IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 22, 2020, accepted August 2, 2020, date of publication August 6, 2020, date of current version August 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014631

Regression Based Clustering by Deep Adversarial Learning

FEI TANG[1,2], DABIN ZHANG[1], TIE CAI[2], AND QIN LI[2]
[1]College of Mathematics and Informatics, South China Agriculture University, Guangzhou 510642, China
[2]School of Software Engineering, Shenzhen Institute of Information Technology, Shenzhen 518172, China

Corresponding author: Dabin Zhang (zdbff@scau.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 71971089.

ABSTRACT Despite the great success, existing regression clustering methods based on shallow models are vulnerable due to: (1) They often pay no attention to the combination between learning representations and clustering, thus resulting in unsatisfactory clustering performance. (2) They ignore the relationship of data distribution and target distribution such that those methods are noise and illumination-change sensitive. (3) These nonlinear regression methods usually impose the hard constraint to minimize the mismatch between the discrete cluster assignment matrix and latent representations, which leads to over-fitting. In this paper, we utilize deep adversarial regression to tackle these problems and formulate regression based clustering by deep adversarial learning (RCDA). By seamlessly combining with the stacked autoencoder, the proposed model integrates learning deep nonlinear latent representation and clustering in a unified framework. Specifically, RCDA uses a kind of relax constraint between latent representations and continuous cluster assignment matrix to avoid over-fitting, and simultaneously utilizes the t-SNE algorithm and adversarial learning to analyze data distribution and target distribution so that improve representations learning. Experimental results on public benchmark datasets demonstrate that the proposed architecture achieves better performance than state-of-the-art clustering models in image clustering task.

INDEX TERMS Unsupervised learning, image clustering, regression based clustering.

I. INTRODUCTION

Clustering, primitive exploration with little or no prior knowledge, is one of the most indispensable and fundamental research topics in artificial intelligence research, and applies in many fields such as image retrieval, image annotation, document analysis and image segmentation, etc. In the past few decades, many classic clustering algorithms have been proposed, including spectral clustering (SC) [1], [2], subspace clustering [3], [4], graph based clustering [5] and so on. Despite extensive study, the performance of traditional clustering methods deteriorates with high dimensional data due to unreliable similarity metrics, known as the curse of dimensionality, when working with large-scale real-world image datasets.

To deal with the problem of dimensional curse, a common way is to transform data from a high dimensional data space to a lower feature space by applying hand-crafted feature extraction or dimension reduction techniques like principle component analysis (PCA), scale invariant feature transform (SIFT feature) and histogram of oriented gradients (HOG feature). Then, clustering can be performed in the lower dimensional feature space. However, these hand-crafted features ignore the interconnection between features learning and clustering. To address this issue, Torre and Kanade [6] propose a shallow model to perform clustering and feature learning simultaneously by integrating K-Means and linear discriminant analysis (LDA) into a joint framework. Nevertheless, the representation ability of features learned via these shallow models is limited.

To address the above challenges, lately, deep clustering models have emerged, which apply deep neural network to clustering tasks. For instance, Tian *et al*. [7] utilizes deep neural networks (DNNs) to transform feature at first phase, and then clustering. Xie *et al*. [8] propose deep embedded clustering (DEC) that simultaneously learns feature representations and cluster assignments using DNN, in which feature mapping and clustering are jointly learned. Guo *et al*. [9] present an improved deep embedded clustering (IDEC) method with local structure preserved based on DEC. Dizaji *et al*. [10]

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Wang.

base their new deep clustering model, termed DEPICT on a multi-layer convolutional autoencoder, in which a regularized relative entropy loss function is employed for clustering.

Traditional clustering methods refer to unsupervised settings. Regression as one kind of classic machine learning algorithm has been applied to deal with many classic supervised learning tasks, e.g., object classification and face recognition [11]. When the prior label knowledge of instances is unknown in regression learning, it becomes unsupervised regression. However, there are few studies to utilize the property of regression to tackle clustering tasks. In practice, many high dimensional data may exhibit dense grouping in a low dimensional subspace, and the true cluster indicator matrix can be always embedded into a low dimensional mapping of the data [12]. Hence, regression can help guide the partitioning process by modeling the dissimilarity of each cluster in the low dimensional subspace. To take advantage of this property, [13] developed a local and global discriminative framework for balanced clustering via minimizing distribution entropy and the least-squares regression between cluster indicator matrix and low dimensional features. Since the cluster indicator matrix consists of discrete binary values, on the contrary, the low-dimensional feature is continuous. This hard constraint allows continuous values to approximate discrete values, which may make the overall model hard to optimize. In order to embed the discriminative information in the cluster indicator matrix of spectral clustering, thereby boosting clustering performance, [14] proposed to take controlling the regression constraint between the cluster indicator matrix and the latent features of the data into account, in which relaxing the cluster indicator matrix is considered, but keep the orthogonality intact. This problem also exists in [15], in which a robust regression-based clustering method was presented to tackle cancer genome data. However, a vital constraint is usually ignored by above methods, *i.e.*, all the elements of the cluster indicator matrix should be nonnegative by definition.

Although all the above regression-based clustering methods provide impressive results, they still have several limitations: 1) Overlooking the relationship of data distribution makes the model sensitive to noise and illumination changes; 2) Being unable to capture the no-linear structure of data, because these methods based on shallow and linear objective function; 3) Using strict restrictions, which leads to algorithms overfitting; 4) Separation of the latent representation and clustering.

To handle the problems mentioned above, motivated by DEC [8], stacked autoencoder [16] and adversarial learning [17], we propose a novel deep adversarial regression clustering model (**RCDA**) to learn an effective parameterized non-linear mapping from the data space $\mathbf{X}$ to a lower-dimensional feature space $\mathbf{F}$, which takes the advantages of regression clustering methods, deep embedding models and adversarial learning. RCDA basically consists of two training procedures: pre-training of the autoencoder and training of deep adversarial regression model. The pre-trained autoencoder makes sure the output of encoder is reliable. RCDA simultaneously solves for cluster assignment and the underlying feature representation via iteratively refining clusters with regression clustering loss and an auxiliary target distribution derived from the current data distribution. The adversarial learning between target distribution and data distribution significantly improves the effectiveness of two distributions, thereby improving the effectiveness of the feature representation. Moreover, experimental results show that RCDA achieves superior results compared to the state-of-the-art algorithms on the image benchmark datasets. The main contributions of this paper are summarized as follows:

- We propose a novel deep adversarial regression clustering architecture RCDA to simultaneously learn feature transformation and cluster assignment. To our best knowledge, this is the first work that uses the property of deep learning to help regression-based clustering.
- We derive a loss function to guide agglomerative clustering and deep representation learning which makes optimization over the two tasks seamless.
- We propose a method to make the learned data distribution and target distribution more effective, thereby achieving superior clustering results on high-dimensional and large-scale datasets.

II. RELATED WORKS

A. REGRESSION CLUSTERING

Regression-based clustering [14] is one of the most representative clustering methods. The objective is

$$\min_{\mathbf{W},\mathbf{b},\mathbf{L}} \|\mathbf{X}^T\mathbf{W} + \mathbf{1}\mathbf{b}^T - \mathbf{L}\|_F^2 + \xi\|\mathbf{W}\|_F^2, \tag{1}$$

where ξ is the penalty coefficient. $\mathbf{W}$ and $\mathbf{b}$ are the parameters. $\mathbf{X}$ and $\mathbf{L}$ are the raw data matrix and the cluster indicator matrix, respectively. Problem 1 leverages hard constraint to make the continuous low-dimensional features approximate to the discrete cluster indicator matrix $\mathbf{L}$. However, discrete zero and one elements are too ideal, leading to a suboptimal solution. Although some methods usually relax the cluster indicator matrix, keep the orthogonality intact. Under this circumstance, the relaxed solution may severely deviate from the true solution and thus degrade the clustering performance. Because, all the elements of the cluster indicator matrix should be nonnegative by definition. Also, they use K-Means to cluster indicator matrix to get the clustering results in the last step, this postprocessing operation will increase the instability of the original performance due to the uncertainty of K-Means. Moreover, these methods directly utilize the handcraft features and dimension reduction skills, which neglect the distribution of input data. And these shallow methods cannot model the non-linear structure of image data so that the algorithms is not robust enough. RCDA takes full advantages of stacked autoencoder to transform the data with a nonlinear mapping and integrate clustering and representation learning in a unified framework, which can consistently produce

semantically meaningful and well-separated representations on real-world datasets.

B. DEEP CLUSTERING

Deep clustering is a new kind of clustering that has arisen in recent years. Inspired by the similarity between eigenvalue composition in spectral methods and stacked autoencoder [16] in learning lower-dimensional representation, Tian *et al*. [7] were the first to introduce DNN to tackle clustering tasks, which combines a nonlinear embedding of the original graph and K-Means algorithm in the embedding feature space. Law [18] proposed a deep supervised clustering metric learning method to learn data representation, given the ground-truth partition. These methods mentioned above firstly learn representations in a low dimensional feature space, and then run clustering algorithm on the embedding space, which can be divided into a two-stage procedure. RCDA integrates unsupervised learning of deep representations and clustering into a framework. Yang *et al*. [19] proposed a recurrent framework for joint unsupervised learning of deep representations and image clusters. Unlike these models that ignore the distribution of input data and target distribution, RCDA utilizes Student's t-distribution as a kernel to measure the distribution of input data. Xie *et al*. [8] and Guo *et al*. [9] use KL divergence between soft assignment and target distribution minimization to simultaneously learn feature representations and cluster assignments in a deep neural network. Although these two methods consider the data distribution and target distribution, they ignore the noise between distributions. Differently, RCDA utilizes the adversarial learning between data and target distribution to suppress the noise, thus improving the performance of clustering.

III. REGRESSION BASED CLUSTERING BY DEEP ADVERSARIAL LEARNING

In this section, we first elaborate the representation learning model and clustering module of the RCDA. Then, we will introduce the implementation details of RCDA. Our model is made up of three sub-networks: one stack fully-connected autoencoder (encoder: $\mathbf{E}_n$, decoder: $\mathbf{D}_e$) that is used to learn latent representations, one deep embedding clustering layer that to cluster samples, and one discriminator $\mathbf{D}$ that is used to supervise the clustering. Figure 1 shows the framework of our model with example $\mathbf{X}$, the detailed information of the framework will be given as follows.

Notations: For ease of explanation, suppose we aim to cluster N instances$\{\mathbf{x}_i \in \mathbf{X}\}_{i=1}^{N}$ into K clusters according to their feature attributes, where the label information of each instance is unknown. Meanwhile, we utilize $\mu_j (j = 1, 2, \cdots K)$ to represent the centroid of each cluster.

A. REPRESENTATION LEARNING MODEL

To learn the latent representations $\mathbf{F} \in \mathbb{R}^{N\times K}$, we introduce the encoder $\mathbf{E}_n$ and decoder $\mathbf{D}_e$: $\mathbb{R}^{N\times d} \rightarrow \mathbb{R}^{N\times K} \rightarrow \mathbb{R}^{N\times d}$. The autoencoder consists of four fully connected layers, aims to learn a latent feature $\mathbf{F} = \{\mathbf{f}_1, \cdots, \mathbf{f}_i, \cdots, \mathbf{f}_N\}(\mathbf{f} \in \mathbb{R}_i^{N\times K})$ of original input data $\mathbf{X}$. There we choose autoencoder based on the fact that autoencoder consistently produces semantically meaningful and well-separated representations on real-world datasets. To be specific, encoder transforms the raw input data to a low-dimensional representation $\mathbf{F}$ via a non-linear mapping

$$\mathbf{f}_i = \mathbf{E}_n(x_i; \theta) = \mathbf{E}_n(\mathbf{x}_i^T \mathbf{W} + \mathbf{1}\mathbf{b}^T), \tag{2}$$

where $\mathbf{E}_n$ refers to the non-linear function and $\theta = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}$ is the l-th layer's learnable parameters of encoder $\mathbf{E}_n$. Then, a decoder is exploited to reconstruct the input data $\mathbf{X}$ from low-dimensional representation, where the output of decoder is the reconstructed data $\widetilde{\mathbf{X}}$. To ensure that the latent features obtained by the encoder are effective, the network minimizes the least mean square loss L_{AE} between $\mathbf{X}$ and $\widetilde{\mathbf{X}}$ to update the learnable parameters of $\mathbf{E}_n$ and $\mathbf{D}_e$, so we have

$$\begin{aligned} L_{AE} &= \min_{\theta,\omega,\mathbf{E}_n,\mathbf{D}_e} \frac{1}{N}(\mathbf{X} - \mathbf{D}_e(\mathbf{E}_n(\mathbf{X}; \theta); \omega))^2 \\ &= \min_{\theta,\omega,\mathbf{E}_n,\mathbf{D}_e} \frac{1}{N}(\mathbf{X} - \widetilde{\mathbf{X}})^2, \end{aligned} \tag{3}$$

This loss is used for training encoder $\mathbf{E}_n$, and decoder $\mathbf{D}_e$. It encourages encoder to catch essential structure for the latent representation from input data, and the latent representation recover the real data exactly. The encoder $\mathbf{E}_n$ takes $\mathbf{X}$ as input and learns one latent representations $\mathbf{F} = \mathbf{E}_n(\mathbf{X}; \theta)$. The decoder reconstructs the single view from the latent representation $\mathbf{F}$. The output is $\widetilde{\mathbf{X}} = \mathbf{D}_e(\mathbf{E}_n(\mathbf{X}; \theta); \omega)$. $\omega = \{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}$ represents learnable parameters of $\mathbf{D}_e$ at m-th layer.

We minimize the reconstruction error between the output of decoder and the input of encoder to optimize the encoder and decoder networks in Eq. (3). In order to improve the performance of clustering and ensure the prime target distribution is available, we pre-train the encoder and decoder.

B. CLUSTERING MODELS

To perform clustering, we map the output of the encoder, *i.e.* $\mathbf{F}$, to the corresponding clusters by using the t-SNE like algorithm. To be specific, given an initial estimate of the non-linear mapping $\mathbf{f}_i$, we get a latent representation $\mathbf{F}$. Unlike t-SNE [20], we employ a mapping function *Student's t-distribution* to measure the similarity between representation $\mathbf{f}_i$ of data point $\mathbf{x}_i$ and cluster centroid μ_j instead of measuring the similarity between data point $\mathbf{x}_i$ and data point $\mathbf{x}_j$. Hence, we can calculate the soft cluster assignment by

$$q_{ij} = \frac{(1 + ||\mathbf{f}_i - \mu_j||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1 + ||\mathbf{f}_i - \mu_{j'}||^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{4}$$

where q_{ij} is the probability of assigning sample i to cluster j, α is the degree of freedom of the *Student's t-distribution*. The K centroids $\{\mu_j\}_{j=1}^{K}$ is defined the trainable parameters, and the initial values of μ is obtained by implement K-Means on latent representations $\mathbf{F}$. We herein call matrix

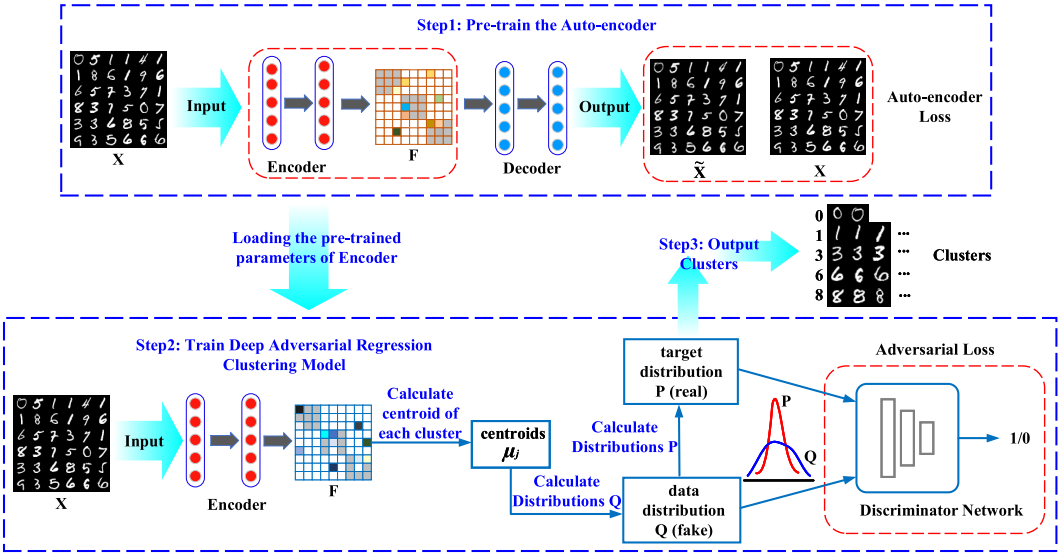

FIGURE 1. The framework of regression based clustering by deep adversarial learning (RCDA) with unlabeled examples **X**. Totally, there are three steps in the proposed model **Step 1**, pre-train the autoencoder via a set of examples **X**. At the beginning of **Step 2**, load the pre-trained parameters of encoder in **Step 1**, and set them as initial parameters of the encoder in **Step 2**, next we train the RCDA model. When the loss function of **Step 2** is up to convergence condition, the model outputs the clusters of unlabeled samples **X** in **Step 3**.

Q the actual distribution. In order to simultaneously relax the discrete values of **L** in Eq. (1) and supervise the quality of clustering, thereby improve clustering performance, we here introduce a target distribution **P** as cluster indicator matrix. Thus, the regression-based clustering objective is defined by

$$L_C = \lambda_{21} \|\mathbf{F} - \mathbf{P}\|_F^2 + \lambda_{22} \|\mathbf{P} - \mathbf{Q}\|_F^2 \tag{5}$$

where λ_{21} and λ_{22} are two tradeoff parameters. The first item is regression-based clustering objective, the second item is employed to supervise clustering. Our aim is to match the soft assignment **Q** to the target distribution **P**. In this way, we can sharpen the data distribution and concentrate the same class data. In addition, we will get a more effective and latent representation for clustering task.

We hope the target distribution has the following properties: 1) it can further emphasize more on the nodes assigned with high confidence, 2) it can strengthen predictions, 3) it can prevent large clusters from distorting the latent representations of the nodes. Hence, we computer target distribution p_{ij} by first raising q_{ij} to the squared and then normalizing by frequency per cluster. Hence, we have

$$p_{ij} = \frac{q_{ij}^2 / t_i}{\sum_{j'} q_{ij'}^2 / t_{j'}}, \tag{6}$$

where $t_j = \sum_i q_{ij}$ is soft cluster frequency of each cluster, which is adopted to normalize the loss contribution itself so that distorting the hidden space by larger clusters is prevented. In our method, we raise q_{ij} to the second power (squared closeness), because it can simultaneously suppresses the responses from dissimilar points and enhances the responses from similar points, which makes the result more robust and sparser, as shown in Fig. 2.

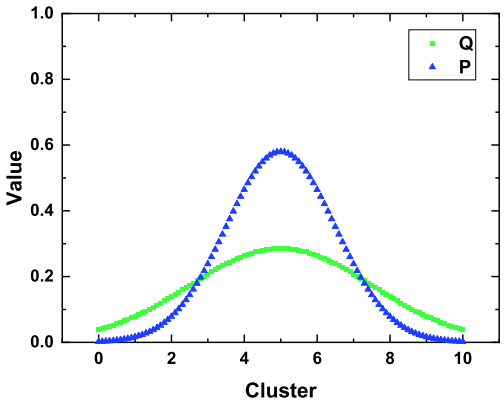

FIGURE 2. Illustration of the effectiveness of squared closeness. The green curve is the closeness, and the blue curve is the squared closeness.

C. ADVERSARIAL MODELS

Although the error between distribution **P** and **Q** can be measured by $\|\mathbf{P} - \mathbf{Q}\|_F^2$ in Eq. (5), it cannot ensure that the differences in salient features is small. Accordingly, we utilize the adversarial learning to tackle this problem, that is to say, we introduce the adversarial learning between **P** and **Q** to further minimize the mismatch of them. Hence, in adversarial learning phase, we take autoencoder as a generator and combined a discriminator to make up a GAN-like module in RCDA. The discriminator aims to distinguish the target distributions **P** and the actual distribution **Q**, and it consists of three-layer fully connected networks. For **D**, we hope it

can distinguish that $\mathbf{Q}$ is the actual distribution of input data points and $\mathbf{P}$ is the real target distribution. The loss function of autoencoder minimizes the likelihood that data distribution $\mathbf{Q}$ assigns to the fake source, while the discriminator is maximizes the likelihood that data distribution $\mathbf{Q}$ is assigned to the fake source, so the objective of adversarial learning is

$$L_D = \min_{\mathbf{E}_n, \mathbf{D}_e} \max_{\mathbf{D}} \lambda_{11} \mathbb{E}\left[\log(\mathbf{D}(\mathbf{P}))\right] + \lambda_{12} \mathbb{E}\left[\log(1 - \mathbf{D}(\mathbf{Q}))\right], \tag{7}$$

where λ_{11} and λ_{12} are two tradeoff parameters.

The encoder is trained to generate data distribution $\mathbf{Q}$ which are similar to target distribution $\mathbf{P}$. The discriminators is trained to distinguish the data distribution $\mathbf{Q}$ from the real data. They play a min-max game until convergence. The adversarial loss can assist the encoder in mapping a given sample $\mathbf{X}$ to a desired output $\mathbf{F}$. Thus, the combination of adversarial loss and clustering loss further ensures the encoder map the input data points $\mathbf{X}$ to a desired latent representations, thereby boosting clustering performance.

D. IMPLEMENTATION DETAILS

In this section, we present the detailed implementation of the unsupervised regression based clustering model. The overall objective function of our model contains three terms covering autoencoder loss, adversarial loss, and clustering loss, each being linked to one sub-network of our model. The overall objective function of RCDA is given by

$$L = \min_{\mathbf{E}_n, \mathbf{D}_e, \mu} \max_{\mathbf{D}} L_{AE} + L_D + L_C. \tag{8}$$

1) STEP 1: PRE-TRAINING ENCODER $\mathbf{E}_n$ AND DECODER $\mathbf{D}_e$

We utilize original data to train stacked encoder and decoder because the unsupervised representation learned by stacked autoencoder naturally facilitates the learning of clustering representations with RCDA. Similar to Vincent *et al.* [16] we initialize the SAE network layer by layer with each layer being a denoising autoencoder trained to reconstruct the previous layer's output after random corruption. After greedy layer-wise training, we concatenate all encoder layers followed by all decoder layers, in reverse layer-wise training order, to form a deep autoencoder and then fine-tune it to minimize reconstruction loss and update the parameters of $\theta, \omega, \mathbf{E}_n, \mathbf{D}_e$. The final result is a multi-layer deep autoencoder with a bottleneck coding layer in the middle.

2) STEP 2: TRAINING ENCODER $\mathbf{E}_n$, CLUSTERING LAYER AND DISCRIMINATOR $\mathbf{D}$ ON ALL DATA

We discard the decoder layers and use the encoder layers as our initial mapping between the data space and the feature space. Then we pass the data through the initialized encoder $\mathbf{E}_n$ to get latent representation point $\mathbf{f}_i$ and then perform standard K-means clustering in the feature space $\mathbf{F}$ to obtain k initial centroids $\{\mu_j\}_{j=1}^{k}$. According to the centroids μ_j and representation point $\mathbf{f}_i$, we next utilize Eq. (4) to calculate the data distribution $\mathbf{Q}$, and easily calculate the target distribution $\mathbf{P}$ via Eq. (6). Finally, we enter the distribution $\mathbf{P}$ and $\mathbf{Q}$ into discriminator networks for adversarial learning. We minimize the total loss function Eq. (4) to alternately optimize the parameters of discriminator network, centroids μ_j and autoencoder network via back propagation algorithm until the objective function converges. Algorithm 1 reports a brief description of RCDA model.

Algorithm 1: Regression Based Clustering by Deep Adversarial Learning

Input: Data $\mathbf{X} \in \mathbf{R}^{N \times d}$, number of clusters: K, Parameter $\lambda_{11}, \lambda_{12}, \lambda_{21}, \lambda_{22}$.
Output: Cluster label c_i of $\mathbf{x}_i \in \mathbf{X}$.

```
Randomly initialize the parameters of E_n, D_e;
for not converged do
    // Step 1 ⟶ Pre-train the autoencoder
    Updating E_n and D_e by Eq. (3);
end
Use the pre-trained parameters of E_n and D_e to project raw sample and gain F;
Implement K-Means on feature space F, obtain the initial clustering centroids {μ_j}_{j=1}^K;
Calculate initial Q and P via Eqs. (4, 6);
Input Q and P to discriminator networks;
for not converged do
    // Step 2 ⟶ Jointly training overall networks.
    Alternately updating autoencoder and the discriminator by Eq. (8), where the centroid of j-th cluster is updated by
    μ_{j+1} ← μ_j + 2η Σ_{i=1}^N (1 + ||z_i − μ_j||^2)^{-1} × (p_ij − q_ij)(z_i − μ_j)//η: learning rate of autoencoder;
end
for all x_i ∈ X do
    // Step 3 ⟶ Calculating the clusters
    c_i := maxIndex(p_i), p_i ∈ P^{N×K};
end
return: Cluster label c_i.
```

IV. EXPERIMENTS

In this section, we apply the proposed RCDA model to image clustering and evaluate the performance on four popular datasets (MNIST, CIFAR10, CIFAR100 and STL-10) with three frequently-used measures (Accuracy, Normalized Mutual Information and Adjusted Rand Index).

A. DATASETS AND EXPERIMENTAL SETTINGS

Four widely-used clustering benchmark datasets *i.e.* MNIST [21], CIFAR-10 [22], CIFAR-100 [22] and STL-10 [23] are used to verify the effectiveness of the proposed method. Statistics of four datasets are shown in Table 1.

TABLE 1. Details of datasets, where # means the number of.

Dataset	Attribute dim	# classes	# samples
MNIST [21]	784	10	70,000
CIFAR-10 [22]	3,072	10	60,000
CIFAR-100 [22]	3,072	20	60,000
STL-10 [23]	1,428	10	13,000

TABLE 2. Details of hyper-parameter, where # means the number of.

Parameters	Value	Parameters	Value
λ_{11}	0.5	Learning rate of $\mathbf{E}_n, \mathbf{D}_e$	0.1
λ_{12}	1	Learning rate of $\mathbf{D}$	0.001
λ_{21}	0.0001	Learning rate of SAE	0.1
λ_{22}	1	# Iterations of K-Means	20

• **MNIST** The MNIST dataset consists of 70000 handwritten digits of 28 × 28 pixel size. The digits are centered and size-normalized. We transform the image to a vector (dimension is 784 = 28 × 28) as input to all algorithms.

• **CIFAR-10** The CIFAR-10 dataset consists of 60000 32 × 32 colour images in 10 classes, with 6000 images per class. We transform the color image to a vector (dimension is 3072 = 32 × 32 × 3) as input to all algorithms.

• **CIFAR-100** This dataset is just like the CIFAR-10, except it has 100 classes containing 600 images each. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. 20 superclasses are considered in our experiments. We also transform the color image to a 3072-dimensional vector.

• **STL-10** The dataset consists of 96 × 96 color images. There are 10 classes with 1300 examples each. It also contains 100000 unlabeled images of the same resolution. We used the unlabeled set when training our autoencoder.

For a fair comparison, the training and testing samples of each dataset are jointly utilized in our experiments for all algorithms, and we set the number of clusters is the number of ground-truth categories. Similar to DEC [8], on STL-10 dataset, we concatenated HOG feature and a 8 × 8 color map to use as input to all algorithms, the remaining datasets and methods, the pixel intensities serve as inputs.

All the hyper-parameters and their values of our approach are listed in Table 2. We use TensorFlow to implement our approach. Stochastic gradient descent (SGD) with momentum is adopted in the autoencoder loss Eq. (3) minimization phase. During optimizing clustering loss Eq. (5) and adversarial loss Eq. (7), Adam stochastic optimization is adopted. In our experiments, the stacked autoencoder described in [8] is utilized in our model. For the discriminator networks $\mathbf{D}$, we utilize a three-layer fully-connect layers with dimension $K \rightarrow 2000 \rightarrow 2000 \rightarrow 1$, where the number of last layer neurons is changed to one to discriminate the input distribution is real or fake.

B. EVALUATION METRICS

In our experiments, we utilize three popular measures in the literature to evaluate the performance of clustering methods, accuracy (ACC), normalized mutual information (NMI) and adjusted rand index (ARI).

• **ACC** Accuracy is the best mapping between cluster assignments and true labels, which is defined by

$$ACC = \max_{m} \frac{\sum_{i=1}^{n} \sigma(l_i, m(c_i))}{n} \tag{9}$$

where l_i is true label of sample i, c_i is the cluster assignment produced by the algorithm, and $m(\cdot)$ ranges over all possible one-to-one mappings between clusters and labels and n means the number of samples. When $m(c_i) = l_i$, $\sigma(l_i, m(c_i)) = 1$.

• **NMI** Normalized mutual information is the normalized measure of similarity between two labels of the same data, which is defined by

$$NMI = \frac{I(l, c)}{\frac{1}{2}[H(l) + H(c)]}, \tag{10}$$

where I is the mutual information metric and H is entropy.

• **ARI** Adjusted rand index is defined by

$$ARI = \frac{\sum_{i,j=1}^{k} C_{n_{ij}}^2 - \frac{\sum_{i=1}^{k} C_{n_i^\tau}^2 \cdot \sum_{i=1}^{k} C_{n_i^r}^2}{C_n^2}}{\frac{1}{2}(\sum_{i=1}^{k} C_{n_i^\tau}^2 + \sum_{i=1}^{k} C_{n_i^r}^2) - \frac{\sum_{i=1}^{k} C_{n_i^\tau}^2 \cdot \sum_{i=1}^{k} C_{n_i^r}^2}{C_n^2}}, \tag{11}$$

where combination operation C_n^m is defined as a selection of m items from a collection n.

All above measures range in [0, 1], and higher scores imply better clustering performance.

C. COMPARISON METHODS

In the experiment, we compare the proposed model with many competitive or representative methods, including traditional methods K-Means [24], SC [3], AC [25], SEC [14] and the clustering based on NMF [26]. For deep representation based clustering approaches, we employ some unsupervised learning methods including AE [27], SAE [28], DAE [29], DeCNN [30], SWWAE [31] and DEC [8], deep subspace clustering-L2 (DSC) [32], latent distribution preserving deep subspace clustering (DPSC) [33], deep clustering with sample-assignment invariance prior (DCSAIP) [34].

D. EXPERIMENT RESULTS AND ANALYSIS

1) IMAGE CLUSTERING

Table 3 reports the clustering results, including ACC, NMI and ARI of the algorithms on the aforementioned datasets. Comparing the experimental results, we have several interesting observations as follows:

(1) For image clustering task, our model achieves the best results on all datasets except the CIFAR-10 dataset. Specifically, the ACC on the MNIST dataset increase 6.67% compared the strongest competitor DEC [8]. On the CIFAR-10 and CIFAR-100 datasets, the advantage of the ARI is not

TABLE 3. The clustering results of various methods on four datasets. The best results are highlighted in bold. ⊗ means the results are unavailable from the corresponding paper or code. The data marked with ⋆ in the upper right corner is obtained by running the code provided by the author.

Dataset	MNIST			CIFAR-10			CIFAR-100			STL-10		
Metric	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-means [24]	57.23	49.97	36.52	22.89	8.71	4.87	12.97	8.39	2.80	19.20	12.45	6.08
SC [3]	69.58	66.26	52.14	24.67	10.28	8.53	13.60	9.01	2.18	15.88	9.78	4.79
AC [25]	69.53	60.94	48.07	22.75	10.46	6.46	13.78	9.79	3.44	33.22	23.86	14.02
SEC [14]	80.40	77.90	64.07	27.35⋆	22.50⋆	8.31⋆	17.52⋆	12.60⋆	3.85⋆	30.40⋆	21.57⋆	13.10⋆
NMF [26]	54.47	60.82	42.98	18.95	8.14	3.38	11.75	7.91	2.63	18.04	9.62	4.58
AE [27]	81.23	72.57	61.39	31.35	23.93	16.89	16.45	10.04	4.76	30.30	24.96	16.10
SAE [28]	82.71	75.65	63.93	29.73	24.68	15.55	15.67	10.90	4.36	32.03	25.20	16.05
DAE [29]	83.16	75.63	64.67	29.71	25.06	16.27	15.05	11.05	4.60	30.22	22.42	15.19
DeCNN [30]	81.79	75.77	66.91	28.20	23.95	**17.36**	13.27	9.23	3.78	29.88	22.67	16.21
SWWAE [31]	82.51	73.60	65.18	28.40	23.30	16.38	14.72	10.34	3.91	27.04	19.62	13.58
DSC-L2 [32]	71.50⋆	70.40⋆	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
DEC [8]	84.30	77.16	74.14	30.10	**25.68**	16.07	18.52	13.58	**4.95**	35.90	27.60	18.61
DPSC [33]	79.70	82.30	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
DCSAIP(HOE) [34]	87.16	75.50	74.27	22.06	7.02	3.93	⊗	⊗	⊗	⊗	⊗	⊗
Ours	**90.97**	**89.55**	**86.51**	**31.40**	21.51	9.13	**22.60**	**26.93**	3.94	**38.27**	**34.28**	**22.19**

FIGURE 3. Part of clustering result on MNIST data sets. For each data set, each row represents one class.

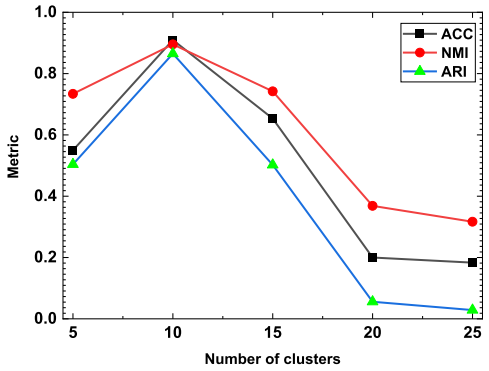

FIGURE 4. Comparison of clustering performance with changing number of clusters on MNIST dataset.

very obvious. However, the proposed method get much more improvement than the DEC, which demonstrates that the proposed method can effectively learning the latent representation hidden in visual features. Furthermore, the approaches of deep representation (such as DeCNN [30], SWWAE [31]) is dramatically outperforms the traditional methods (such as K-Means [24], AC [25]), by which we can draw a conclusion that representation learning is significant to image clustering. Additionally, the proposed RCDA is better than some deep subspace clustering models, *e.g.,* DPSC [33] and DSC [32], which demonstrate we learned better latent space. Fig. 3 shows part of the clustering result on MNIST dataset.

(2) For large-scale image datasets such as CIFAR-100 and STL-10, the proposed method is more distinct superiority than other methods. Hence, RCDA is able to handle complex and massive image clustering task.

2) PERFORMANCE ON VARIOUS NUMBER OF CLUSTERS

Fig. 4 shows the clustering results on MNIST dataset when the number of clusters various between 5 and 25 with an interval 5. In summary, as the number of clusters changes, our method generally degraded. This is because more uncertainty is triggered as the number of clusters changes. The results demonstrate that CADR possesses adequate capability to tackle various clusters.

3) SENSITIVITY ANALYSIS

We then test the sensitivity of our method w.r.t the parameter λ_{11} of the adversarial learning and parameter λ_{21} of regression term. We first analyze the sensitivity of the parameter λ_{11}. The tested range is $[0, 1.0]$. The ACC, NMI and ARI metric values on MNIST dataset of different $\lambda_{11} \in [0, 1.0]$ are shown in Fig. 5 (a), from which we can observe that our method performs stably in a wide range of λ_{11}. When we make this experiment, the parameter λ_{21} is a constant (10^{-4}). Next, we test the sensitivity of the parameter λ_{21}, in which we set $\lambda_{11} = 0.5$. Due to the fact that the regression term $\|\mathbf{F} - \mathbf{P}\|_F^2$ in Eq. (5) is a huge number, so we set $\lambda_{21} \in [10^{-11}, 10^{-1}]$ to keep the clustering loss in Eq. (5) balanced. As shown in Fig. 5 (b), our method achieves stably performance in a wide range of λ_{21}. When λ_{21} is bigger than 10^{-3}, the clustering loss cannot maintain balance between these two terms in Eq. (5), which lead to bad clustering results. The

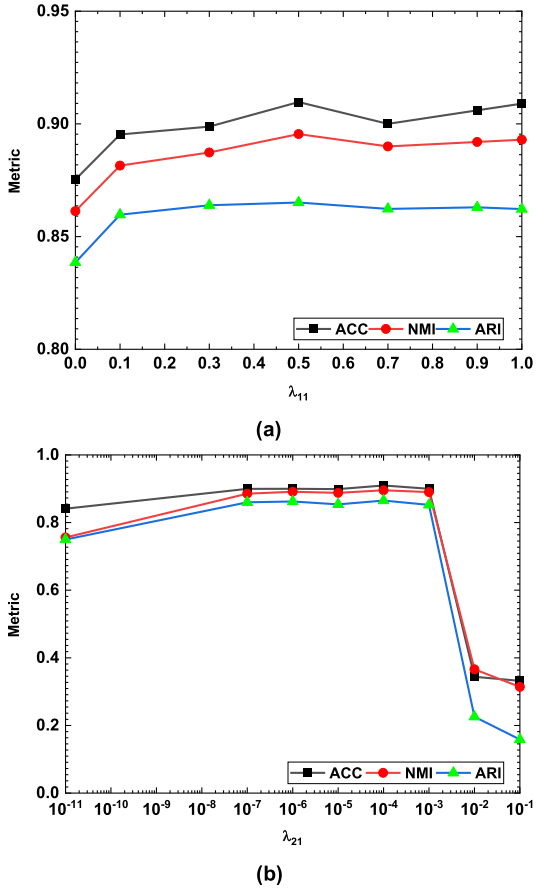

FIGURE 5. Sensitivity analysis of parameter λ_{11} and λ_{21} on MNIST dataset.

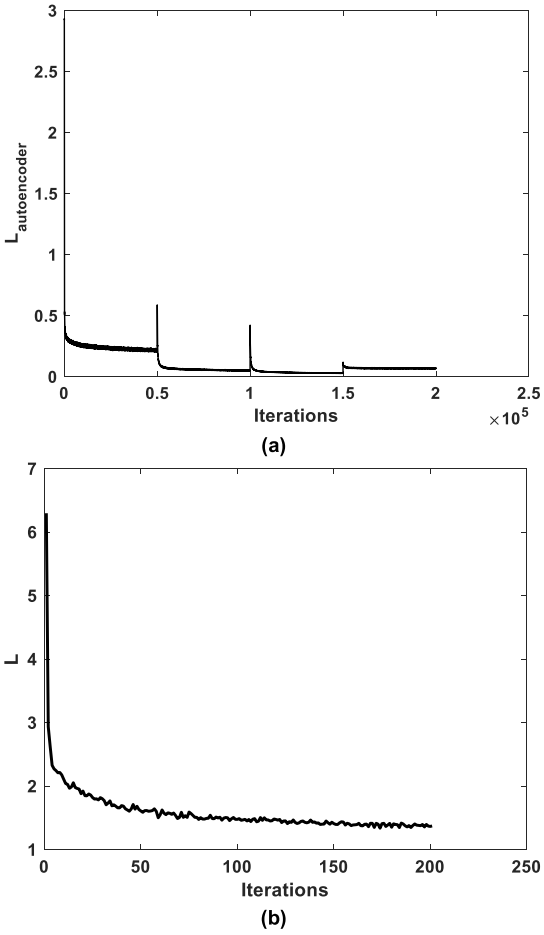

FIGURE 6. The objective convergence curves of our proposed RCDA.

default value of the parameter λ_{11} of adversarial learning and ratio λ_{21} of regression term is recommended to be set to 0.5 and 10^{-4}, respectively.

4) CONVERGENCE ANALYSIS

As shown in Fig. 6 (a), we show the objective value convergence curve of pre-training autoencoder, *i.e.,* Eq. (3), on MNIST dataset. As shown in Fig. 6 (b), we show the total objective value convergence curve of the proposed RCDA, *i.e.,* Eq. (8). As seen, the proposed method has good convergence in both the pre-training stage and the clustering stage. Especially in the clustering stage, the proposed method converges very quickly, which ensures the running speed of RCDA.

5) VISUALIZATION

For convenience, we randomly choose 5, 000 samples from MNIST dataset, and provide a t-SNE visualization of our proposed RCDA. As shown in Fig. 7 (a), we apply t-SNE on the raw sample. As shown in Fig. 7 (b), we apply t-SNE on the latent representation learned by RCDA, *i.e.,* the representation **F** obtained via Eq. (2). As can be seen, our approach exhibits a clearer and more compact cluster structure than the raw sample. This nice cluster-structured property is attributed to adversarial regression learning of our proposed RCDA.

E. DISCUSSION OF ADVERSARIAL REGRESSION

According to objective function Eq. (8), we discard the discriminator networks **D** for proving that the adversarial learning between data distribution **Q** and target distribution **P** can help to improve clustering effect via improving the performance of latent representation. Hence, the Eq. (8) can be changed as

$$L = \min_{\mathbf{E}_n, \mathbf{D}_e} \max_{\mathbf{D}} L_{AE} + \beta_1 L_C \tag{12}$$

where β_1 is the parameters of clustering loss. With similar experiment settings for four datasets, Table 4 shows the difference between the results of containing discriminator networks **D** and discarding **D**.

In Table 4, we report the clustering results of containing **P**, **Q** adversarial learning or not. Note that the performance when adding **P**, **Q** adversarial learning outperforms the methods without **P**, **Q** adversarial learning on all the three clustering quality measures. Hence, the **P**, **Q** adversarial learning in Eq. (8) is advantageous in the process of latent representation

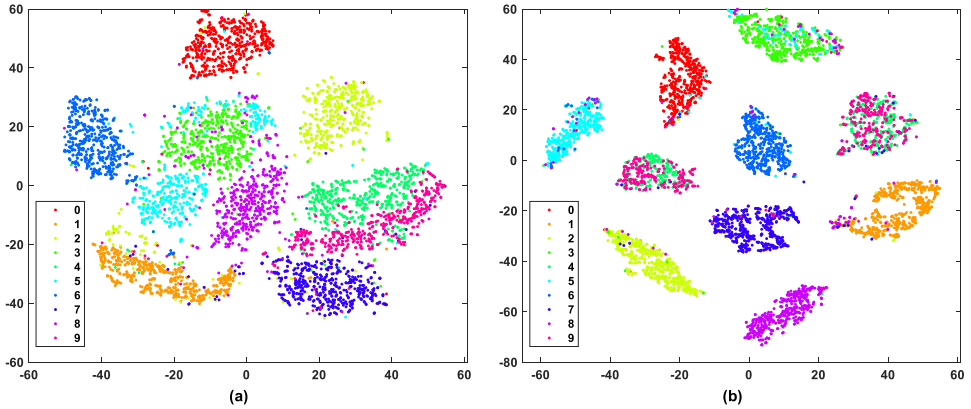

FIGURE 7. Visualization of raw sample and the latent representations learned by RCDA on MNIST dataset.

TABLE 4. The clustering results of adversarial regression analysis on four datasets, where ✗ denote the network does not contain discriminator network D and ✓ is exactly the opposite.

Dataset	MNIST			CIFAR-10		
Metric	ACC	NMI	ARI	ACC	NMI	ARI
✗	87.53	86.13	80.89	29.60	20.40	8.00
✓	**90.97**	**89.55**	**86.51**	**31.40**	**21.51**	**9.13**
Dataset	CIFAR-100			STL-10		
Metric	ACC	NMI	ARI	ACC	NMI	ARI
✗	20.47	23.61	3.59	36.70	31.52	19.36
✓	**22.60**	**26.93**	**3.94**	**38.27**	**34.28**	**22.19**

learning of our deep adversarial regression for clustering model.

V. CONCLUSION

In this paper, we propose a novel clustering model called regression based clustering by deep adversarial learning (RCDA), which jointly learns a mapping from the data space to a lower dimensional feature space and precisely predicts cluster assignments. In our method, we consider the distribution relationship between data distribution and target distribution, and utilize adversarial learning to supervise clustering. To enhance the representation ability of latent representations, we utilize a soft regression constraint as clustering loss to update learnable parameters of autoencoder. Empirical results on four widely used datasets show this new deep clustering model outperforms existing clustering methods.

REFERENCES

[1] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[2] D. Xie, W. Xia, Q. Wang, Q. Gao, and S. Xiao, "Multi-view clustering by joint manifold learning and tensor nuclear norm," *Neurocomputing*, vol. 380, pp. 105–114, Mar. 2020.
[3] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. NeurIPS*, 2005, pp. 1601–1608.
[4] D. Xie, X. Zhang, Q. Gao, J. Han, S. Xiao, and X. Gao, "Multiview clustering by joint latent representation and similarity learning," *IEEE Trans. Cybern.*, early access, Jun. 26, 2019, doi: 10.1109/TCYB.2019.2922042.
[5] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowl.-Based Syst.*, vol. 163, pp. 510–517, Jan. 2019.
[6] F. De la Torre and T. Kanade, "Discriminative cluster analysis," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 241–248.
[7] F. Tian, B. Gao, Q. Cui, E. Chen, and T. Liu, "Learning deep representations for graph clustering," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1293–1299.
[8] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.
[9] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1753–1759.
[10] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5736–5745.
[11] S. Zheng, X. Cai, C. H. Q. Ding, F. Nie, and H. Huang, "A closed form solution to multi-view low-rank regression," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 1973–1979.
[12] J. Ye, "Least squares linear discriminant analysis," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 1087–1093.
[13] J. Han, H. Liu, and F. Nie, "A local and global discriminative framework and optimization for balanced clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3059–3071, Oct. 2019.
[14] F. Nie, D. Xu, I. W. Tsang, and C. Zhang, "Spectral embedded clustering," in *Proc. IJCAI*, 2009, pp. 1181–1186.
[15] H. Gao, X. Wang, and H. Huang, "New robust clustering model for identifying cancer genome landscapes," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 151–160.
[16] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.
[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.
[18] M. T. Law, R. Urtasun, and R. S. Zemel, "Deep spectral clustering learning," in *Proc. ICML*, 2017, pp. 1985–1994.
[19] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5147–5156.
[20] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[22] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.
[23] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. IJCAI*, 2011, pp. 215–223.

[24] J. Wang, J. Wang, J. Song, X.-S. Xu, H. Tao Shen, and S. Li, "Optimized Cartesian K-Means," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 180–192, Jan. 2015.
[25] K. Chidananda Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighbourhood," *Pattern Recognit.*, vol. 10, no. 2, pp. 105–112, Jan. 1978.
[26] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality preserving nonnegative matrix factorization," in *Proc. IJCAI*, 2009, pp. 1–6.
[27] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. NeurIPS*, 2007, pp. 153–160.
[28] A. Ng, "Sparse autoencoder," *CS294A Lect. Notes*, vol. 72, pp. 1–19, 2011.
[29] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.
[30] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.
[31] J. Zhao, M. Mathieu, R. Goroshin, and Y. Lecun, "Stacked what-where auto-encoders," 2015, *arXiv:1506.02351*. [Online]. Available: https://arxiv.org/abs/1506.02351
[32] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. NeurIPS*, 2017, pp. 24–33.
[33] L. Zhou, X. Bai, D. Wang, X. Liu, J. Zhou, and E. Hancock, "Latent distribution preserving deep subspace clustering," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4440–4446.
[34] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 31, 2020, doi: 10.1109/TNNLS.2019.2958324.

FEI TANG received the master's degree in communication and information system from Shenzhen University, in 2007, and the Ph.D. degree from South China Agricultural University, in 2017. Since 2007, he has been working with the Shenzhen Institute of Information Technology. He presided over two projects of the Guangdong Natural Science Foundation and two projects of the Shenzhen Science and Technology Plan. He has published more than 20 academic articles. His research interests include intelligent computing, evolutionary algorithm, and image processing.

DABIN ZHANG received the Ph.D. degree in computer software and theory from Wuhan University, Wuhan, China, in 2007. He is currently a Professor with the College of Mathematics and Informatics, South China Agriculture University (SCAU), Guangzhou, China. He has published four books and over 80 journal articles. His research interests include big data analysis, data mining, artificial intelligence, and forecast modeling.

TIE CAI received the Ph.D. degree in circuits and systems from Shanghai Jiao Tong University, Shanghai, China, in 2006. He is currently a Professor with the School of Software Engineering, Shenzhen Institute of Information Technology (SZIIT), Shenzhen, China. His research interests include signal processing, pattern recognition, and machine learning.

QIN LI received the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong, in 2010. He is currently a Professional Teacher, a Senior Engineer, and a Shenzhen Peacock Scholar with the Shenzhen Institute of Information Technology. His current research interests include image processing, pattern recognition, and biometrics-based on mobile terminals.