Deep Subspace Clustering via Latent Distribution Preserving

Lei Zhou[1] , **Xiao Bai**[1] , **Dong Wang**[1] , **Xianglong Liu**[2,1*] , **Jun Zhou**[3] , **Edwin Hancock**[4,1]
[1]School of Computer Science and Engineering, Beijing Advanced Innovation Center for Big Data and Brain Computing, Qingdao Research Institute, Beihang University, Beijing, China
[2]State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
[3]School of Information and Communication Technology, Griffith University, Nathan, Australia
[4]Department of Computer Science, University of York, York, U.K.
{leizhou, baixiao, dongwang}@buaa.edu.cn, xlliu@nlsde.buaa.edu.cn, jun.zhou@griffith.edu.au, edwin.hancock@york.ac.uk

Abstract

Subspace clustering is a useful technique for many computer vision applications in which the intrinsic dimension of high-dimensional data is smaller than the ambient dimension. Traditional subspace clustering methods often rely on the self-expressiveness property, which has proven effective for linear subspace clustering. However, they perform unsatisfactorily on real data with complex nonlinear subspaces. More recently, deep autoencoder based subspace clustering methods have achieved success owning to the more powerful representation extracted by the autoencoder network. Unfortunately, these methods only considering the reconstruction of original input data can hardly guarantee the latent representation for the data distributed in subspaces, which inevitably limits the performance in practice. In this paper, we propose a novel deep subspace clustering method based on a latent distribution-preserving autoencoder, which introduces a distribution consistency loss to guide the learning of distribution-preserving latent representation, and consequently enables strong capacity of characterizing the real-world data for subspace clustering. Experimental results on several public databases show that our method achieves significant improvement compared with the state-of-the-art subspace clustering methods.

1 Introduction

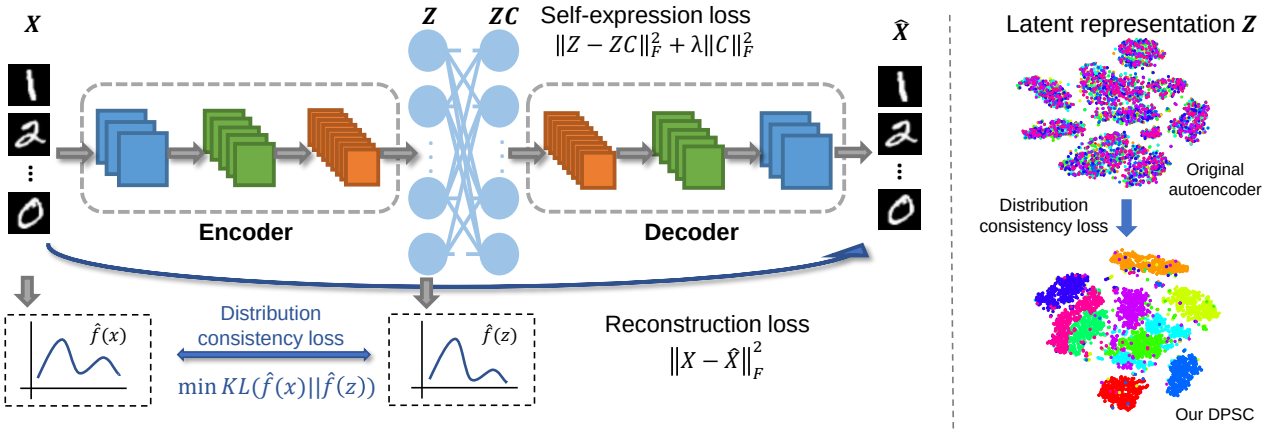
In many computer vision applications, such as face recognition [Liu *et al.*, 2013; Zhou *et al.*, 2018a], texture recognition [Lu *et al.*, 2014; Wang *et al.*, 2018] and motion segmentation [Elhamifar and Vidal, 2013], visual data can be well characterized by subspaces. Moreover, the intrinsic dimension of high-dimensional data is often much smaller than the ambient dimension [Vidal, 2011]. This has motivated the development of subspace clustering technique which simultaneously cluster the data into multiple subspaces and find a low-dimensional subspace for each class of data.

*Corresponding Author

Many subspace clustering algorithms have been developed during the past decade, including algebraic [Vidal *et al.*, 2005], iterative [Agarwal and Mustafa, 2004], statistical [Rao *et al.*, 2008], and spectral clustering methods [Elhamifar and Vidal, 2013; Liu *et al.*, 2013; Lu *et al.*, 2012; Patel *et al.*, 2013; Lu *et al.*, 2014; Peng *et al.*, 2016a; Li *et al.*, 2017b; Zhou *et al.*, 2018b]. Among these approaches, spectral clustering methods have been intensively studied thanks to their theoretical soundness and empirical success. These methods are based on the self-expressiveness property of data lying in a union of linear subspaces, which states that each point in a union of subspaces can be written as a linear combination of other data points in the subspaces. Two typical methods are sparse subspace clustering (SSC) [Elhamifar and Vidal, 2013] and low-rank representation (LRR) [Liu *et al.*, 2013]. SSC uses the ℓ_1 norm to enforce the sparsity of self-representation coefficient matrix. LRR uses nuclear norm minimization to make coefficient matrix low-rank.

Recently, deep neural network based subspace clustering methods [Peng *et al.*, 2016b; Peng *et al.*, 2017; Li *et al.*, 2017a; Ji *et al.*, 2017; Peng *et al.*, 2018; Zhou *et al.*, 2018c] have been proposed to learn better sample representations for subspace clustering with complex structures rather than the linear ones. However, like these conventional shallow methods [Elhamifar and Vidal, 2013; Liu *et al.*, 2013; Favaro *et al.*, 2011; Lu *et al.*, 2012], they still hinge on self-expression as supervision, which may not perform well on data with inconvenient distributions. This is because both these shallow and deep subspace clustering methods only reveal the intrinsic Euclidean structure of data, and do not consider the intrinsic cluster structure which is often the union of some non-linear subspaces.

In this paper, in light of the above arguments, we propose a novel deep subspace clustering method based on a latent distribution-preserving autoencoder, namely, **D**istribution-**P**reserving **S**ubspace **C**lustering (DPSC). Motivated by the fact that the data points are drawn from the union of some low-dimensional subspaces embedded in a high-dimensional ambient space, and the subspace structure of each cluster can be described by the distribution of the cluster elements. The key idea of DPSC is to preserve the intrinsic cluster structure of data space by minimizing the inconsistency between the



(a) Architecture of DPSC (b) Visualization result

Figure 1: (a) Architecture of DPSC. The top network is an autoencoder with one self-expressive layer. The bottom diagram shows a distribution consistency loss which is measured by the KL divergence between the probability density distribution of latent representations and the original images. (b) Visualization result of MNIST with 5,000 data points on the latent space learned by pre-trained autoencoder and DPSC.

original data distribution and the latent representation distribution. We first utilize a nonparametric technique, i.e. Kernel Density Estimation (KDE) [Hinneburg and Gabriel, 2007], to estimate the distribution of the data. Then we design a distribution consistency loss by minimizing the Kullback-Leibler (KL) divergence between the original data distribution and the learned latent representation distribution. The architecture of DPSC is shown in Fig. 1(a). It consists of a subspace clustering autoencoder network and a distribution consistency loss term that learns to supervise the distribution of latent representation. As the visualization result shown in Fig. 1(b), our method can better preserve the cluster structure of the original data space by introducing the distribution consistency loss which leads to a better clustering result.

The main contributions of this paper are as follows:

1. We propose a novel distribution-preserving subspace clustering method. By developing a latent distribution-preserving autoencoder, DPSC can preserve the intrinsic cluster structure of data space and supervise the encoder to produce more favorable representation for subspace clustering.
2. We design a simple but effective distribution consistency loss by minimizing the KL divergence between the original data distribution and the learned latent representation distribution, which is complementary to the autoencoder induced self-expression loss.
3. Experimental results on several public databases with different subspace applications show that our method leads to significant improvement compared with the state-of-the-arts on both linear and non-linear subspace clustering problems.

2 Related Works

In this section, we review some works related to the proposed subspace clustering method.

Traditional Subspace Clustering. Given a data matrix $\mathbf{X}$ that contains N data points drawn from k subspaces $\{S_i\}_{i=1}^k$, SSC [Elhamifar and Vidal, 2013] aims to find a sparse representation matrix $\mathbf{C}$ showing mutual similarity of the points, i.e., $\mathbf{X} = \mathbf{XC}$. Since each point in S_i can be expressed in terms of the other points in S_i, such a sparse representation matrix $\mathbf{C}$ always exists. As pointed out in LRR [Liu *et al.*, 2013], SSC finds the sparsest representation of each data vector individually. There is no global constraint on its solution, so SSC method may be inaccurate when it is used to capture the global structures of data. [Liu *et al.*, 2013] proposed that low rankness can be a more appropriate criterion.

SSC and LRR methods solve the robust subspace clustering problem by removing the outliers from the original data space and obtaining a good affinity matrix based on a clean database. Thus they need prior knowledge on the structures of the errors, which usually is unknown in practice. [Peng *et al.*, 2015] proposed a robust subspace clustering method which overcomes this limitation by eliminating the effect of errors from the projection space with a model of thresholding ridge regression (TRR).

Deep Subspace Clustering. Deep subspace clustering methods have been proposed to solve the linear subspace assumption. [Peng *et al.*, 2017] simultaneously learned a compact representation using a neural network and a clustering assignment by minimizing the discrepancy between pair-wise sample-centers distributions. [Ji *et al.*, 2017] proposed a deep autoencoder framework for subspace clustering by developing a self-expressive layer to mimic the "self-expressiveness" property of a union of subspaces. [Peng *et al.*, 2018] proposed a structured autoencoder which learns a set of transformations to map input data points into nonlinear latent spaces. Most recently, [Zhou *et al.*, 2018c] adopted a GAN-alike model to supervise sample representation learning for subspace clustering. Different from these methods, the proposed DPSC introduces distribution consistency loss between the original data distribution and the latent representation distribution to preserve the cluster structure.

3 Distribution-Preserving Deep Subspace Clustering

We propose a distribution-preserving subspace clustering (DPSC) method, which can learn more favorable data representation for subspace clustering via a latent distribution-preserving autoencoder. DPSC clusters data in three steps: 1) Initializing the parameters of autoencoder with data reconstruction and latent representations self-expression; 2) Estimating the distributions of original data and learned latent representation; 3) Updating the network parameters by adding the distribution consistency constraint, and clustering the latent data representations into multiple subspaces.

3.1 Subspace Clustering with Self-Expression Loss

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ denote the input data points and $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$ denote their corresponding latent representation learned by the encoder. The parameters of encoder and decoder are denoted by Θ_e and Θ_d respectively. The number of clusters is denoted by k.

DPSC first learns the latent representations for the input data through a traditional autoencoder with the reconstruction loss:

$$\mathcal{L}_r(\Theta_e; \Theta_d) = \left\| \mathbf{X} - \widehat{\mathbf{X}} \right\|_F^2 \tag{1}$$

where $\widehat{\mathbf{X}} = \Theta_d(\Theta_e(\mathbf{X}))$ denotes the reconstruction of $\mathbf{X}$ by the autoencoder. Then the latent representations $\mathbf{Z} = \Theta_e(\mathbf{X})$. As done in [Ji *et al.*, 2017], we add a self-expressive layer in the autoencoder to learn better latent representations for subspace clustering. The self-expression loss is defined as:

$$\mathcal{L}_s(\mathbf{C}) = \|\mathbf{Z} - \mathbf{Z}\mathbf{C}\|_F^2 + \lambda \|\mathbf{C}\|_F^2 \tag{2}$$

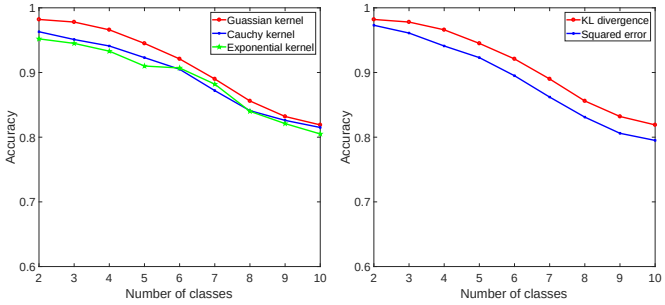
where $\mathbf{C} \in \mathbb{R}^{N \times N}$ is the self-expression coefficient matrix. We adopt an F-norm penalty on $\mathbf{C}$, since compared with the non-smooth penalty term, e.g. the ℓ_1 norm, it can be learned more easily while also achieving comparable or even better performance. With the learned latent representations $\mathbf{Z}$, we can use a spectral clustering algorithm on the affinity matrix $\mathbf{W} = \frac{1}{2}(|\mathbf{C}| + |\mathbf{C}^T|)$ to obtain the subspace clustering result $\mathbf{c}_i (i = 1, \ldots, k)$.

3.2 Distribution-Preserving with Distribution Consistency Loss

To address inconvenient sample distributions, we then design a distribution consistency loss as a complementary unsupervised solution to the self-expression loss.

The proposed DPSC clusters data by simultaneously learning a set of k cluster centers $\{\mathbf{c}_i\}_{i=1}^k$ in the latent feature space $\mathbf{Z}$ and the parameters Θ_e of the encoder that map the data points into $\mathbf{Z}$. Given an initial estimate of the non-linear mapping Θ_e, Θ_d and the initial cluster centroids $\{\mathbf{c}_i\}_{i=1}^k$, we propose to improve the clustering in an unsupervised manner. We first estimate the probability density distribution of the original data space and the latent representational space initialized by the autoencoder. Then a distribution consistency loss between these two distributions is learned to guide the update of the distribution-preserving autoencoder network.

Kernel Density Estimation. Minimizing the inconsistency of distributions of data between the original space and the latent representational space is a challenging task because in practice there is little prior information to help us to identify the distribution of the data. To solve this problem, we propose to use kernel density estimation, which is a nonparametric technique ,to estimate the real distribution of the data.



(a) (b)

Figure 2: Clustering accuracy of the proposed method on the MNIST database: (a) different kernel functions; (b) different loss functions.

The kernel density estimation of a given database $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ is:

$$\begin{aligned} \hat{f}(\mathbf{x}) &= \frac{1}{N} \sum_{i=1}^{N} \kappa_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i) \\ &= \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{H}\|^{-1/2} \kappa(\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{x}_i)) \end{aligned} \tag{3}$$

where $\mathbf{H} = h^2\mathbf{I}$ is a matrix of smoothness and specifies the width of the kernel around each sample point $\mathbf{x}_i$, h is the bandwidth of the neighbourhood. $\kappa(\mathbf{x})$ is the kernel function. The experimental result in Fig. 2(a) shows that the Guassian kernel is better than both the Cauchy kernel and the exponential kernel. Thus the Gaussian kernel is chosen in our method. Then the kernel density estimation of the database $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ is:

$$\begin{aligned} \hat{f}(\mathbf{x}) &= \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{H}\|^{-1/2} \kappa(\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{x}_i)) \\ &= \frac{1}{Nh} \sum_{i=1}^{N} \kappa\left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{h} \right\| \right) \end{aligned} \tag{4}$$

Distribution Consistency Loss. Our aim is to find the mapping function to preserve the distribution of the original database so that $\hat{f}(\mathbf{x}_i) = \hat{f}(\Theta_e(\mathbf{x}_i))$ for each data point in $\mathbf{X}$, where $\Theta_e(\mathbf{x}_i) = \mathbf{z}_i$. This can be achieved by minimizing the KL divergence based distribution consistency loss:

$$\mathcal{L}_d(\Theta_e) = \sum_{i=1}^{N} \hat{f}(\mathbf{x}_i) \log \frac{\hat{f}(\mathbf{x}_i)}{\hat{f}(\mathbf{z}_i)} \tag{5}$$

which is a well-known criterion for describing the dissimilarity between two distributions. Here, $\hat{f}(\mathbf{x}_i)$ and $\hat{f}(\mathbf{z}_i)$ are the

(a) MNIST Database (b) CMU PIE Database (c) COIL-20 and COIL-100

Figure 3: Sample images from the MNIST database, CMU PIE database, COIL-20 and COIL-100 Database.

probability density distributions of $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$, respectively. Fig. 2(b) shows the superiority of KL divergence compared with squared error. Clearly, our objective function is proposed to achieve invariance of distribution by minimizing the discrepancy between the target distribution $\hat{f}(\mathbf{x})$ and the predicted distribution $\hat{f}(\mathbf{z})$.

3.3 The Formulation and Training Strategy

Finally, we obtain the loss function for the DPSC by combining the data reconstruction loss, the self-expression loss and the distribution consistency loss:

$$\mathcal{L}(\Theta_e; \mathbf{C}; \Theta_d) = \mathcal{L}_r + \lambda_1 \mathcal{L}_s + \lambda_2 \mathcal{L}_d \tag{6}$$

The first term of the loss function denotes the reconstruction loss of the autoencoder in Eq. (1). The second term corresponds to self-expressive loss in Eq. (2). The last term is the distribution consistency loss as shown in Eq. (5) which preserves the cluster structure between original data space and latent representational space.

Due to the large amount of parameters, it is intractable to directly train the network from scratch. To address this, we design a two-stages training strategy. First we pre-train the deep autoencoder without considering the distribution consistency loss term, i.e., minimizing the reconstruction loss and self-expression loss in Eq. (6) while discarding the last term. In this way, the deep autoencoder can produce a reasonable good initial representation. After that, we estimate the probability density distributions of the original data space and the latent representational space initialized by the deep autoencoder. Then a distribution consistency loss between these two distributions is learned to guide the update of the autoencoder network. We train the whole DPSC network by minimizing the loss function (6) with the Adam algorithm [Kingma and Ba, 2015]. The learning rate is set as 1×10^{-3} for all experiments.

After the whole network is trained, we can use the parameters of the self-expressive layer, i.e. $\mathbf{C}$, to construct an affinity matrix for spectral clustering. During testing, we perform spectral clustering on the constructed affinity matrix $\mathbf{W} = \frac{1}{2}(|\mathbf{C}| + |\mathbf{C}^T|)$. For fairness of comparing with other methods, we use the NCut algorithm as in [Elhamifar and Vidal, 2013; Ji *et al.*, 2017].

4 Experiments

We conducted experiments on three subspace clustering tasks: a) handwritten digit clustering, b) face recognition, and c) object clustering. The first two tasks are relatively easier since handwriting digit and face images approximately lie on a union of linear subspaces. The last task is more challenging with the non-linear subspace representation.

The baseline subspace clustering methods include sparse subspace clustering (SSC) [Elhamifar and Vidal, 2013], kernel SSC (KSSC) [Patel and Vidal, 2014], elastic net subspace clustering (ENSC) [You *et al.*, 2016], efficient dense subspace clustering (EDSC) [Ji *et al.*, 2014], low-rank representation (LRR) [Liu *et al.*, 2013], low-rank subspace clustering (LRSC) [Favaro *et al.*, 2011], deep subspace clustering network (DSC-Net) [Ji *et al.*, 2017], structured autoencoder (StructAE) [Peng *et al.*, 2018], deep adversarial subspace clustering (DASC) [Zhou *et al.*, 2018c] and SSC with the pre-trained convolutional autoencoder features (AE+SSC). Among these methods, AE+SSC only uses the features from pre-trained autoencoder without self-expression layer and distribution consistency loss. DSC-Net adds self-expression layer in autoencoder but has no distribution consistency loss. These two methods can better show the superiority of our latent distribution-preserving autoencoder with distribution consistency loss.

We adopted the following widely used clustering metrics to measure the clustering performance: accuracy (ACC), normalized mutual information (NMI) and the purity (PUR).

4.1 Handwritten Digit Clustering

We test the proposed method on handwritten digit clustering using the MNIST database [Lecun *et al.*, 1998]. This database contains 10 clusters, including handwritten digits 0-9 as shown in Fig. 3(a). Each cluster contains 6,000 images for training and 1,000 images for testing, with a size of 28×28 pixels in each image. We randomly selected 1,000 images from each digit for our experiment. We fixed the number of clusters $k = 10$ and chose different numbers of data points for each cluster. Each cluster contained N_i data points randomly chosen from the corresponding 1,000 images, where $N_i \in \{100, 500, 1000\}$, so that the number of total points $N \in \{1000, 5000, 10000\}$. For DPSC, we set the bandwidth $h = 2$, $\lambda = \lambda_1 = 1$ and $\lambda_2 = 2$. Then we applied all methods on this database for comparison.

Table 1: Clustering results on the MNIST database with different numbers of data points.

No. Points	1000			5000			10000		
Metric	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
SSC [Elhamifar and Vidal, 2013]	0.4530	0.4709	0.4940	0.4214	0.4325	0.4510	0.3852	0.3982	0.4045
KSSC [Patel and Vidal, 2014]	0.5220	0.5623	0.5810	0.4715	0.4956	0.5120	0.4421	0.4736	0.4950
ENSC [You *et al.*, 2016]	0.4983	0.5495	0.5483	0.4425	0.4750	0.4802	0.4100	0.4355	0.4382
EDSC [Ji *et al.*, 2014]	0.5650	0.5752	0.6120	0.5312	0.5520	0.5732	0.4910	0.5070	0.5104
LRR [Liu *et al.*, 2013]	0.5386	0.5632	0.5684	0.5122	0.5410	0.5468	0.4871	0.5015	0.5038
LRSC [Favaro *et al.*, 2011]	0.5140	0.5576	0.5550	0.4825	0.5105	0.5082	0.4521	0.4835	0.4802
AE+SSC	0.4840	0.5337	0.5290	0.4512	0.4920	0.4827	0.4250	0.4628	0.4572
DSC-Net-L1 [Ji *et al.*, 2017]	0.7280	0.7217	0.7890	0.7105	0.7067	0.7554	0.6985	0.6921	0.7482
DSC-Net-L2 [Ji *et al.*, 2017]	0.7500	0.7319	0.7991	0.7358	0.7214	0.7746	0.7167	0.7002	0.7621
StructAE [Peng *et al.*, 2018]	0.7832	0.7610	0.8125	0.7615	0.7412	0.7920	0.7438	0.7314	0.7750
DASC [Zhou *et al.*, 2018c]	0.8040	0.7800	0.8370	0.7885	0.7622	0.8145	0.7725	0.7604	0.8120
DPSC	**0.8252**	**0.8014**	**0.8571**	**0.8172**	**0.7924**	**0.8359**	**0.8057**	**0.7912**	**0.8432**

Table 2: Clustering results on the CMU PIE database with different numbers of objects.

No. Objects	5			10			20		
Metric	ACC	NMI	PUR	ACC	NMI	PUR	ACC	NMI	PUR
SSC [Elhamifar and Vidal, 2013]	0.8912	0.9005	0.8924	0.8623	0.8745	0.8662	0.8254	0.8342	0.8276
KSSC [Patel and Vidal, 2014]	0.8041	0.8325	0.7150	0.7822	0.8175	0.7936	0.7563	0.7955	0.7702
ENSC [You *et al.*, 2016]	0.8875	0.9014	0.8927	0.8652	0.8873	0.8720	0.8324	0.8652	0.8465
EDSC [Ji *et al.*, 2014]	0.8337	0.8855	0.8520	0.8150	0.8642	0.8315	0.7920	0.8405	0.8127
LRR [Liu *et al.*, 2013]	0.8950	0.9124	0.8980	0.8735	0.8920	0.8804	0.8468	0.8721	0.8536
LRSC [Favaro *et al.*, 2011]	0.8125	0.8642	0.8355	0.8010	0.8469	0.8152	0.7802	0.8382	0.7968
AE+SSC	0.8924	0.9155	0.9075	0.8724	0.9053	0.8865	0.8505	0.8927	0.8714
DSC-Net-L1 [Ji *et al.*, 2017]	0.9315	0.9426	0.9384	0.9204	0.9352	0.9287	0.9050	0.9168	0.9076
DSC-Net-L2 [Ji *et al.*, 2017]	0.9408	0.9542	0.9465	0.9245	0.9405	0.9327	0.9087	0.9304	0.9185
StructAE [Peng *et al.*, 2018]	0.9215	0.9243	0.9250	0.9027	0.9062	0.9155	0.8905	0.9048	0.9075
DASC [Zhou *et al.*, 2018c]	0.9455	0.9612	0.9637	0.9332	0.9534	0.9487	0.9168	0.9365	0.9250
DPSC	**0.9670**	**0.9824**	**0.9805**	**0.9568**	**0.9750**	**0.9723**	**0.9382**	**0.9620**	**0.9517**

Table 1 shows the clustering results on MNIST. Here DPSC outperforms the baselines in all three metrics given different numbers of data points. Specifically, when each cluster contains 1,000 data points, our method outperforms the second best method DASC by 3.32%, 3.08% and 3.12% in terms of ACC, NMI and PUR, respectively. Moreover, DPSC achieves a significant improvement over the shallow subspace clustering methods, e.g., SSC and LRR. This is because compared with shallow methods, DPSC uses a multi-layer convolutional autoencoder as the feature extractor. Therefore, DPSC can better handle translation, rotation and shift in the handwritten images while mapping the input data into a union of linear subspaces.

4.2 Face Recognition

Since subspaces are commonly used to capture the appearance of faces under varying illuminations, we also test the performance of our method on face clustering with the CMU PIE database [Sim *et al.*, 2001]. The CMU PIE database contains 41,368 images of 68 people under 13 different poses, 43 different illumination conditions, and 4 different expressions as shown in Fig. 3(b). In our experiment, we used the face images in five near frontal poses (P05, P07, P09, P27, P29). Then each people had 170 face images under different illuminations and expressions. Each image was manually cropped and normalized to a size of 32×32 pixels. We randomly picked $k \in \{5, 10, 20\}$ individuals to investigate the performance of all methods. For our method, we set the bandwidth $h = 2$, $\lambda = \lambda_1 = 1$ and $\lambda_2 = 2$.

Table 2 reports the clustering results on the CMU PIE face database. It can be observed that DPSC consistently outperforms the baselines on all three metrics. For $k = \{5, 10, 20\}$ individuals, DPSC improves the performance by 2.15%, 2.36% and 2.14% over the second best method DASC on ACC. For both NMI and PUR metrics, respectively, DPSC also brings about 3% improvement over the state-of-the-art DASC. All these results clearly prove the superior effectiveness and robustness of DPSC.

These results also clearly demonstrate that deep clustering methods perform much better than the shallow ones, benefiting from integrating representation learning with self-expression learning. The deep autoencoder extracts more powerful representations and the following self-expression layer enforces the representations to favorably locate in a u-

Table 3: Clustering results on COIL-20.

Metric	ACC	NMI	PUR
SSC [Elhamifar and Vidal, 2013]	0.8631	0.8892	0.8747
KSSC [Patel and Vidal, 2014]	0.7087	0.8243	0.7497
ENSC [You *et al.*, 2016]	0.8760	0.8952	0.8892
EDSC [Ji *et al.*, 2014]	0.8371	0.8828	0.8585
LRR [Liu *et al.*, 2013]	0.8118	0.8747	0.8361
LRSC [Favaro *et al.*, 2011]	0.7416	0.8452	0.7937
AE+SSC	0.8711	0.8990	0.8901
DSC-Net-L1 [Ji *et al.*, 2017]	0.9314	0.9353	0.9306
DSC-Net-L2 [Ji *et al.*, 2017]	0.9368	0.9408	0.9397
StructAE [Peng *et al.*, 2018]	0.9450	0.9485	0.9412
DASC [Zhou *et al.*, 2018c]	0.9639	0.9686	0.9632
DPSC	**0.9754**	**0.9792**	**0.9752**

Table 4: Clustering results on COIL-100.

Metric	ACC	NMI	PUR
SSC [Elhamifar and Vidal, 2013]	0.5500	0.5841	0.5720
KSSC [Patel and Vidal, 2014]	0.5282	0.6047	0.5534
ENSC [You *et al.*, 2016]	0.5732	0.5924	0.5843
EDSC [Ji *et al.*, 2014]	0.6187	0.6751	0.6547
LRR [Liu *et al.*, 2013]	0.4018	0.4721	0.4315
LRSC [Favaro *et al.*, 2011]	0.4933	0.5810	0.5450
AE+SSC	0.4607	0.4871	0.4782
DSC-Net-L1 [Ji *et al.*, 2017]	0.6638	0.6720	0.6701
DSC-Net-L2 [Ji *et al.*, 2017]	0.6904	0.7015	0.6972
StructAE [Peng *et al.*, 2018]	0.7143	0.7251	0.7203
DASC [Zhou *et al.*, 2018c]	0.7215	0.7286	0.7234
DPSC	**0.7540**	**0.7592**	**0.7585**

nion of linear subspaces, effectively getting rid of strict linear subspace assumptions. Comparatively, DPSC outperforms the DASC, StructAE and DSC-Net, on all metrics. This outstanding performance is attributed to the distribution consistency loss in DPSC. Unlike DPSC, the DASC, StructAE and DSC-Net do not consider the intrinsic cluster structure.

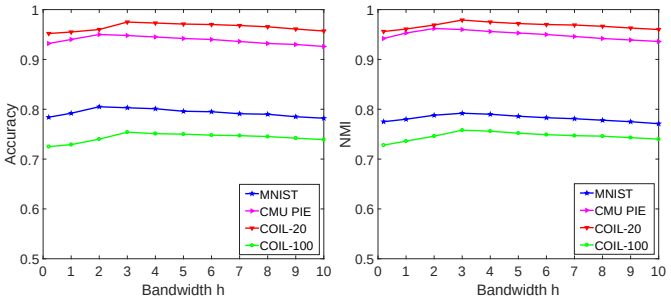
4.3 Object Clustering

We further evaluated DPSC on the challenging object clustering task using the COIL-20 and COIL-100 [Nene *et al.*, 1996] databases which provide various objects as shown in Fig. 3 (c). COIL-20 has 1,440 toy images from 20 classes, and COIL-100 contains 7,200 images of 100 objects. In both databases, each object was taken with poses varying at an interval of 5 degrees, producing a total of 72 images per object. This implies that the images are not distributed in a union of linear subspaces and thus are more challenging. In contrast with the previous human face databases, in which faces are well aligned and have similar structures, the object images from COIL-20 and COIL-100 are more diverse. Samples from the same object differ from each other due to the change of viewing angle, introducing additional challenge for subspace clustering techniques. For these databases, we downsample the images to 32×32 and set the bandwidth $h = 3$, $\lambda = \lambda_1 = 1$ and $\lambda_2 = 2$.

Table 3 and table 4 depict the ACC, NMI and PUR of different methods on clustering 20 classes for COIL-20 and 100 classes for COIL-100, respectively. Note that, in both cases and metrics, our DPSC achieves the best performance. In particular, for COIL-100, our method obtains an accuracy of 75.40%, which improves 3.25% over the best-performing baseline DASC.

4.4 Parameter Sensitivity

In our DPSC, there are four hyper-parameters λ, λ_1, λ_2 and h. Experiments show that the balancing parameters λ, λ_1 and λ_2 have slight influence to the performance. We set them by several attempts. Due to the limited space, we do not show the varying results with different balancing parameters. We focus on the effect of h, bandwidth of the kernel density estimator, which is important to the proposed method. Here, we report the clustering accuracy and NMI on four databases



(a) (b)

Figure 4: (a) The clustering accuracy of our DPSC with different bandwidth h. (b) The NMI of our DPSC with different bandwidth h.

with different bandwidth h. Fig. 4 shows that our DPSC is insensitive with different h in a certain range.

5 Conclusion

In this paper, we propose a novel distribution-preserving deep subspace clustering (DPSC) method. The distribution consistency loss of DPSC guarantees that the posterior latent representation distribution matches the prior original data space distribution and preserves the cluster structure of high-dimensional data space. This solves the inconvenient subspace distribution conditions. Extensive experiments on MNIST, CMU PIE and COIL-20/100 show the superiority of DPSC on both linear and non-linear subspace clustering problems over state-of-the-arts. In the future work, it is interesting to further investigate the cluster structure of complex sample distributions and try different distribution-preserving strategies.

Acknowledgments

This work was supported by the National Natural Science Foundation of China project no. 61772057 and the support funding from State Key Lab. of Software Development Environment and Qingdao Research Institute.

References

[Agarwal and Mustafa, 2004] Pankaj K Agarwal and Nabil H Mustafa. K-means projective clustering. In *PODS*, pages 155–165, 2004.

[Elhamifar and Vidal, 2013] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

[Favaro *et al.*, 2011] Paolo Favaro, Rene Vidal, and Avinash Ravichandran. A closed form solution to robust subspace estimation and clustering. In *CVPR*, pages 1801–1807, 2011.

[Hinneburg and Gabriel, 2007] Alexander Hinneburg and Hans-Henning Gabriel. Denclue 2.0: Fast clustering based on kernel density estimation. In *IDA*, pages 70–80. Springer, 2007.

[Ji *et al.*, 2014] Pan Ji, Mathieu Salzmann, and Hongdong Li. Efficient dense subspace clustering. In *WACV*, pages 461–468. IEEE, 2014.

[Ji *et al.*, 2017] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. In *NIPS*, pages 24–33, 2017.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Lecun *et al.*, 1998] Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li *et al.*, 2017a] Jun Li, Hongfu Liu, Handong Zhao, and Yun Fu. Projective low-rank subspace clustering via learning deep encoder. In *IJCAI*, pages 2145–2151, 2017.

[Li *et al.*, 2017b] Jun Li, Handong Zhao, Zhiqiang Tao, and Yun Fu. Large-scale subspace clustering by fast regression coding. In *IJCAI*, pages 2138–2144, 2017.

[Liu *et al.*, 2013] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[Lu *et al.*, 2012] Canyi Lu, Hai Min, Zhongqiu Zhao, Lin Zhu, Deshuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, pages 347–360, 2012.

[Lu *et al.*, 2014] Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. Correlation adaptive subspace segmentation by trace lasso. In *ICCV*, pages 1345–1352, 2014.

[Nene *et al.*, 1996] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library. 1996.

[Patel and Vidal, 2014] Vishal M. Patel and Rene Vidal. Kernel sparse subspace clustering. In *ICIP*, pages 2849–2853, 2014.

[Patel *et al.*, 2013] Vishal M Patel, Hien Van Nguyen, and Rene Vidal. Latent space sparse subspace clustering. In *ICCV*, pages 225–232, 2013.

[Peng *et al.*, 2015] Xi Peng, Zhang Yi, and Huajin Tang. Robust subspace clustering via thresholding ridge regression. In *AAAI*, pages 3827–3833, 2015.

[Peng *et al.*, 2016a] Chong Peng, Zhao Kang, Ming Yang, and Qiang Cheng. Feature selection embedded subspace clustering. *IEEE Signal Processing Letters*, 23(7):1018–1022, 2016.

[Peng *et al.*, 2016b] Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Zhang Yi. Deep subspace clustering with sparsity prior. In *IJCAI*, pages 1925–1931, 2016.

[Peng *et al.*, 2017] Xi Peng, Jiashi Feng, Jiwen Lu, Wei-Yun Yau, and Zhang Yi. Cascade subspace clustering. In *AAAI*, pages 2478–2484, 2017.

[Peng *et al.*, 2018] Xi Peng, Jiashi Feng, Shijie Xiao, Wei-Yun Yau, Joey Tianyi Zhou, and Songfan Yang. Structured autoencoders for subspace clustering. *IEEE Transactions on Image Processing*, 2018.

[Rao *et al.*, 2008] Shankar R Rao, Roberto Tron, Rene Vidal, and Yi Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *CVPR*, pages 1–8, 2008.

[Sim *et al.*, 2001] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression (pie) database of human faces. Technical Report CMU-RI-TR-01-02, Pittsburgh, PA, January 2001.

[Vidal *et al.*, 2005] Rene Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (gpca). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

[Vidal, 2011] René Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

[Wang *et al.*, 2018] Binshuai Wang, Xianglong Liu, Ke Xia, Kotagiri Ramamohanarao, and Dacheng Tao. Random angular projection for fast nearest subspace search. In *PCM*, pages 15–26. Springer, 2018.

[You *et al.*, 2016] Chong You, Chun-Guang Li, Daniel P Robinson, and Rene Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. In *CVPR*, pages 3928–3937, 2016.

[Zhou *et al.*, 2018a] Lei Zhou, Xiao Bai, Xianglong Liu, and Jun Zhou. Binary coding by matrix classifier for efficient subspace retrieval. In *ICMR*, pages 82–90, 2018.

[Zhou *et al.*, 2018b] Lei Zhou, Shuai Wang, Xiao Bai, Jun Zhou, and Edwin Hancock. Iterative deep subspace clustering. In *S+SSPR*, pages 42–51, 2018.

[Zhou *et al.*, 2018c] Pan Zhou, Yunqing Hou, and Jiashi Feng. Deep adversarial subspace clustering. In *CVPR*, pages 1596–1604, 2018.