

Deep Spectral Clustering using Dual Autoencoder Network

Xu Yang¹, Cheng Deng^{1*}, Feng Zheng², Junchi Yan³, Wei Liu^{4*}

¹School of Electronic Engineering, Xidian University, Xian 710071, China

²Department of Computer Science and Engineering, Southern University of Science and Technology

³Department of CSE, and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University

⁴Tencent AI Lab, Shenzhen, China

{xuyang.xd, chdeng.xd}@gmail.com, zhengf@sustc.edu.cn,
yanjunchi@sjtu.edu.cn, wl2223@columbia.edu

Abstract

The clustering methods have recently absorbed even-increasing attention in learning and vision. Deep clustering combines embedding and clustering together to obtain optimal embedding subspace for clustering, which can be more effective compared with conventional clustering methods. In this paper, we propose a joint learning framework for discriminative embedding and spectral clustering. We first devise a dual autoencoder network, which enforces the reconstruction constraint for the latent representations and their noisy versions, to embed the inputs into a latent space for clustering. As such the learned latent representations can be more robust to noise. Then the mutual information estimation is utilized to provide more discriminative information from the inputs. Furthermore, a deep spectral clustering method is applied to embed the latent representations into the eigenspace and subsequently clusters them, which can fully exploit the relationship between inputs to achieve optimal clustering results. Experimental results on benchmark datasets show that our method can significantly outperform state-of-the-art clustering approaches.

1. Introduction

As an important task in unsupervised learning [39, 8, 20] and vision communities, clustering has been widely used in image segmentation [33], image categorization [41], and digital media analysis [1]. The goal of clustering is to find a partition in order to keep similar data points in the same cluster while dissimilar ones in different clusters. In recent years, many clustering methods have been proposed, such as K -means clustering [24], spectral clustering [27, 42], and non-negative matrix factorization clustering [37], among which K -means and spectral clustering are two well-known

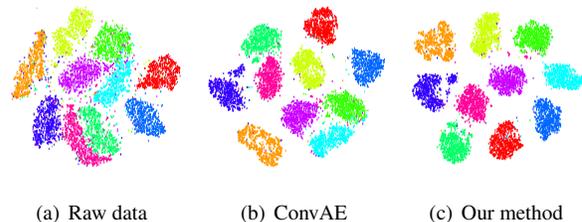


Figure 1. Visualizing the discriminative embedding capability on MNIST-test with t -SNE algorithm. (a): the space of raw data, (b): data points in the latent subspace of convolution autoencoder; (c): data points in the latent subspace of the proposed autoencoder network. Our method can provide a more discriminative embedding subspace.

conventional algorithms that are applicable to a wide range of various tasks. However, these shallow clustering methods depend on low-level features such as raw pixels, SIFT [28] or HOG [7] of the inputs. Their distance metrics are only exploited to describe local relationships in data space, and have limitation to represent the latent dependencies among the inputs [3].

This paper presents a novel deep learning based unsupervised clustering approach. Deep clustering, which integrates embedding and clustering processes to obtain optimal embedding subspace for clustering, can be more effective than shallow clustering methods. The main reason is that the deep clustering methods can effectively model the distribution of the inputs and capture the non-linear property, being more suitable to real-world clustering scenarios.

Recently, many clustering methods are promoted by deep generative approaches, such as autoencoder network [25]. The popularity of the autoencoder network lies in its powerful ability to capture high dimensional probability distributions of the inputs without supervised information. The encoder model projects the inputs into the latent

*Corresponding author.

space, and adopts an explicit approximation of maximum likelihood to estimate the distribution diversity between the latent representations and the inputs. Simultaneously, the decoder model reconstructs the latent representations to ensure the output maintaining all of the details in the inputs [34]. Almost all existing deep clustering methods endeavor to minimize the reconstruction loss. The hope is making the latent representations more discriminative which directly determines the clustering quality. However, in fact, the discriminative ability of the latent representations has no substantial connection with the reconstruction loss, causing the performance gap that is to be bridged in this paper.

We propose a novel dual autoencoder network for deep spectral clustering. First, a dual autoencoder, which enforces the reconstruction constraint for the latent representations and their noisy versions, is utilized to establish the relationships between the inputs and their latent representations. Such a mechanism is performed to make the latent representations more robust. In addition, we adopt the mutual information estimation to reserve discriminative information from the inputs to an extreme. In this way, the decoder can be viewed as a discriminator to determine whether the latent representations are discriminative. Fig. 1 demonstrates the performance of our proposed autoencoder network by comparing different data representations on MNIST-*test* data points. Obviously, our method can provide more discriminative embedding subspace than the convolution autoencoder network. Furthermore, deep spectral clustering is harnessed to embed the latent representations into the eigenspace, which followed by clustering. This procedure can exploit the relationships between the data points effectively and obtain the optimal results. The proposed dual autoencoder network and deep spectral clustering network are jointly optimized.

The main contributions of this paper are in three-folds:

- We propose a novel dual autoencoder network for generating discriminative and robust latent representations, which is trained with the mutual information estimation and different reconstruction results.
- We present a joint learning framework to embed the inputs into a discriminative latent space with a dual autoencoder and assign them to the ideal distribution by a deep spectral clustering model simultaneously.
- Empirical experiments demonstrate that our method outperforms state-of-the-art methods over the five benchmark datasets, including both traditional and deep network-based models.

2. Related Work

Recently, a number of deep learning-based clustering methods are proposed. Deep Embedding Clustering [36]

(DEC) adopts a fully connected stacked autoencoder network in order to learn the latent representations by minimizing the reconstruction loss in the pre-training phase. The objective function applied to the clustering phase is the Kullback Leibler (KL) divergence between the soft assignments of clustering modelled by a t -distribution. And then, a K -means loss is adopted at the clustering phase to train a fully connected autoencoder network [38], which is a joint approach of dimensionality reduction and K -means clustering. In addition, Gaussian Mixture Variational Autoencoder (GMVAE) [9] shows that minimum information constraint can be utilized to mitigate the effect of over-regularization in VAEs and provides an unsupervised clustering within the VAE framework considering a Gaussian mixture as a prior distribution. Discriminatively Boosted Clustering [21], a fully convolutional network with layer-wised batch normalization, adopts the same objective function as DEC and uses a boosting factor to the relatively train a stacked autoencoder.

Shah and Koltun [30] jointly solve the tasks of clustering and dimensionality reduction by efficiently optimizing a continuous global objective based on robust statistics, which allows heavily mixed clusters to be untangled. Following this method, a deep continuous clustering approach is suggested in [31], where the autoencoder parameters and a set of representatives defined against each data-point are simultaneously optimized. The convex clustering approach proposed by [6] optimizes the representatives by minimizing the distances between each representative and its associated data-point. Non-convex objectives are involved to penalize for the pairwise distances between the representatives.

Furthermore, to improve the performance of clustering, some methods combine convolutional layers with fully connected layers. Joint Unsupervised Learning (JULE) [40] jointly optimizes a convolutional neural network with the clustering parameters in a recurrent manner using an agglomerative clustering approach, where image clustering is conducted in the forward pass and representation learning is performed in the backward pass. Dizaji [10] proposes DEPICT, a method that trains a convolutional autoencoder with a softmax layer stacked on-top of the encoder. The softmax entries represent the assignment of each data-point to one cluster. VaDE [16] is a variational autoencoder method for deep embedding, and combines a Gaussian Mixture Model for clustering. In [15], a deep autoencoder is trained to minimize a reconstruction loss together with a self-expressive layer. This objective encourages a sparse representation of the original data. Zhou *et al.* [44] presents a deep adversarial subspace clustering (DASC) method to learn more favorable representations and supervise sample representation learning by adversarial deep learning [19]. However, the results of reconstruc-

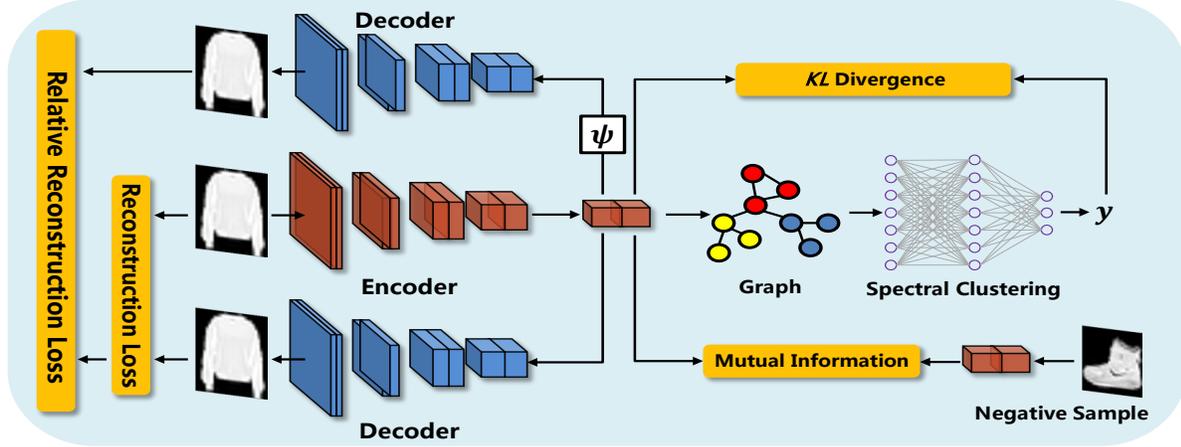


Figure 2. Illustration of the overall architecture. We first pre-train a dual autoencoder to embed the inputs into a latent space, and reconstruction results are obtained by the latent representations and their noise versions based on the noisy-transformer ψ . The mutual information calculated with negative sampling estimation is utilized to learn the discriminative information from inputs. Then, we assign the latent representations to the ideal clusters by a deep spectral clustering model, and jointly optimize the dual autoencoder and spectral clustering network simultaneously.

tion through low-dimensional representations are often very blurry. One possible way is to train a discriminator with adversarial learning but it can further increase the difficulty of training. Comparatively, our method introduces a relative reconstruction loss and mutual information estimation to obtain more discriminative representations, and jointly optimize the autoencoder network and the deep spectral clustering network for optimal clustering.

3. Methodology

As aforementioned, our framework consists of two main components: a dual autoencoder and a deep spectral clustering network. The dual autoencoder, which reconstructs the inputs using the latent representations and their noise versions, is introduced to make the latent representations more robust. In addition, the mutual information estimation between the inputs and the latent representations is applied to preserve the input information as much as possible. Then we utilize the deep spectral clustering network to embed the latent representations into the eigenspace and subsequently clustering is performed. The two networks are merged into a unified framework and jointly optimized with KL divergence. The framework is shown in Fig. 2.

Let $X = \{x_1, \dots, x_n\}$ denote the input samples, $Z = \{z_1, \dots, z_n\}$ denote their corresponding latent representations where $z_i = f(x_i; \theta_e) \in \mathbb{R}^d$ is learned by the encoder \mathbb{E} . The parameters of the encoder are defined by θ_e , and d is the feature dimension. $\tilde{x}_{z_i} = g(z_i; \theta_d)$ represents the reconstructed data point, which is the output of the decoder \mathbb{D} , and the parameters of the decoder are denoted by θ_d . We adopt a deep spectral clustering network \mathbb{C} to map z_i to $y_i = c(z_i; \theta_y) \in \mathbb{R}^K$, where K is the number of clusters.

3.1. Discriminative latent representation

We first train the dual autoencoder network to embed the inputs into a latent space. Based on the original reconstruction loss, we add a noise-disturbed reconstruction loss to learn the decoder network. In addition, we introduce the maximization of mutual information [13] to the learning procedure of the encoder network, so that the network can obtain more robust representations.

Encoder: Feature extraction is the major step in clustering and a good feature can effectively improve clustering performance. However, a single reconstruction loss cannot well guarantee the quality of the latent representations. We hope that the representations will help us to identify the sample from the inputs, which means it is the most unique information extracted from the inputs. Mutual information measures the essential correlation between two samples and can effectively estimate the similarity between features Z and inputs X . The definition of mutual information is defined as:

$$I(X, Z) = \iint p(z|x)p(x) \log \frac{p(z|x)}{p(z)} dx dz \quad (1)$$

$$= KL(p(z|x)p(x) || p(z)p(x)),$$

where $p(x)$ is the distribution of the inputs, $p(z|x)$ is the distribution of the latent representations, and the distribution of latent space $p(z)$ can be calculated by $p(z) = \int p(z|x)p(x)dx$. The mutual information is expected to be as large as possible when training the encoder network, hence we have:

$$p(z|x) = \max_{\theta_e} I(X, Z). \quad (2)$$

In addition, the learned latent representations are required to obey the prior distribution of the standard normal distribution with KL divergence. This is beneficial to make the latent space more regular. The distribution difference between $p(z)$ and its prior $q(z)$ is defined as.

$$KL(p(z)||q(z)) = \int p(z) \log \frac{p(z)}{q(z)} dz. \quad (3)$$

According to Eqs. (2) and (3), we have:

$$p(z|x) = \min_{\theta_e} \left\{ - \iint p(z|x)p(x) \log \frac{p(z|x)}{p(z)} dx dz + \alpha \int p(z) \log \frac{p(z)}{q(z)} dz \right\}. \quad (4)$$

It can be further rewritten as:

$$p(z|x) = \min_{\theta_e} \left\{ \iint p(z|x)p(x) \left[-(\alpha + 1) \log \frac{p(z|x)}{p(z)} + \alpha \log \frac{p(z|x)}{q(z)} \right] dx dz \right\}. \quad (5)$$

According to Eq. (1), the Eq. (5) can be viewed as:

$$p(z|x) = \min_{\theta_e} \left\{ -\beta I(X, Z) + \gamma \mathbb{E}_{x \sim p(x)} [KL(p(z|x)||q(z))] \right\}. \quad (6)$$

Unfortunately, KL divergence is unbounded. Instead of using KL divergence, JS divergence is adopted for mutual information maximization:

$$p(z|x) = \min_{\theta_e} \left\{ -\beta JS(p(z|x)p(x), p(z)p(x)) + \gamma \mathbb{E}_{x \sim p(x)} [KL(p(z|x)||q(z))] \right\}. \quad (7)$$

We have known that the variational estimation of JS divergence [29] is defined as:

$$JS(p(x)||q(x)) = \max_T \left(\mathbb{E}_{x \sim p(x)} [\log \sigma(T(x))] + \mathbb{E}_{x \sim q(x)} [\log(1 - \sigma(T(x)))] \right). \quad (8)$$

where $T(x) = \log \frac{2p(x)}{p(x)+q(x)}$ [29]. Here $p(z|x)p(x)$ and $p(z)p(x)$ are utilized to replace $p(x)$ and $q(x)$. As a result, Eq. (7) can be defined as:

$$p(z|x) = \min_{\theta_e} \left\{ -\beta (\mathbb{E}_{(x,z) \sim p(z|x)p(x)} [\log \sigma(T(x, z))] + \mathbb{E}_{(x,z) \sim p(z)p(x)} [\log(1 - \sigma(T(x, z)))] + \gamma \mathbb{E}_{x \sim p(x)} [KL(p(z|x)||q(z))] \right\}. \quad (9)$$

Negative sampling estimation [13], which is the process of using a discriminator to distinguish the real and noisy

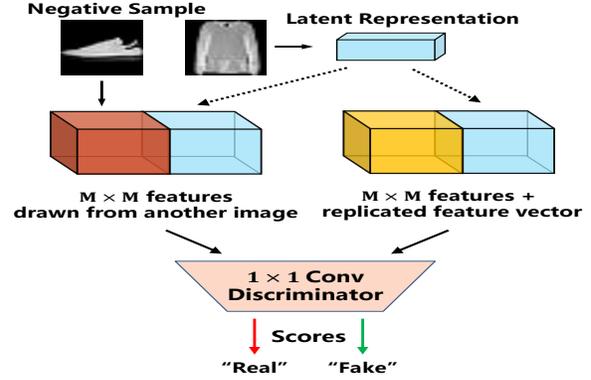


Figure 3. Local mutual information estimation.

samples to estimate the distribution of real samples, is generally utilized to solve the problem in Eq. (9). $\sigma(T(x, z))$ is a discriminator, where x and its latent representation z together form a positive sample pair. We randomly select z_t from the disturbed batch to construct a negative sample pair according to x . Note that Eq. (9) represents the global mutual information between X and Z .

Furthermore, we extract the feature map from the middle layer of the convolutional network, and construct the relationship between the feature map and the latent representation, which is the local mutual information. The estimation method plays the same role as global mutual information. The middle layer feature are combined with the latent representation to obtain a new feature map. Then a 1×1 convolution is considered as the estimation network of local mutual information, as shown in Fig. 3. The selection method of negative samples is the same as global mutual information estimation. Therefore, the objective function that needs to be optimized can be defined as:

$$L_e = -\beta (\mathbb{E}_{(x,z) \sim p(z|x)p(x)} [\log \sigma(T_1(x, z))] + \mathbb{E}_{(x,z) \sim p(z)p(x)} [\log(1 - \sigma(T_1(x, z)))] - \frac{\beta}{hw} \sum_{i,j} (\mathbb{E}_{(x,z) \sim p(z|x)p(x)} [\log \sigma(T_2(C_{ij}, z))] + \mathbb{E}_{(x,z) \sim p(z)p(x)} [\log(1 - \sigma(T_2(C_{ij}, z)))] + \gamma \mathbb{E}_{x \sim p(x)} [KL(p(z|x)||q(z))], \quad (10)$$

where h and w represent the height and width of the feature map. C_{ij} represents the feature vector of the middle feature map at coordinates (i, j) and $q(z)$ is the standard normal distribution.

Decoder: In the existing decoder networks, the reconstruction loss is generally a suboptimal scheme for clustering, due to the natural trade-off between the reconstruction and the clustering tasks. The reconstruction loss mainly depends on the two parts: the distribution of the latent representations and the generative capacity of decoder network. How-

ever, the generative capacity of the decoder network is not required in the clustering task. Our real goal is not to obtain the best reconstruction results, but to get more discriminative features for clustering. We directly use noise disturbance in the latent space to discard known nuisance factors from the latent representations. Models trained in this fashion become robust by exclusion rather than inclusion, and are expected to perform well on clustering tasks, where even the inputs contain unseen nuisance [14]. A noisy-transformer ψ is utilized to convert the latent representations Z into their noisy versions \hat{Z} , and then the decoder reconstructs the inputs from \hat{Z} and Z . The reconstruction results can be defined as $\tilde{x}_{\hat{z}_i} = g(\hat{z}_i; \theta_d)$ and $\tilde{x}_{z_i} = g(z_i; \theta_d)$, and the relative reconstruction loss can be written as:

$$L_r(\tilde{x}_{\hat{z}_i}, \tilde{x}_{z_i}) = \|\tilde{x}_{\hat{z}_i} - \tilde{x}_{z_i}\|_F^2, \quad (11)$$

where $\|\cdot\|_F$ stands for the Frobenius norm. We also use the original reconstruction loss to ensure the performance of the decoder network and consider ψ as multiplicative Gaussian noise. The complete reconstruction loss can be defined as:

$$L_r = \|\tilde{x}_{\hat{z}_i} - \tilde{x}_{z_i}\|_F^2 + \delta \|x - \tilde{x}_{z_i}\|_F^2. \quad (12)$$

where δ stands for the strength of different reconstruction loss.

Hence, by considering all the items, the total loss of the autoencoder network can be defined as:

$$\min_{\theta_d, \theta_e} L_r + L_e. \quad (13)$$

3.2. Deep Spectral Clustering

The learned autoencoder parameters θ_e and θ_d are considered as an initial condition in the clustering phase. Spectral clustering can effectively use the relationship between samples to reduce intra-class differences, and produce better clustering results than K -means. In this step, we first adopt the autoencoder network to learn the latent representations. Next, a spectral clustering method is used to embed the latent representations into the eigenspace of their associated graph Laplacian matrix. All the samples will be subsequently clustered in this space. Finally, both the autoencoder parameters and clustering objective are jointly optimized.

Specifically, we first utilize the latent representations Z to construct the non-negative affinity matrix W :

$$W_{i,j} = e^{-\frac{\|z_i - z_j\|^2}{2\sigma^2}}. \quad (14)$$

The loss function of spectral clustering is defined as:

$$L_c = \mathbb{E}[W_{i,j} \|y_i - y_j\|^2], \quad (15)$$

where y_i is the output of the network. When we adopt the general neural network to output y , we randomly select a

minibatch of m samples at each iteration and thus the loss function can be defined as:

$$L_c = \frac{1}{m^2} \sum_{i,j=1}^m W_{i,j} \|y_i - y_j\|^2. \quad (16)$$

In order to prevent that all points are grouped into the same cluster in network maps, the output y is required to be orthonormal in expectation. That is to say:

$$\frac{1}{m} Y^T Y = I_{k \times k}, \quad (17)$$

where Y is a $m \times k$ matrix of the outputs whose i th row is y_i^T . The last layer of the network is utilized to enforce the orthogonality [32] constraint. This layer gets input from K units, and acts as a linear layer with K outputs, in which the weights are required to be orthogonal, producing the orthogonalized output Y for a minibatch. Let \tilde{Y} denote the $m \times k$ matrix containing the inputs to this layer for Z , a linear map that orthogonalizes the columns of \tilde{Y} is computed through its QR decomposition. Since integrated $A^T A$ is full rank for any matrix A , the QR decomposition can be obtained by the Cholesky decomposition:

$$A^T A = B B^T, \quad (18)$$

where B is a lower triangular matrix, and $Q = A(B^{-1})^T$. Therefore, in order to orthogonalize \tilde{Y} , the last layer multiplies \tilde{Y} from the right by $\sqrt{m}(L^{-1})^T$. Actually, \tilde{L} can be obtained from the Cholesky decomposition of \tilde{Y} and the \sqrt{m} factor is needed to satisfy Eq. (17).

We unify the latent representation learning and the spectral clustering using KL divergence. In the clustering phase, the last term of Eq. (10) can be rewritten as:

$$\mathbb{E}_{x \sim p(x)} [KL(p((y, z)|x) \| q(y, z))], \quad (19)$$

where $p((y, z)|x) = p(y|z)p(z|x)$ and $q(y, z) = q(z|y)q(y)$. Note $q(z|y)$ is a normal distribution with mean μ_y and variance 1. Therefore, the overall loss of the autoencoder and the spectral clustering network is defined as:

$$\min_{\theta_d, \theta_e, \theta_c} L_r + L_e + L_c. \quad (20)$$

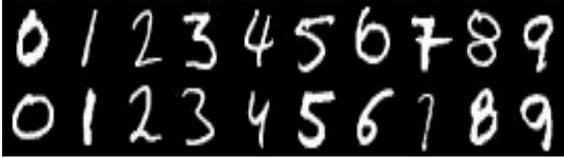
Finally, we jointly optimize the two networks until convergence to obtain the desired clustering results.

4. Experiments

In this section, we evaluate the effectiveness of the proposed clustering method in five benchmark datasets, and then compare the performance with several state-of-the-arts.

Table 1. Description of Datasets

Dataset	Samples	Classes	Dimensions
MNIST- <i>full</i>	70,000	10	1×28×28
MNIST- <i>test</i>	10,000	10	1×28×28
USPS	9298	10	1×16×16
Fashion-Mnist	70,000	10	1×28×28
YTF	10,000	41	3×55×55



(a) MNIST



(b) Fashion-Mnist

Figure 4. The image samples from the benchmark datasets used in our experiments

4.1. Datasets

In order to show that our method works well with various kinds of datasets, we choose the following image datasets. Considering that clustering tasks are fully unsupervised, we concatenate the training and testing samples when applicable. MNIST-*full* [18]: A dataset containing a total of 70,000 handwritten digits with 60,000 training and 10,000 testing samples, each being a 32×32 monochrome image. MNIST-*test*: A dataset only consists of the testing part of MNIST-*full* data. USPS: A handwritten digits dataset from the USPS postal service, containing 9,298 samples of 16×16 images. Fashion-MNIST [35]: This dataset has the same number of images and the same image size with MNIST, but it is fairly more complicated. Instead of digits, it consists of various types of fashion products. YTF: We adopt the first 41 subjects of YTF dataset and the images are first cropped and resized to 55×55 . Some image samples are shown in Fig. 4. The brief descriptions of the datasets are given in Tab. 1.

4.2. Clustering Metrics

To evaluate the clustering results, we adopt two standard evaluation metrics: Accuracy (ACC) and Normalized Mutual Information (NMI) [37].

The best mapping between cluster assignments and true labels is computed using the Hungarian algorithm to mea-

sure accuracy [17]. For completeness, we define ACC by:

$$ACC = \max_m \frac{\sum_{i=1}^n \mathbf{1}\{l_i = m(c_i)\}}{n}, \quad (21)$$

where l_i and c_i are the true label and predicted cluster of data point x_i .

NMI calculates the normalized measure of similarity between two labels of the same data, which is defined as:

$$NMI = \frac{I(l; c)}{\max\{H(l), H(c)\}}, \quad (22)$$

where $I(l, c)$ denotes the mutual information between true label l and predicted cluster c , and H represents their entropy. Results of NMI do not change by permutations of clusters (classes), and they are normalized to the range of $[0, 1]$, with 0 meaning no correlation and 1 exhibiting perfect correlation.

4.3. Implementation Details

In our experiments, we set $\beta = 0.01$, $\gamma = 1$, and $\delta = 0.5$. The channel numbers and kernel sizes of the autoencoder network are shown in Tab. 2, and the dimension of latent space is set to 120. The deep spectral clustering network consists of four fully connected layers, and we adopt ReLU [22] as the non-linear activations. We construct the original weight matrix W with probabilistic K -nearest neighbors for each dataset. The weight W_{ij} is calculated as nearest-neighbor graph [11], and the number of neighbors is set to 3.

4.4. Comparison Methods

We compare our clustering model with several baselines, including K -means [24], spectral clustering with normalized cuts (SC-Ncut) [33], large-scale spectral clustering (SC-LS) [4], NMF [2], graph degree linkage-based agglomerative clustering (AC-GDL) [43]. In addition, we also evaluate the performance of our method with several state-of-the-art clustering algorithms based on deep learning, including deep adversarial subspace clustering (DASC) [44], deep embedded clustering (DEC) [36], variational deep embedding (VaDE) [16], joint unsupervised learning (JULE) [40], deep embedded regularized clustering (DEPICT) [10], improved deep embedded clustering with locality preservation (IDEC) [12], deep spectral clustering with a set of nearest neighbor pairs (SpectralNet) [32], clustering with GAN (ClusterGAN) [26] and GAN with the mutual information (InfoGAN) [5].

4.5. Evaluation of Clustering Algorithm

We run our method with 10 random trials and report the average performance, the error range is no more than 2%. In terms of the compared methods, if the results of their methods on some datasets are not reported, we run the released

Table 2. Description the structure of the autoencoder network

Method	encoder-1/decoder-4	encoder-2/decoder-3	encoder-3/decoder-2	encoder-4/decoder-1
MNIST	$3 \times 3 \times 16$	$3 \times 3 \times 16$	$3 \times 3 \times 32$	$3 \times 3 \times 32$
USPS	$3 \times 3 \times 16$	$3 \times 3 \times 32$	-	-
Fashion-Mnist	$3 \times 3 \times 16$	$3 \times 3 \times 16$	$3 \times 3 \times 32$	$3 \times 3 \times 32$
YTF	$5 \times 5 \times 16$	$5 \times 5 \times 16$	$5 \times 5 \times 32$	$5 \times 5 \times 32$

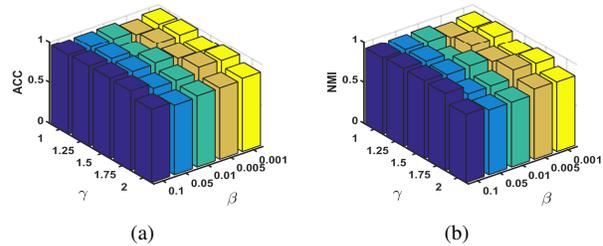
Table 3. Clustering performance of different algorithms on five datasets based on ACC and NMI

Method	MNIST- <i>full</i>		MNIST- <i>test</i>		USPS		Fashion-10		YTF	
	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC
K-means [24]	0.500	0.532	0.501	0.546	0.601	0.668	0.512	0.474	0.776	0.601
SC-Neut [33]	0.731	0.656	0.704	0.660	0.794	0.649	0.575	0.508	0.701	0.510
SC-LS [4]	0.706	0.714	0.756	0.740	0.755	0.746	0.497	0.496	0.759	0.544
NMF [2]	0.452	0.471	0.467	0.479	0.693	0.652	0.425	0.434	-	-
AC-GDL [43]	0.017	0.113	0.864	0.933	0.825	0.725	0.010	0.112	0.622	0.430
DASC [44]	0.784*	0.801*	0.780	0.804	-	-	-	-	-	-
DEC [36]	0.834*	0.863*	0.830*	0.856*	0.767*	0.762*	0.546*	0.518*	0.446*	0.371*
VaDE [16]	0.876	0.945	-	-	0.512	0.566	0.630	0.578	-	-
JULE [40]	0.913*	0.964*	0.915*	0.961*	0.913	0.950	0.608	0.563	0.848	0.684
DEPICT [10]	0.917*	0.965*	0.915*	0.963*	0.906	0.899	0.392	0.392	0.802	0.621
IDEC [12]	0.867*	0.881*	0.802	0.846	0.785*	0.761*	0.557	0.529	-	-
SpectralNet [32]	0.814	0.800	0.821	0.817	-	-	-	-	0.798	0.685
InfoGAN [5]	0.840	0.870	-	-	-	-	0.590	0.610	-	-
ClusterGAN [26]	0.890	0.950	-	-	-	-	0.640	0.630	-	-
Our Method	0.941	0.978	0.946	0.980	0.857	0.869	0.645	0.662	0.857	0.691

code with hyper-parameters mentioned in their papers, and the results are marked by (*) on top. When the code is not publicly available, or running the released code is not practical, we put dash marks (-) instead of the corresponding results.

The clustering results are shown in Tab. 3, where the first five are conventional clustering methods. In the table, we can notice that our proposed method outperforms the competing methods on these benchmark datasets. We observe that the proposed method can improve the clustering performance whether in digital datasets or in other product dataset. Especially when performing on the object dataset MNIST-*test*, the clustering accuracy is over 98%. Specifically, it exceeds the second best DEPICT which is trained on the noisy versions of the inputs by 1.6% and 3.1% on ACC and NMI respectively. Moreover, our method achieves much better clustering results than several classical shallow baselines. This is because compared with shallow methods, our method uses a multi-layer convolutional autoencoder as the feature extractor and adopts deep clustering network to obtain the most optimal clustering results. The Fashion-MNIST dataset is very difficult to deal with due to the complexity of samples, but our method still harvests good results.

We also investigate the parameter sensitivity on MNIST-*test*, and the results are shown in Fig. 5, where Fig. 5(a)

Figure 5. ACC and NMI of Our method with different β and γ on MNIST dataset

represents the results of ACC from different parameters and Fig. 5(b) is the results of NMI. It intuitively demonstrates that our method maintains acceptable results with most parameter combinations and has relative stability.

4.6. Evaluation of Learning Approach

We compare different strategies for training our model. For training a multi-layer convolutional autoencoder, we analyze the following four approaches: (1) convolutional autoencoder with original reconstruction loss (ConvAE), (2) convolutional autoencoder with original reconstruction loss and mutual information (ConvAE+MI), (3) convolutional autoencoder with improved reconstruction loss (ConvAE+RS) and (4) convolutional autoencoder with im-

Table 4. Clustering performance with different strategies on five datasets based on ACC and NMI

Method	MNIST- <i>full</i>		MNIST- <i>test</i>		USPS		Fashion-10		YTF	
	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC
ConvAE	0.745	0.776	0.751	0.781	0.652	0.698	0.556	0.546	0.642	0.476
ConvAE+MI	0.800	0.835	0.796	0.844	0.744	0.785	0.609	0.592	0.738	0.571
ConvAE+RS	0.803	0.841	0.801	0.850	0.752	0.798	0.597	0.614	0.721	0.558
ConvAE+MI+RS	0.910	0.957	0.914	0.961	0.827	0.831	0.640	0.656	0.801	0.606
ConvAE+MI+RS+SN	0.941	0.978	0.946	0.980	0.857	0.869	0.645	0.662	0.857	0.691

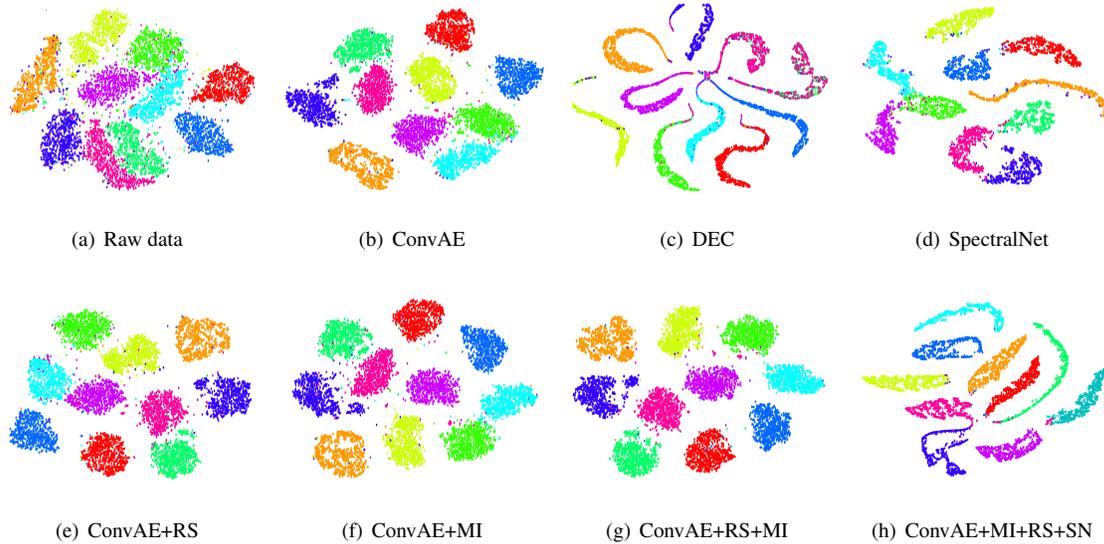


Figure 6. Visualization to show the discriminative capability of embedding subspaces using MNIST-*test* data.

proved reconstruction loss and mutual information (ConvAE+MI+RS). The last one is the joint training of convolutional autoencoder and deep spectral clustering. Tab. 4 represents the performance of different strategies for training our model. It clearly demonstrates that each kind of strategy of our method can improve the accuracy of clustering effectively, especially after adding mutual information and the improved reconstruction loss in the convolutional autoencoder network. Fig. 6 demonstrates the importance of our proposed strategy by comparing different data representations of MNIST-*test* data points using *t*-SNE visualization [23], Fig. 6(a) represents the space of raw data, Fig. 6(b) is the data points in the latent subspace of convolution autoencoder, Fig. 6(c) and 6(d) are the results of DEC and SpectralNet respectively, and the rest are our proposed model with different strategies. The results demonstrate the latent representations obtained by our method have more clear distribution structure.

5. Conclusion

In this paper, we propose an unsupervised deep clustering method with a dual autoencoder network and a deep spectral network. First, the dual autoencoder, which reconstructs the inputs using the latent representations and their

noise-contaminated versions, is utilized to establish the relationships between the inputs and the latent representations in order to obtain more robust latent representations. Furthermore, we maximize the mutual information between the inputs and the latent representations, which can preserve the information of the inputs as much as possible. Hence, the features of the latent space obtained by our autoencoder are robust to noise and more discriminative. Finally, the spectral network is fused to a unified framework to cluster the features of the latent space, so that the relationship between the samples can be effectively utilized. We evaluate our method on several benchmarks and experimental results show that our method outperforms those state-of-the-art approaches.

6. Acknowledgement

Our work was also supported by the National Natural Science Foundation of China under Grant 61572388, 61703327 and 61602176, the Key R&D Program-The Key Industry Innovation Chain of Shaanxi under Grant 2017ZDCXL-GY-05-04-02, 2017ZDCXL-GY-05-02 and 2018ZDXM-GY-176, and the National Key R&D Program of China under Grant 2017YFE0104100.

References

- [1] Lingling An, Xinbo Gao, Xuelong Li, Dacheng Tao, Cheng Deng, Jie Li, et al. Robust reversible watermarking via clustering and enhanced pixel-wise masking. *IEEE Trans. Image Processing*, 21(8):3598–3611, 2012.
- [2] Deng Cai, Xiaofei He, Xuanhui Wang, Hujun Bao, and Jiawei Han. Locality preserving nonnegative matrix factorization. In *IJCAI*, volume 9, pages 1010–1015, 2009.
- [3] Pu Chen, Xinyi Xu, and Cheng Deng. Deep view-aware metric learning for person re-identification. In *IJCAI*, pages 620–626, 2018.
- [4] Xinlei Chen and Deng Cai. Large scale spectral clustering with landmark-based representation. In *AAAI*, volume 5, page 14, 2011.
- [5] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.
- [6] Eric C Chi and Kenneth Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013, 2015.
- [7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
- [8] C Deng, E Yang, T Liu, W Liu, J Li, and D Tao. Unsupervised semantic-preserving adversarial hashing for image search. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 2019.
- [9] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.
- [10] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 5747–5756. IEEE, 2017.
- [11] Quanquan Gu and Jie Zhou. Co-clustering on manifolds. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 359–368. ACM, 2009.
- [12] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 1753–1759, 2017.
- [13] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.
- [14] Ayush Jaiswal, Rex Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Unsupervised adversarial invariance. In *Advances in Neural Information Processing Systems*, pages 5097–5107, 2018.
- [15] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. In *Advances in Neural Information Processing Systems*, pages 24–33, 2017.
- [16] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.
- [17] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [19] Chao Li, Cheng Deng, Ning Li, Wei Liu, Xinbo Gao, and Dacheng Tao. Self-supervised adversarial hashing networks for cross-modal retrieval. In *CVPR*, pages 4242–4251, 2018.
- [20] Chao Li, Cheng Deng, Lei Wang, De Xie, and Xianglong Liu. Coupled cyclegan: Unsupervised hashing network for cross-modal retrieval. *arXiv preprint arXiv:1903.02149*, 2019.
- [21] Fengfu Li, Hong Qiao, and Bo Zhang. Discriminatively boosted image clustering with fully convolutional autoencoders. *Pattern Recognition*, 83:161–173, 2018.
- [22] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.
- [23] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [24] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.
- [25] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011.
- [26] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. Clustergan: Latent space clustering in generative adversarial networks. *arXiv preprint arXiv:1809.03627*, 2018.
- [27] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.
- [28] Pauline C Ng and Steven Henikoff. Sift: Predicting amino acid changes that affect protein function. *Nucleic acids research*, 31(13):3812–3814, 2003.
- [29] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pages 271–279, 2016.
- [30] Sohil Atul Shah and Vladlen Koltun. Robust continuous clustering. *Proceedings of the National Academy of Sciences*, 114(37):9814–9819, 2017.
- [31] Sohil Atul Shah and Vladlen Koltun. Deep continuous clustering. *arXiv preprint arXiv:1803.01449*, 2018.

- [32] Uri Shaham, Kelly Stanton, Henry Li, Boaz Nadler, Ronen Basri, and Yuval Kluger. Spectralnet: Spectral clustering using deep neural networks. *arXiv preprint arXiv:1801.01587*, 2018.
- [33] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.
- [34] Elad Tzoreff, Olga Kogan, and Yoni Choukroun. Deep discriminative latent space for clustering. *arXiv preprint arXiv:1805.10795*, 2018.
- [35] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [36] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.
- [37] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.
- [38] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. *arXiv preprint arXiv:1610.04794*, 2016.
- [39] Erkun Yang, Cheng Deng, Tongliang Liu, Wei Liu, and Dacheng Tao. Semantic structure-based unsupervised deep hashing. In *IJCAI*, pages 1064–1070, 2018.
- [40] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016.
- [41] Muli Yang, Cheng Deng, and Feiping Nie. Adaptive-weighting discriminative regression for multi-view classification. *Pattern Recogn.*, 88(4):236–245, 2019.
- [42] Xu Yang, Cheng Deng, Xianglong Liu, and Feiping Nie. New l2, 1-norm relaxation of multi-way graph cut for clustering. In *AAAI*, 2018.
- [43] Wei Zhang, Deli Zhao, and Xiaogang Wang. Agglomerative clustering via maximum incremental path integral. *Pattern Recognition*, 46(11):3056–3065, 2013.
- [44] Pan Zhou, Yunqing Hou, and Jiashi Feng. Deep adversarial subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1596–1604, 2018.