Voronin V (2022) Nonconvex low-rank and sparse tensor representation for multi-view subspace clustering. Appl Intell 1–14
12. Chen H, Tai X, Wang W (2022) Multi-view subspace clustering with inter-cluster consistency and intra-cluster diversity among views. Appl Intell 1–17
13. Ke G, Hong Z, Yu W, Zhang X, Liu Z (2022) Efficient multi-view clustering networks. Appl Intell 1–17
14. Xu J, Yu M, Shao L, Zuo W, Meng D, Zhang L, Zhang D (2021) Scaled simplex representation for subspace clustering. IEEE Trans Cybern 51(3):1493–1505
15. Zuo W, Meng D, Zhang L, Feng X, Zhang D (2013) A generalized iterated shrinkage algorithm for non-convex sparse coding. In: Proceedings of the IEEE International Conference on Computer Vision. pp 217–224
16. 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: European Conference on Computer Vision. pp 347–360
17. Zhang H, Yang J, Shang F, Gong C, Zhang Z (2018) LRR for subspace segmentation via tractable Schatten-p norm minimization and factorization. IEEE Trans Cybern 49(5):1722–1734
18. Peng X, Yi Z, Tang H (2015) Robust subspace clustering via thresholding ridge regression. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp 3827–3833
19. Lu C, Feng J, Lin Z, Mei T, Yan S (2018) Subspace clustering by block diagonal representation. IEEE Trans Pattern Anal Mach Intell 41(2):487–501
20. Vidal R, Favaro P (2014) Low rank subspace clustering (LRSC). Pattern Recogn Lett 43:47–61
21. Patel VM, Vidal R (2014) Kernel sparse subspace clustering. In: 2014 IEEE International Conference on Image Processing. pp 2849–2853
22. Xiao S, Tan M, Xu D, Dong ZY (2015) Robust kernel low-rank representation. IEEE Trans Neural Netw Learn Syst 27(11):2268–2281
23. Abhadiomhen SE, Wang Z, Shen X (2021) Coupled low rank representation and subspace clustering. Appl Intell 1–17
24. Zhen L, Peng D, Wang W, Yao X (2020) Kernel truncated regression representation for robust subspace clustering. Inf Sci 524:59–76
25. Peng X, Xiao S, Feng J, Yau W-Y, Yi Z (2016) Deep subspace clustering with sparsity prior. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. pp. 1925–1931
26. Ji P, Zhang T, Li H, Salzmann M, Reid I (2017) Deep subspace clustering networks. Adv Neural Inf Proces Syst 24–33
27. Zhou P, Hou Y, Feng J (2018) Deep adversarial subspace clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 1596–1604
28. Zhang T, Ji P, Harandi M, Huang W, Li, H (2019) Neural collaborative subspace clustering. In: International Conference on Machine Learning. pp 7384–7393
29. Kheirandishfard M, Zohrizadeh F, Kamangar F (2020) Multi-level representation learning for deep subspace clustering. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp 2039–2048
30. Abavisani M, Patel VM (2018) Deep multimodal subspace clustering networks. IEEE J Sel Top Sign Proces 12(6):1601–1614
31. Chen Y, Guo Y, Wang Y, Wang D, Peng C, He G (2017) Denoising of hyperspectral images using nonconvex low rank matrix approximation. IEEE Trans Geosci Remote Sens 55(9):5366–5380
32. Ke Z, Huang W, Cui Z-X, Cheng J, Jia S, Wang H, Liu X, Zheng H, Ying L, Zhu Y et al (2021) Learned low-rank priors in dynamic MR imaging. IEEE Trans Med Imaging 1–1
33. Liu X, Zhao G, Yao J, Qi C (2015) Background subtraction based on low-rank and structured sparse decomposition. IEEE Trans Image Process 24(8):2502–2514
34. Liu Y, Yuan X, Suo J, Brady DJ, Dai Q (2018) Rank minimization for snapshot compressive imaging. IEEE Trans Pattern Anal Mach Intell 41(12):2990–3006
35. Chen Y, Wang S, Peng C, Hua Z, Zhou Y (2021) Generalized nonconvex low-rank tensor approximation for multi-view subspace clustering. IEEE Trans Image Process 30:4022–4035
36. Kheirandishfard M, Zohrizadeh F, Kamangar F (2020) Deep low-rank subspace clustering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp 864–865
37. Yin M, Gao J, Lin Z (2015) Laplacian regularized low-rank representation and its applications. IEEE Trans Pattern Anal Mach Intell 38(3):504–517
38. Deng T, Ye D, Ma R, Fujita H, Xiong L (2020) Low-rank local tangent space embedding for subspace clustering. Inf Sci 508:1–21
39. Li C-G, You C, Vidal R (2017) Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework. IEEE Trans Image Process 26(6):2988–3001
40. Zhang J, Li C-G, You C, Qi X, Zhang H, Guo J, Lin Z (2019) Self-supervised convolutional subspace clustering network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 5473–5482
41. Kang Z, Lu X, Liang J, Bai K, Xu Z (2020) Relation-guided representation learning. Neural Netw 131:93–102
42. Kanatani KI (2001) Motion segmentation by subspace separation and model selection, vol 2. In: Proceedings Eighth IEEE International Conference on Computer Vision. pp 586–591
43. Ho J, Yang M-H, Lim J, Lee K-C, Kriegman, D (2003) Clustering appearances of objects under varying illumination conditions, vol 1. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. p 2003
44. Rao SR, Tron R, Vidal R, Ma Y (2008) Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: IEEE Conference on Computer Vision and Pattern Recognition. pp 1–8
45. Ji P, Salzmann M, Li H (2014) Efficient dense subspace clustering. In: IEEE Winter Conference on Applications of Computer Vision. pp 461–468
46. Liu G, Lin Z, Yu Y (2010) Robust subspace segmentation by low-rank representation. In: Proceedings of the 27th International Conference on Machine Learning. pp 663–670

47. Yang X, Jiang X, Tian C, Wang P, Zhou F, Fujita H (2020) Inverse projection group sparse representation for tumor classification: a low rank variation dictionary approach. Knowl-Based Syst 196:105768
48. Fazel M (2002) Matrix rank minimization with applications. PhD thesis, PhD thesis, Stanford University
49. Patel VM, Van Nguyen H, Vidal R (2013) Latent space sparse subspace clustering. In: Proceedings of the IEEE International Conference on Computer Vision. pp 225–232
50. Huang P, Huang Y, Wang W, Wang L (2014) Deep embedding network for clustering. In: 2014 22nd International Conference on Pattern Recognition. pp 1532–1537
51. Xie J, Girshick R, Farhadi A (2016) Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning. pp 478–487
52. Belkin M, Niyogi P (2001) Laplacian eigenmaps and spectral techniques for embedding and clustering, vol 14. In: Advances in Neural Information Processing Systems. pp 585–591
53. Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems. pp 849–856
54. Munkres J (1957) Algorithms for the assignment and transportation problems. J Soc Ind Appl Math 5(1):32–38
55. Schütze H, Manning CD, Raghavan P (2008) Introduction to Information Retrieval. Cambridge University Press Cambridge, ???
56. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. Preprint at http://arxiv.org/abs/1412.6980
57. Samaria FS, Harter AC (1994) Parameterisation of a stochastic model for human face identification. In: Proceedings of 1994 IEEE Workshop on Applications of Computer Vision. pp 138–142
58. You C, Li C-G, Robinson DP, Vidal R (2016) Oracle based active set algorithm for scalable elastic net subspace clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 3928–3937
59. You C, Robinson D, Vidal R (2016) Scalable sparse subspace clustering by orthogonal matching pursuit. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 3918–3927
60. Majumdar A (2018) Graph structured autoencoder. Neural Netw 106:271–280