Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Unsupervised deep clustering via contractive feature representation and focal loss

Jinyu Cai[a,b], Shiping Wang[a,b], Chaoyang Xu[c], Wenzhong Guo[a,b,*]

[a] College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China
[b] Network Computing and Intelligent Information Processing Laboratory, Fuzhou University, Fuzhou 350116, China
[c] School of Information Engineering, Putian University, Putian 351100, China

ARTICLE INFO

Article history:
Received 25 February 2021
Revised 9 September 2021
Accepted 20 October 2021
Available online 22 October 2021

Keywords:
Unsupervised learning
Clustering
Contractive feature representation
Focal loss
Auto-encoder

ABSTRACT

Deep clustering aims to promote clustering tasks by combining deep learning and clustering together to learn the clustering-oriented representation, and many approaches have shown their validity. However, the feature learning modules in existing methods hardly learn a discriminative representation. In addition, the label assignment mechanism becomes inefficient when dealing with some hard samples. To address these issues, a new joint optimization clustering framework is proposed through introducing the contractive representation in feature learning and utilizing focal loss in the clustering layer. The contractive penalty term added in feature learning would cause the local feature space contraction, resulting in learning more discriminative features. To our certain knowledge, this is also the first work to utilize the focal loss to improve the label assignment in deep clustering method. Moreover, the construction of the joint optimization framework enables the proposed method to learn feature representation and label assignment simultaneously in an end-to-end way. Finally, we comprehensively compare with some state-of-the-art clustering approaches on several clustering tasks to demonstrate the effectiveness of the proposed method.

1. Introduction

As a classic research field of artificial intelligence, clustering has been widely studied in recent decades, and its applications cover many aspects, such as text data analysis [1], image segmentation [2] and object detection [3], etc. The target of clustering is to divide the samples that are similar to each other into the same category, while separating different categories as much as possible. In the past few decades, numbers of classic clustering techniques have been proposed, such as *k*-means, Mean-shift and DBSCAN, etc. Although classical clustering methods have obtained a lot of achievements, their disadvantages have become increasingly apparent in time and space costs with the explosive growth of data dimensions in recent years.

To address this issue, researchers first proposed applying feature representation methods, such as principle components analysis (PCA), non-negative matrix factorization (NMF), and auto-encoder (AE) to facilitate the clustering tasks. Through mapping the original data into a low-dimensional feature space, the feature representation methods can significantly save time and space costs. For instance, Alzate et al. [4] took advantage of weighted kernel PCA to propose a new multi-way spectral clustering method, and improved the performance in terms of computation times. Zheng et al. [5] proposed an effective tumor clustering approach by using NMF and its extensions, and achieved encouraging clustering performance on several gene expression data sets. Tian et al. [6] applied a stacked auto-encoder to extract the non-linear embedded features for the initial graph, then executed the *k*-means algorithm and acquired clustering results. Dang et al. [7] improved the deep subspace clustering framework through introducing a new multi-scale fusion model and a similarity constraint model to promote the learning of a more representative self-expression coefficient matrix.

In recent years, the framework that considers both feature representation learning and clustering tasks has been widely studied, i.e., deep clustering [8–10]. Through the practical experiments of most researchers, it has been proved that deep learning can benefit clustering tasks. For example, by applying an auto-encoder to extract the embedded features of inputs and designing a clustering layer, Xie et al. [11] proposed deep embedded clustering (DEC) to improve the clustering performance. Then Guo et al. [12] discovered the drawbacks of DEC that did not consider preserving the local structure, and proposed improved deep clustering method

[*] Corresponding author.
E-mail address: guowenzhong@fzu.edu.cn (W. Guo).

https://doi.org/10.1016/j.patcog.2021.108386
0031-3203/© 2021 Elsevier Ltd. All rights reserved.

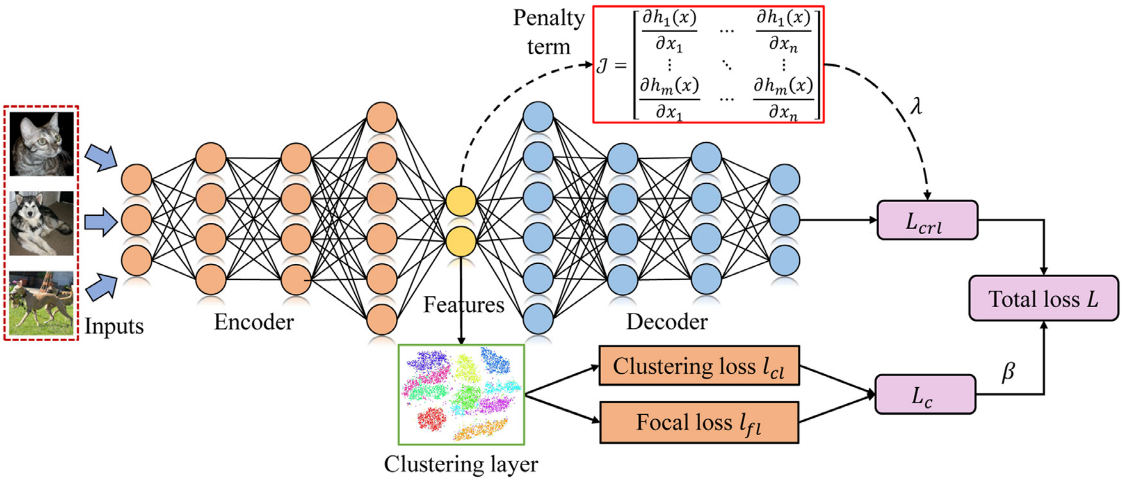

Penalty term
$\mathcal{J} = \begin{bmatrix} \frac{\partial h_1(x)}{\partial x_1} & \cdots & \frac{\partial h_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m(x)}{\partial x_1} & \cdots & \frac{\partial h_m(x)}{\partial x_n} \end{bmatrix}$
λ
Inputs
Encoder
Features
Decoder
L_{crl}
Total loss L
Clustering loss l_{cl}
Focal loss l_{fl}
L_c
β
Clustering layer

Fig. 1. Illustration of the network structure of our method. The upper part is the contractive representation learning module, whose objective is represented by L_{crl}. A Jacobian penalty term is imposed in the hidden layer, which causes the local feature space contraction, resulting in learning more discriminative features. The lower part is the clustering module, in which a soft labels distribution and a target distribution are generated to measure the objective L_c. By taking advantage of the self-training in our network to treat the target distribution in the clustering layer as the real labels distribution, the focal loss is introduced into our clustering module and employed in an unsupervised manner. The proposed method jointly optimizes these two modules to learn the clustering-oriented representation in an end-to-end way.

(IDEC) to ameliorate DEC method by keeping the decoder in network structure to prevent the distortion of feature space. Yang et al. [13] designed a dual auto-encoder network to enforce a reconstruction constraint on embedded features in the latent layer, so as to realize the joint optimization of the discriminative embedding learning and spectral clustering.

However, the existing deep clustering methods still have some challenging issues that are worth investigating. For example, the feature learning modules such as those in DEC and IDEC hardly learn a discriminative representation, and some recent studies [14–16] have proved that a discriminative feature representation can significantly promote clustering. In addition, the label assignment mechanism in existing deep clustering methods becomes inefficient when dealing with some hard samples. Specifically, we found empirically that some samples are always misclassified in the clustering layer, which we called them hard samples, and this may be the reason for limiting the further improvement of clustering performance. Therefore, how to effectively mine these hard samples is a promising issue. Inspired by the remarkable achievements of focal loss in the field of image classification and object detection [17,18], we consider the potential of focal loss on mining hard samples may help to improve the label assignment mechanism in the existing clustering framework.

In this paper, a new clustering framework namely deep clustering with contractive representation learning and focal loss (DCCF) is proposed to solve the aforementioned issues. The proposed method is a joint optimization framework that can learn the feature representation and label assignment simultaneously in an end-to-end manner. Fig. 1 illustrates its network structure. First, to learn the more effective features, we introduce the contractive representation learning [19] in our method. Specifically, a penalty term of the Frobenius norm of the Jacobian matrix is added in our representation learning module, which will cause the local feature space contraction, resulting in learning more discriminative features. Second, we adopt focal loss in our clustering module to help improve the label assignment mechanism. However, it is well known that the focal loss is commonly applied in supervised learning scenarios since it requires real labels. Therefore, the most challenge in implementation is how to embed focal loss in an unsupervised clustering framework. Faced with this issue, we take advantage of the self-training in our network to treat the target distribution in the clustering layer as the real labels distribution, and thus design a mechanism to apply focal loss in an unsupervised manner. To our certain knowledge, the proposed DCCF method is the first work to introduce the focal loss to data clustering tasks. Then, we carry out comprehensive experiments to assess the proposed method. To be specific, we compare the proposed DCCF method with some prevalent clustering methods in several clustering tasks, including the clustering on handwritten digits, real-world images and text. The experimental results indicate that DCCF has remarkable advantages over other comparative approaches and achieves state-of-the-art clustering performance. Moreover, we further discuss the feasibility of large-scale clustering, as well as the ablation study, parameter sensitivity analysis and convergence analysis of our method. The overall experiments demonstrate the effectiveness of our method.

The main contributions of our work are summed up as follows:

- Propose an end-to-end clustering framework that learns embedded representation and implements clustering simultaneously. The two components of the framework can mutually promote learning clustering-oriented representation.
- Improve the representation learning module by introducing the contractive penalty term, which forces the local feature space contraction, leading to capture more discriminative representation.
- Design a mechanism to embed focal loss into the clustering framework in an unsupervised manner by exploiting the target distribution generated in the clustering layer. This is also the first work that employs focal loss to improve the label assignment in deep clustering.
- Demonstrate the superiority of the proposed method in comparison with several state-of-the-art clustering approaches on seven publicly available data sets.

The structure of this paper is described as follows. First, in Section 2, we briefly review some related works on deep clustering and focal loss. Second, the architecture, training strategy and implementation details of the proposed method are illustrated in Section 3. Then, in Section 4, extensive experiments are conducted to evaluate the effectiveness of our method, including the comparison with other popular clustering approaches, the feasibility of large-scale clustering and some discussions of ablation study and parameter sensitivity. Eventually, we summarize the paper in Section 5.

2. Related works

In this section, we briefly present some related researches on deep clustering and focal loss, which are the foundations of our work.

2.1. Deep clustering

Deep clustering [20,21] can be partitioned into two forms, which aims to take advantage of deep learning to help improve clustering. The first form is the two-stage clustering framework, which separates the two processes of representation learning and clustering. Because of the powerful representation learning capability of deep neural networks, the learned representation can remove redundant features of the original data and map the data with high dimension into low-dimensional feature space, thereby improving the efficiency and performance of clustering tasks. Bharti et al. [22] integrated the feature extraction and feature selection to propose a new clustering method for text clustering. Zhu et al.[23] combined the subspace clustering and unsupervised feature selection to propose a new clustering framework, and the learned features can well retain the cluster labels thereby improving the clustering performance. Peng et al. [14] introduced the L2-graph to develop a new subspace clustering method which can remove the influence of the errors from representation. Yang et al. [24] introduced an effective relaxation constructed by $\ell_{2,1}$-norm distance to improve the graph cut clustering algorithm, which can yield the more sparse representation to maintain a clearer clustering structure.

Another form of deep clustering is the combination of representation learning and clustering, whose target is to allow the network to learn a cluster-oriented representation. Xie et al. proposed the deep embedded clustering approach [11] to accomplish feature representation learning and clustering tasks simultaneously. Xu et al. [25] proposed a new joint optimization multi-view clustering method to overcome the drawback that the high dimensionality of each view of data. Yang et al. [26] developed an effective adversarial attack method to apply adversarial learning into different types of deep clustering models, thereby improving their robustness. Dang et al. [27] proposed to achieve the matching of samples and their nearest neighbors from two perspectives, i.e., local and global, thus improving the clustering performance by exploiting both local and global features. Based on empirical evidence, compared to the two-stage clustering methods, the joint optimization framework is more conducive to the clustering tasks due to its specifically set loss term for clustering.

2.2. Focal loss

Focal loss [17,28] was proposed to address the problem of serious imbalance between the positive and negative sample ratios in one-stage target detection, and can also be understood as a hard sample mining approach. For binary classification case, we let p and y denotes the predicted label and true label respectively, and define p_t as follows:

$$p_t = \begin{cases} p & \text{if } y = 1, \\ 1 - p & \text{otherwise,} \end{cases} \tag{1}$$

then the cross entropy loss (CEloss) can be written as $\text{CEloss}(p, y) = \text{CEloss}(p_t) = -\log(p_t)$. The idea of focal loss is to add a modulating factor on the basis of CEloss, and it could be formalized as follows:

$$FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t), \tag{2}$$

where $(1 - p_t)^{\gamma}$ indicates the modulating factor, and γ is a tunable focusing parameter that satisfies $\gamma \geq 0$.

In recent years, focal loss has been widely applied in the fields of object detection and image classification. Lin et al. [17] utilized the focal loss to train a dense detector, namely RetinaNet, which achieved encouraging detection accuracy. Shu et al. [18] solved the issue of the category imbalance during training by applying focal loss in their proposed neural network, and realized higher performance on the task of breast cancer classification. In view of the success of focal loss in the above fields, we believe that the potential of focal loss in clustering tasks is also worthy of investigation and discussion, which is also the main motivation of our work.

3. Proposed method

In this section, the framework of the proposed DCCF method is first introduced. Specifically, we describe in detail the contractive representation learning module, the construction of clustering layer and how they jointly optimize. Thereafter, the implementation details of the proposed method will be presented.

3.1. Contractive representation learning module

Define $\mathbf{X} \in \mathbb{R}^{d \times n}$ as the input data set, where n indicates the number of samples and d denotes the dimension of data point. The objective of our method is to divide each sample $\mathbf{x}_i \in \mathbf{X}$ into the correct cluster as much as possible without the ground-truth label information. For the purpose, we design a new deep clustering framework to execute clustering in an end-to-end joint optimization way. To be specific, the framework is composed of two main modules. One is the feature learning module which utilizes the contractive representation learning to obtain more discriminative features, and the other module is an improved clustering layer which introduces the focal loss in data clustering to mine the hard samples.

The basic idea of feature learning in deep clustering is to apply an auto-encoder to capture the embedded representation of original data. The typical auto-encoder consists of an encoder and a decoder. By using non-linear transformation, the encoder projects the original data $\mathbf{X}$ into a embedded representation $\mathbf{H} \in \mathbb{R}^{d' \times n}$ with d' dimensions, which is the so-called embedded feature. Since the embedded feature $\mathbf{h}_i \in \mathbf{H}$ corresponds to data point $\mathbf{x}_i \in \mathbf{X}$, the encoder can be formulated as follows:

$$\mathbf{h}_i = f(\mathbf{x}_i | \mathbf{W}_e, b_e) = f(\mathbf{W}_e \mathbf{x}_i + b_e), \tag{3}$$

where $f(\cdot)$ denotes the activation function of encoder, $\mathbf{W}_e \in \mathbb{R}^{d' \times d}$ and $b_e \in \mathbb{R}^{d'}$ are the weight matrix and bias of encoder network respectively. Similarly, the decoder aims to project the embedded feature $\mathbf{H}$ back to the reconstructed data $\mathbf{X}' \in \mathbb{R}^{d \times n}$ as the following formula:

$$\mathbf{x}_i' = g(\mathbf{h}_i | \mathbf{W}_d, b_d) = g(\mathbf{W}_d \mathbf{h}_i + b_d), \tag{4}$$

where $g(\cdot)$, $\mathbf{W}_d \in \mathbb{R}^{d \times d'}$ and $b_d \in \mathbb{R}^{d}$ are defined similarly to the encoder. Therefore, the objective of auto-encoder is to find a mapping to minimize the reconstruction error as follows:

$$\min_{\Omega} \left\| \mathbf{X} - \mathbf{X}' \right\|_F^2, \tag{5}$$

where $\Omega = (\mathbf{W}_e, \mathbf{W}_d, b_e, b_d)$ denotes the parameters of auto-encoder network.

However, it will lead to the perturbations of intermediate representation if the training only guided by the reconstruction error in feature learning. To capture the more effective feature representation, we propose to introduce the Frobenius norm of Jacobian matrix $J(\mathbf{x})$ as a penalty term in feature learning. Fig. 2 shows the architecture of the contractive representation learning module. Specifically, the penalty term $J(\mathbf{x})$ aims to penalize the sensitivity

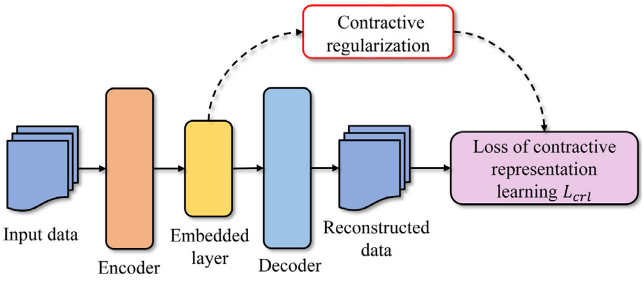

Contractive regularization
Loss of contractive representation learning L_{crl}
Input data
Encoder
Embedded layer
Decoder
Reconstructed data

Fig. 2. Structure of the contractive representation learning module. A contractive regularization term is imposed on the embedded layer, which can produce discriminative embedded features for our clustering module.

to input data, which can be defined as follows:

$$\|J(\mathbf{x})\|_F^2 = \sum_{i,j}\left(\frac{\partial \mathbf{h}_j(\mathbf{x})}{\partial \mathbf{x}_i}\right)^2. \tag{6}$$

For the embedded features relative to the dimension of inputs, $\|J(\mathbf{x})\|_F^2$ represents the summary of squares of all their partial derivatives. Considering that the flatness caused by the first derivative with low-value indicates the robustness or invariance of the feature representation for the small alterations of inputs, penalizing the term $\|J(\mathbf{x})\|_F^2$ can help the mapping in representation learning to be contractive near to the input data space. Consequently, the loss function of the contractive representation learning module can be formalized as follows:

$$L_{crl} = \sum_{\mathbf{x}\in\mathbf{X}} \left\|\mathbf{x}-\mathbf{x}'\right\|_F^2 + \lambda\|J(\mathbf{x})\|_F^2, \tag{7}$$

where λ is the parameter that controls the degree of contractive regularization. The contractive representation learning module will result in the contraction of local feature space in turn produce discriminative embedded features for our clustering module.

3.2. Clustering modules

An essential part of deep clustering method is the clustering layer, and the embedded feature $\mathbf{H}$ obtained from the contractive feature learning is used as its input. The structure of our clustering module is presented in Fig. 3, which will be described in detail below.

3.2.1. Clustering loss construction

Clustering aims to accurately divide a given set of data points into a specified number of categories. With the help of contractive feature learning and defined clustering layer, the effectiveness of the division can be improved. In the clustering layer, we use the KL-divergence between soft labels distribution S and target distribution T to define the clustering loss, as shown below:

$$l_{cl} = KL(T||S) = \sum_i\sum_j t_{ij}\log\frac{t_{ij}}{s_{ij}}. \tag{8}$$

The soft labels distribution S that calculate the resemblance of a representation $\mathbf{h}_i$ in the embedded layer and a cluster centroid μ_j is defined by utilizing Student's t-distribution as follows:

$$s_{ij} = \frac{(1+\left\|\mathbf{h}_i-\mu_j\right\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_j(1+\left\|\mathbf{h}_i-\mu_j\right\|^2/\alpha)^{-\frac{\alpha+1}{2}}}, \tag{9}$$

where the parameter α controls the extent of freedom of Student's t-distribution. Specifically, α is fixed to 1. Therefore, s_{ij} represents the probability of allocating example i to cluster centroid j, called soft label assignment. And the target distribution T is formalized as:

$$t_{ij} = \frac{s_{ij}^2/\sum_i s_{ij}}{\sum_j s_{ij}^2/\sum_i s_{ij}}, \tag{10}$$

where t_{ij} is defined by s_{ij}, and the intent of optimization is to match S with T, thus we can regard it as a kind of self-training.

3.2.2. Clustering with focal loss

In practical experience, we have found that there are some hard, misclassified samples if the clustering layer only guided by Eq. (8). To address this issue, we proposed to introduce focal loss to improve the label assignment mechanism in our clustering module.

Since our clustering task is usually in a multi-class case, we need to modify Eq. (2) in Section 2.2 to adapt it to this demand. Specifically, we redefine p_t as follows:

$$p_t = y_{pred} * y_{true}, \tag{11}$$

where y_{pred} and y_{true} are the predicted label and true label in the multi-class case, respectively.

In general, true labels are required in this situation, but this turns to be a supervised learning, which is inconsistent with our clustering tasks. Fortunately, as mentioned in Section 3.2.1, the soft labels distribution S and target distribution T produced in our clustering module provide us with a direct solution to this problem. To be specific, the predicted labels are generated from S and the true labels are generated from T, thus we can straightly impose the focal loss in the clustering layer during its self-training, and the objective can be formalized as follows:

$$l_{fl} = -(1-p_t)^\gamma\log(p_t). \tag{12}$$

According to the definition of l_{fl}, we can know that for an easy sample, the probability $(1-p_t)$ is small and p_t is close to 1. For a hard sample, the probability $(1-p_t)$ is large and p_t is close to 0, which means the focal loss can increase the contribution of hard samples in loss function, while reducing the contribution of easy samples in loss function. Therefore, the objective function L_c of our clustering module can be formalized as:

$$L_c = l_{cl} + l_{fl}, \tag{13}$$

where l_{cl} enables the network to realize clustering in a self-training manner, and l_{fl} attempts to address the issue that some hard samples is difficult to mine when only l_{cl} is used in the clustering module.

3.3. Training strategy

Through the contractive representation learning module and clustering module defined above, the objective function of the joint optimization network can be formalized as follows:

$$L = L_{crl} + \beta L_c, \tag{14}$$

where β controls the contribution of the clustering layer loss to the total loss.

During the training progress, we jointly optimize the parameters of contractive auto-encoder network and the cluster centroid μ_j by utilizing stochastic gradient descent (SGD) and back-propagation (BP). Specifically, let ρ denotes the learning rate and m represents the number of samples in a mini-batch, the cluster centroid μ_j is updated as follows:

$$\mu_j = \mu_j - \frac{\rho}{m}\sum_{i=1}^m\frac{\partial L_c}{\partial \mu_j}, \tag{15}$$

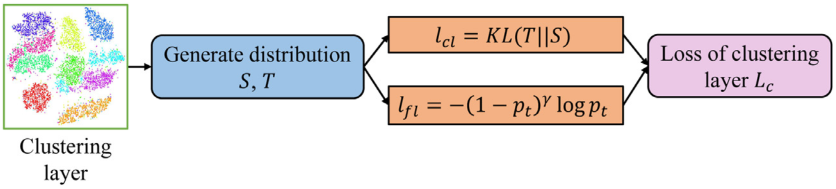

Clustering layer
Generate distribution S, T
$l_{cl} = KL(T||S)$
$l_{fl} = -(1-p_t)^{\gamma} \log p_t$
Loss of clustering layer L_c

Fig. 3. Illustration of our clustering module. The learned embedded representation in contractive feature learning are used as the input, resulting in the soft labels distribution S and target distribution T. l_{cl} represents the KL-divergence between S and T, known as clustering loss. l_{fl} indicates the focal loss, where S and T are used to generated the predicted labels and true labels respectively, which allows us to utilize focal loss in our module in an unsupervised manner.

and the weights $\mathbf{W}_e$ and $\mathbf{W}_d$ of our contractive feature learning module are updated by:

$$\mathbf{W}_d = \mathbf{W}_d - \frac{\rho}{m}\sum_{i=1}^{m}\frac{\partial L_{crl}}{\partial \mathbf{W}_d}, \tag{16}$$

$$\mathbf{W}_e = \mathbf{W}_e - \frac{\rho}{m}\sum_{i=1}^{m}(\frac{\partial L_{crl}}{\partial \mathbf{W}_e} + \beta\frac{\partial L_c}{\partial \mathbf{W}_e}). \tag{17}$$

Furthermore, the target distribution is regarded as true label in our clustering module, which is relied on the soft labels distribution and update by Eq. (10). Consequently, the label c_i is assigned to sample $\mathbf{x}_i$ according to the following formula when updating the target distribution:

$$c_i = \arg\max_j s_{ij}, \tag{18}$$

and the training procedure of the proposed method is shown at Algorithm 1.

Algorithm 1 Algorithm of DCCF.

Input: Original data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, pre-training weights, number of clusters K, parameter β, interval for update O and stopping threshold δ.

Output: The embedded representation $\mathbf{H}$ and the predicted cluster labels C.

```
Initialize the set of cluster centroid $\mu$, and import the pre-trained encoder weights $\mathbf{W}_e$ and decoder weights $\mathbf{W}_d$ to initialize the network;
while not converged do
    if $epoch\%O == 0$ then
        Extract the embedded feature $\mathbf{H}$ from the contractive representation learning module;
        Update target distribution $T$ in the clustering module through the learned $\mathbf{H}$ and Equations 9,~10;
        Save the current label assignment $C$ as $C_{old}$, and compute the new label assignment $C$ by Equation 18;
    end if
    Select $m$ samples from the input data matrix $\mathbf{X}$ to form a batch $\mathbf{X}'$;
    Calculate the loss of contractive representation learning $L_{crl}$ and clustering netowrk $L_c$;
    Update the cluster centroid $\mu$, decoder weight $\mathbf{W}_d$ and encoder weight $\mathbf{W}_e$ by Equations 15,~16 and~17, respectively.
end while
return The predicted cluster labels $C$.
```

The proposed DCCF method consists of two modules, i.e, the contractive representation learning module and clustering module. Therefore, the complexity of DCCF is $O(nD_m^2 + nd'K)$, where D_m indicates the maximum size of the hidden layers, n denotes the number of data, d' represents the dimensions of the embedded layer connected to the clustering module, and K is the number of clusters. Furthermore, as we have the condition $K \le d' \le D_m$ holds, the complexity of DCCF can be simplified to $O(nD_m^2)$. This also exhibits that DCCF does not have a high computational complexity.

3.4. Implementation details

To set up our experiment, we first pre-train a stacked auto-encoder network with d-500-500-1000-d'-1000-500-500-d network structure, where d' indicates the size of embedded layer, and the pre-training epochs are fixed as 100. After that, we use the pre-trained weights to initialize our model and start training. The learning rate ρ is set to 0.001, the parameter λ in contractive learning is set to 10^{-7}, the parameter γ in focal loss is fixed as 2, and the training epochs are set to 200.

Note that the target distribution T only needs to be updated after every O epochs to prevent unstable training. And we define a threshold δ to represent the variety in the label allocation among two consecutive target distribution updates, the training will stop when $\delta \le 0.1\%$.

4. Experiments analysis

In this section, we describe the information of the databases used in experiments, the parameter settings and the evaluation metrics at first. Then comprehensive experiments are performed to evaluate the effectiveness of our method.

4.1. Data sets

1. **MNIST**. MNIST is a handwritten digital image data set with 10 classes from (0–9),[1] which consists of 60,000 samples for training and 10,000 samples for testing. Each image is reshaped as a 784-dimensional vector.
2. **Fashion-MNIST**. Fashion-MNIST database consists of 70,000 gray-scale images, which contains 10 categories in total (such as Coat, Sneaker, Trouser, etc.).[2] The division of training set and test set in Fashion-MNIST is consistent with MNIST, and each sample is represented by a 784-dimensional feature vector.
3. **USPS**. USPS data set contains a total of 9298 pieces of handwritten digital images in 10 different categories from 0 to 9,[3] of which the training set contains 7291 images, and the rest are used as test set. The resolution of each sample is 16 × 16.
4. **STL-10**. STL-10 data set is an real-world image data set that contains 10 different classes (such as car, dog, bird, airplane, etc.).[4] There are 1300 images in each category in this data set, and the size of each image is 96 × 96 × 3.
5. **CIFAR-10**. CIFAR-10 data set contains 60,000 color image samples from the real world (such as bird, horse, ship, etc.), of which 50,000 samples are used for training and the remaining

[1] http://yann.lecun.com/exdb/mnist/
[2] https://github.com/zalandoresearch/fashion-mnist
[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html
[4] https://cs.stanford.edu/~acoates/stl10/

Table 1
A brief description of the data sets used in our experiments.

ID	data set	# instances	# classes	# original size
1	MNIST	70,000	10	28 × 28
2	Fashion-MNIST	70,000	10	28 × 28
3	USPS	9298	10	16 × 16
4	STL-10	13,000	10	96 × 96 × 3
5	CIFAR-10	60,000	10	32 × 32 × 3
6	CIFAR-100	60,000	100	32 × 32 × 3
7	Reuters-10K	10,000	4	2000

Fig. 4. Sample images of each image data set used in experiment. Note that for each data set we randomly selected 10 sample, and from the top row to the bottom row are: (a) MNIST, (b) Fashion-MNIST, (c) USPS, (d) STL-10, (e) CIFAR-10, (f) CIFAR-100.

10,000 samples for testing.[5] Each image in this data set is with the size of 32 × 32 × 3.

6. **CIFAR-100**. The size and the number of samples in CIFAR-100 data set is similar to CIFAR-10 database. It consists of 100 different categories in total and each category has 500 training samples and 100 test samples. Beside, these 100 categories are divided into 20 super categories. Each image has a “fine” label (the category it belongs to) and a “coarse” label (the super category it belongs to).
7. **Reuters-10K**. Reuters is a text database that contains 804,414 English news stories for text categorization. [6] Reuters-10K as a subset of it, consists of 10,000 English text documents in 4 root categories (government/social, economics, corporate/industrial, and markets). Each document in this database is denoted as a TF-IDF vector, which contains 2000 most frequent words.

Table 1 shows the description of several attributes of each data set. Furthermore, we also provide brief visualization by randomly selecting 10 samples for each image data set, as shown in Fig. 4.

4.2. Experimental settings

To assess the effectiveness of our method, comparative experiment is carried out. Specifically, we compare the proposed DCCF method with several clustering algorithms. These algorithms could be partitioned into three categories, the classical clustering algorithm k-means, the two-stage clustering methods by applying feature learning, including locality preserving nonnegative matrix factorization (NMF-LP) [29], auto-encoder (AE) [30], label consistent auto-encoder (LCAE) [31], stacked what-where auto-encoders (SWWAE) [32], and the jointly optimized deep clustering methods including deep embedded clustering (DEC) [11], improved deep embedded clustering (IDEC) [12], variational deep embedding (VaDE) [33], joint unsupervised learning (JULE) [34], semi-supervised deep embedded clustering (SDEC) [35], the HOE model of deep clustering with sample-assignment invariance prior (DCSAIP) [36], ClusterGAN [37], k-autoencoder deep clustering (k-DAE) [38], VaGAN-GMM [39], invariant information clustering (IIC) [40], and Gaussian attention clustering (GATCluster) [41].

In particular, k-means is used as the baseline in our experiments. For AE, LCAE, DEC and IDEC, we run the publicly released codes and report their clustering performance, and for other comparative methods we report their performance directly from the related papers. Note that for the three RGB data sets: STL-10, CIFAR10 and CIFAR100, we use a 50-layer deep residual network for pre-processing before training, and obtain the extracted features with a size of 2048 dimensions in the average pooling layer. Regarding the training of each data set, the learning rate ρ is fixed as 0.001, the dimension of the hidden layer is set to the number of classes k of the data set. The number of training epochs is set to 200 for each comparative method, and the network structure of DEC and IDEC are set to d-500-500-2,000-k-2,000-500-500-d according to the original papers, where d is the dimension of original data. All other parameters follow the default settings.

4.3. Evaluation metrics

In this section, two typical clustering evaluation metrics are selected to assess the quality of each comparative algorithm, including clustering accuracy and normalized mutual information.

The clustering accuracy (ACC) is used to compare the predicted labels with the true labels to measure the clustering performance. Assume $\mathbf{p}$ and $\mathbf{y}$ denote the predicted labels and the true labels respectively, ACC can be formalized as follows:

$$ACC = \frac{\sum_{i=1}^{n} \delta(\mathbf{y}_i, map(\mathbf{p}_i))}{n}, \tag{19}$$

where n indicates the number of samples, and $map(\cdot)$ represents a one-to-one mapping that covers every possible mapping between the predicted labels produced through clustering and the true labels. Besides, $\delta(x, y) = 1$ only when $x = y$.

The normalized mutual information (NMI) is based on the mutual information (MI). Assume A and B as two discrete random variables, MI can be formalized as:

$$MI(A, B) = H(A) + H(B) - H(A, B), \tag{20}$$

where $H(A)$ and $H(B)$ indicate the information entropy of the corresponding variables, and $H(A, B)$ represents the joint information entropy of two variables. Further, NMI can be formalized as follows:

$$NMI(A, B) = \frac{MI(A, B)}{(H(A) + H(B))/2}, \tag{21}$$

where NMI is normalized by MI, and its value range is [0,1]. The higher value of NMI denotes the closer relationship between A and B, as well as the better clustering performance.

4.4. Experimental results

In this section, we carry out comprehensive experiments to demonstrate the effectiveness of the proposed DCCF method, including the comparison with several popular clustering approaches and the feasibility analysis of large-scale clustering.

4.4.1. Comparison with several clustering approaches

The proposed DCCF method is compared with several popular clustering approaches on six publicly available data sets, and the results are presented in Tables 2 and 3, from which we can draw the following conclusions.

[5] http://www.cs.toronto.edu/~kriz/cifar.html

[6] http://www.research.att.com/~lewis/reuters21578.html

Table 2
Clustering accuracy of different clustering approaches on each tested data set. The results marked in **bold** represents the best clustering performance. Note that "–" indicates that the source code and score of these approaches were not available in the corresponding database.

Method/Data set	MNIST	Fashion-MNIST	USPS	STL-10	CIFAR-10	Reuters-10K
k-means	57.62	56.34	67.84	28.35	22.19	57.88
NMF-LP	47.10	43.40	65.20	–	17.97	66.48
AE	78.53	56.72	68.03	34.83	21.63	59.76
LCAE	58.57	56.31	70.57	30.02	24.02	61.78
SWWAE	82.51	–	–	27.04	28.40	72.84
DEC	84.46	58.69	76.56	36.12	22.37	61.85
IDEC	84.92	59.23	77.14	37.80	23.49	68.43
VaDE	94.50	55.20	56.60	–	15.60	72.30
JULE	96.40	56.30	**95.00**	27.69	27.15	62.64
SDEC	86.11	–	76.39	38.86	27.26	69.37
DCSAIP	87.16	–	–	–	22.06	69.81
ClusterGAN	95.00	63.00	70.00	–	28.12	79.46
k-DAE	87.82	59.34	76.89	33.56	23.31	71.19
VaGAN-GMM	95.48	**63.84**	–	–	28.79	80.12
DCCF	**97.41**	62.12	85.53	**72.78**	**45.81**	**83.36**

Table 3
Normalized mutual information of different clustering approaches on each tested data set. The results marked in **bold** represents the best clustering performance. Note that "–" indicates that the source code and score of these approaches were not available in the corresponding database.

Method/Data set	MNIST	Fashion-MNIST	USPS	STL-10	CIFAR-10	Reuters-10K
k-means	55.43	52.57	60.39	23.48	8.24	29.59
NMF-LP	45.20	42.50	69.30	–	5.10	34.40
AE	74.90	55.35	62.26	30.08	6.71	32.36
LCAE	48.46	55.10	61.70	27.31	8.80	32.92
SWWAE	73.60	–	–	19.62	23.30	38.05
DEC	80.91	59.09	78.37	31.82	9.62	31.46
IDEC	82.37	60.42	79.15	32.46	10.38	35.15
VaDE	87.60	57.30	51.20	–	3.60	41.60
JULE	91.30	60.80	**91.30**	18.15	19.23	40.54
SDEC	82.89	–	77.68	32.84	17.20	47.62
DCSAIP	75.50	–	–	–	7.02	34.33
ClusterGAN	89.00	64.00	67.90	–	15.60	55.30
k-DAE	85.71	64.17	79.76	24.55	11.79	44.77
VaGAN-GMM	91.70	63.30	–	–	15.80	53.60
DCCF	**93.32**	**64.58**	82.51	**66.84**	**36.19**	**55.52**

First, it is obvious that an appropriate representation learning method is beneficial to clustering tasks. We can see from these tables that the performance of two-stage clustering and deep clustering methods exceed classical *k*-means method with a large margin, which indicates the power of representation learning. At the same time, deep clustering methods outperform the two-stage clustering methods in most case, because the joint optimization in deep clustering forces the network to learn a cluster-oriented representation.

Second, the proposed DCCF method outperforms other comparative approaches on handwritten digital image data sets. Especially on the MNIST database, the DCCF method obtains the state-of-the-art performance, its 97.41% clustering accuracy exceeds the second best method JULE by 1.01%, and is 39.79% higher than the *k*-means. Meanwhile, we can notice that the clustering methods based on the generative model also show their advantages when handling digital image databases. For example, VaDE, ClusterGAN, and VaGAN-GMM obtain encouraging clustering performance on MNIST, and VaGAN-GMM and ClusterGAN obtain the best two results in terms of ACC on the Fashion-MNIST data set. Nevertheless, it is noteworthy that in comparison with them, DCCF still obtains competitive performance, as it achieves the best results in terms of NMI on the Fashion-MNIST database, and is also the runner-up on the USPS database.

Third, DCCF also exhibits its effectiveness on the real-world data sets such as STL-10 and CIFAR-10. For example, the performance gap between DCCF and the second best method SDEC on the STL-10 database is 33.92% and 34% regarding the ACC and NMI metrics. In addition to image data sets, our method also shows the advantages and potential in processing text data. For instance, on the classic text data set Reuters-10K, the ACC and NMI of DCCF are 83.36% and 55.52%, which outperforms the other comparative methods with a large margin. On the whole, the effectiveness of the proposed DCCF method is demonstrated through comprehensive experiments. By comparing with other clustering methods that apply different feature learning methods, we demonstrate the efficiency of utilizing contractive representation learning to extract features. The comparison between the proposed DCCF method and other deep clustering methods also proves that the introduced focal loss in the clustering layer can improve the label assignment mechanism.

4.4.2. *Feasibility of large-scale clustering*

The potential of clustering on the large-scale data sets is also a measure for judging the quality of a clustering algorithm. In order to demonstrate the feasibility of our method on large-scale data set, we introduce the CIFAR-100 data set in our experiment. CIFAR-100 database is a popular large-scale data set, which consists of

Table 4
Clustering performance of the CIFAR-100 data set. Note that the baseline indicates the performance of *k*-means.

Method	Baseline	AE	SWWAE	JULE	DEC	IIC	GATCluster	DCCF
ACC	13.15	16.45	14.72	13.67	18.81	25.70	28.10	**28.73**
NMI	8.47	10.04	10.34	10.26	13.79	22.50	21.50	**27.62**

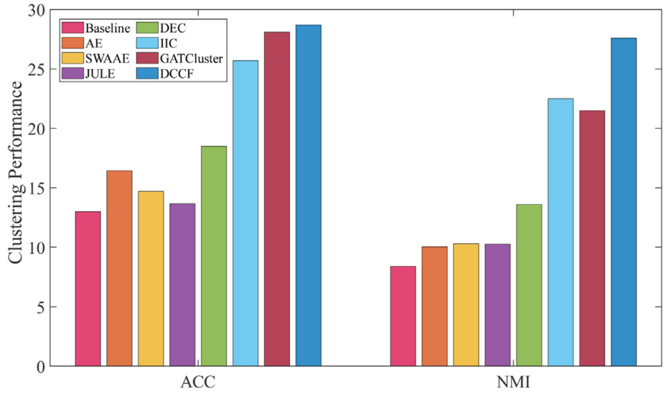


Fig. 5. The histogram for intuitive comparison of the clustering performance of different methods on the CIFAR-100 data set.

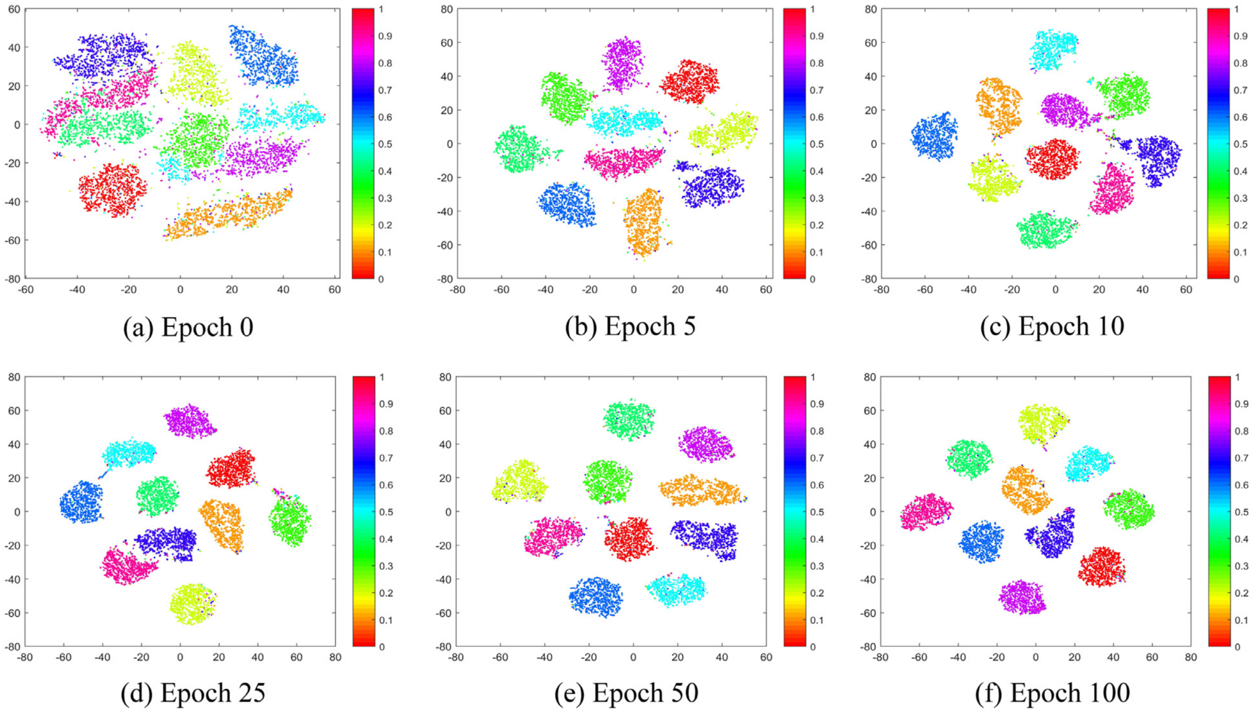
60,000 samples from 100 different classes. In Table 4 and Fig. 5, we show the comparison of clustering performance between DCCF and the other seven clustering methods, and the detail of data processing refers to Section 4.2.

Through the comparison, it can be noticed intuitively that deep clustering promotes clustering on large-scale data set. For example, DEC obtain 18.81% on ACC metric, which is 5.66% higher than *k*-means. Furthermore, our proposed DCCF method also obtain encouraging clustering performance on large-scale data set. The ACC and NMI of DCCF are 9.92% and 13.83% higher than DEC, respectively. Additionally, DCCF also achieves competitive clustering performance compared to the state-of-the-art approaches such as IIC and GATCluster. Specifically, the gaps between DCCF and them in terms of ACC are 3.03% and 0.63%, while the gaps with regard to NMI are 5.12% and 6.12%. This fully indicates the potential and feasibility of the proposed method in large-scale clustering.

4.5. *Experimental visualization*

In order to show the feature learning process and clustering performance in an intuitive way, we conduct a experimental visualization on MNIST data set. Specifically, we randomly select 10,000 samples from the learned representation in the hidden layer and apply t-SNE method to further reduce the representation into 2-dimensional features. Then we provide a 2D visualization from different epochs $\{0, 5, 10, 25, 50, 100\}$ in Fig. 6.

From this figure we can observe that most samples are mixed together and are not clearly distinguished in the epoch 0. However, as the algorithm iterates, the samples are gradually divided. We can also find that in the epoch $\{5, 10, 25, 50, 100\}$, the distance between the same classes is constantly shrinking, and the distance between different classes is constantly increasing, which indicates the better clustering performance. Especially in the epoch 100, the samples can be clearly distinguished, and there are almost no discrete points, which also prove the utility of the contractive feature learning and the applied focal loss in the clustering layer.

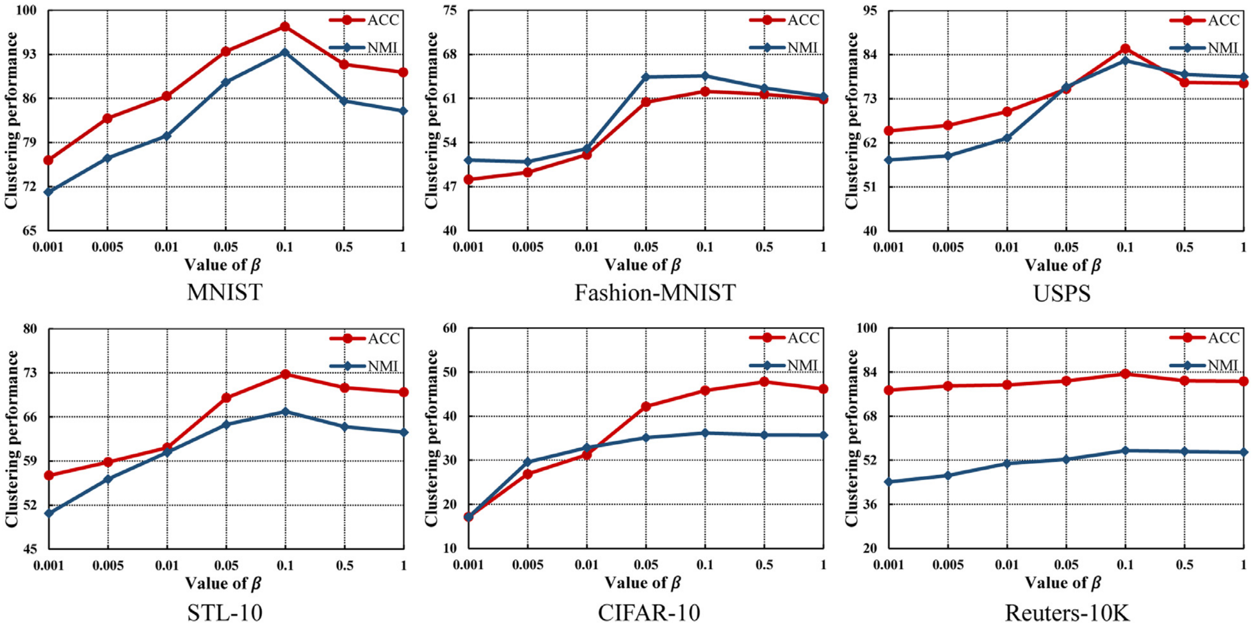


(a) Epoch 0
(b) Epoch 5
(c) Epoch 10
(d) Epoch 25
(e) Epoch 50
(f) Epoch 100

Fig. 6. The t-SNE visualization of the learned representation on MNIST data set. Note that the visualization is generated by randomly selecting 10,000 samples from the learned representation in the hidden layer.

Table 5
Clustering performance of DCCF and three degradation models. The best performance is marked in **bold**.

Contractive penalty term	Focal loss	MNIST		USPS		Reuters-10K	
		ACC	NMI	ACC	NMI	ACC	NMI
×	×	84.79	82.30	77.16	79.29	68.65	35.24
✓	×	94.30	88.65	77.33	79.67	77.34	47.93
×	✓	91.59	87.82	80.44	80.75	78.64	50.19
✓	✓	**97.41**	**93.32**	**85.53**	**82.51**	**83.36**	**55.52**



MNIST

Fashion-MNIST

USPS

STL-10

CIFAR-10

Reuters-10K

Fig. 7. The influence of different values of parameter β to the clustering performance. Note that we report both ACC and NMI at the same time, and the value range of β is $\{0.001, 0.005, \ldots, 0.5, 1\}$.

4.6. Ablation study

To validate the effectiveness of the introduced contractive penalty term and focal loss in our method, we further conduct an ablation study on three databases, including MNIST, USPS and Reuters-10K. Specifically, we construct three degradation models for DCCF by imposing only one particular loss term, i.e., contractive penalty term and focal loss, as well as by not imposing either of them. The experimental results are shown in Table 5, from which the following observations can be drawn.

First, we can see that adding either contractive penalty term or focal loss is beneficial for the clustering tasks, especially on the MNIST and Reuters-10K databases, where significant improvements can be seen. It seems that the benefit brought by adding the contractive penalty term is relatively modest on the USPS database, although there is still a small improvement. Second, applying focal loss on the USPS and Reuters-10K databases leads to more significant improvements in clustering than applying the contractive penalty term, while the opposite is observed on the MNIST database. This demonstrates the respective advantages of the two loss terms when dealing with different clustering tasks. Third, it is obvious to see that the simultaneous adoption of two loss terms can yield further improvements in clustering performance, which indicates that they are mutually reinforcing. In other words, the contractive penalty term enables the algorithm to learn a more representative features, while the focal loss further improves the assignment of clusters.

4.7. Parameter sensitivity

In this section, the sensitivity analysis experiment on two parameters in the DCCF method is carried out, including the parameter β in the loss function and the cluster numbers K.

4.7.1. Influence of the contribution of clustering layer loss

As mentioned in Section 3.3, the parameter β controls the contribution of the loss of clustering layer to the total loss. Therefore, it is necessary to evaluate the influence of the different values of β on the clustering performance of our method. Specifically, we set the variation range of the parameter β in $\{0.001, 0.005, \ldots, 0.5, 1\}$, and display the clustering results under each value of β in Fig. 7.

It can be clearly seen from the figure that in most data sets, the performance of DCCF conducted with small value of β is obviously lower than that of DCCF performed with higher value of β. For instance, in the MNIST data set, when β is 0.1, the ACC is more than 20% higher than when β is 0.001, which demonstrates that the clustering module in our method can effectively promote clustering tasks. In addition, the bigger value of β does not necessarily mean that it is better. In our practical analysis, the best clustering results in the most data sets are obtained when β is 0.1. Although the clustering performance will fluctuate due to the changes in the value of β, the algorithm maintains relative stability for large and small values. It also suggests that a proper value of β can significantly help clustering.

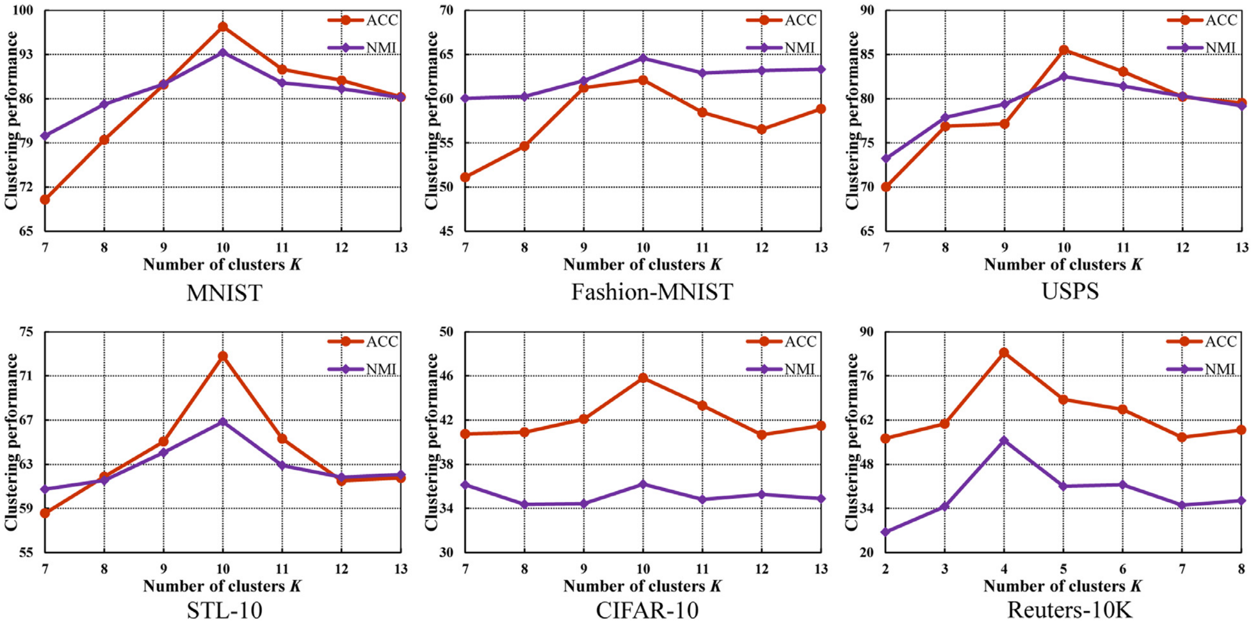


Fig. 8. The influence of different values of cluster numbers K to the clustering performance. Note that the value range of K on Reuters-10K is $\{2, 3, \ldots, 7, 8\}$ due to its true number of categories, and for the other data sets is $\{7, 8, \ldots, 12, 13\}$.

4.7.2. Influence of different cluster numbers

In the real world, it is hard to gain the actual number of clusters. To this end, we conduct experiment to evaluate the influence of different cluster numbers K to the clustering performance of DCCF. To be specific, we set the variation range of the cluster numbers K to $\{7, 8, \ldots, 12, 13\}$ for the data sets other than Reuters-10K. Since the true number of categories of Reuters-10K is 4, we set K to $\{2, 3, \ldots, 7, 8\}$.

As shown in Fig. 8, both ACC and NMI are reported for each data sets. We can observe that when the number of clusters K is different from the real cluster numbers, the clustering performance is affected to a certain extent. Among them, the overestimation of cluster numbers will not seriously hurt the clustering performance, while the underestimation will have a relatively greater impact on it. But the algorithm also shows great stability under different values of K. For instance, on the CIFAR-10 data set, as the value of K changes, the fluctuation of ACC and NMI are within 6% and 2%. In addition, based on empirical analysis, we find that compared with ACC, the fluctuation of NMI is relatively insignificant, which may be due to the fact that ACC is more sensitive to cluster numbers K.

4.8. Convergence analysis

In this section, we conduct a convergence analysis for the proposed DCCF method. Specifically, we experiment with the MNIST database, set the training epoch to 200, and then report the total loss, ACC and NMI. The convergence curve of the DCCF method is illustrated in Fig. 9. Note that the x-axis represents the training epoch and the y-axis from left to right denotes total loss, ACC, and NMI, respectively.

In terms of convergence speed, our method has great advantages. As can be seen from Fig. 9, the algorithm basically reaches convergence after 50 training epochs, which shows the capability of our method to reach convergence quickly. On the other hand, our algorithm has also been proved to have high stability. As shown in Fig. 9, the scores of ACC and NMI in DCCF method fluctuate within 1% after 50 training epochs, which fully proves the robustness of our method.

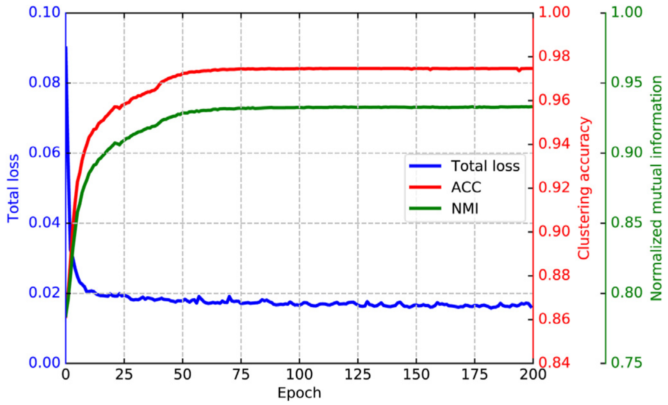


Fig. 9. The convergence curve of the DCCF method in the MNIST data set. Note that the x-axis represents the training epoch which is fixed at 200, and the y-axis from left to right denotes total loss, ACC, and NMI, respectively.

5. Conclusion and further discussion

In this paper, the framework of deep clustering with contractive representation learning and focal loss (DCCF) was proposed to solve the existing shortcomings of deep clustering. The proposed method forced a penalty term of the Jacobian matrix in feature learning to learn the more effective features, and introduced the focal loss to improve the label assignment mechanism in clustering layer. To tackle the challenge that the employment of focal loss requires real labels, we took advantage of the self-training in deep clustering, and designed a mechanism to apply focal loss in an unsupervised manner. To our best knowledge, this is the first work to introduce the focal loss into unsupervised clustering tasks. Moreover, we compared with several clustering methods on seven publicly available data sets, and the comprehensive experiments demonstrated the effectiveness of our method in several clustering tasks. Nevertheless, there are still some issues worth considering that may be beneficial to clustering, such as the potential shown by semi-supervised learning in clustering tasks, which may further improve the label assignment mechanism of this paper. Therefore, in future work, we will focus on exploring the improvement of deep clustering framework more deeply through applying the idea of semi-supervised learning to our work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is in part supported by the National Natural Science Foundation of China (Grant no. U1705262), the Natural Science Foundation of Fujian Province (Grant nos. 2020J01130193 and 2018J07005).

References

[1] Z. Liu, G. Lin, W.L. Goh, Bottom-up scene text detection with Markov clustering networks, Int. J. Comput. Vis. 128 (2020) 1786–1809.
[2] G. Wang, W. Li, M.A. Zuluaga, R. Pratt, P.A. Patel, M. Aertsen, T. Doel, A.L. David, J. Deprest, S. Ourselin, et al., Interactive medical image segmentation using deep learning with image-specific fine tuning, IEEE Trans. Med. Imaging 37 (7) (2018) 1562–1573.
[3] P. Tang, X. Wang, S. Bai, W. Shen, X. Bai, W. Liu, A. Yuille, PCL: proposal cluster learning for weakly supervised object detection, IEEE Trans. Pattern Anal. Mach. Intell. 42 (1) (2018) 176–191.
[4] C. Alzate, J.A. Suykens, Multiway spectral clustering with out-of-sample extensions through weighted kernel pca, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2) (2008) 335–347.
[5] C.-H. Zheng, D.-S. Huang, L. Zhang, X.-Z. Kong, Tumor clustering using nonnegative matrix factorization with gene selection, IEEE Trans. Inf. Technol. Biomed. 13 (4) (2009) 599–607.
[6] F. Tian, B. Gao, Q. Cui, E. Chen, T.-Y. Liu, Learning deep representations for graph clustering, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 1293–1299.
[7] Z. Dang, C. Deng, X. Yang, H. Huang, Multi-scale fusion subspace clustering using similarity constraint, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6658–6667.
[8] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 132–149.
[9] S. Baek, G. Yoon, J. Song, S.M. Yoon, Deep self-representative subspace clustering network, Pattern Recognit. 118 (2021) 108041.
[10] J. Fan, Deep self-representative subspace clustering network, Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, ACM, 2021, pp. 342–352.
[11] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: Proceedings of the International Conference on Machine Learning, 2016, pp. 478–487.
[12] X. Guo, L. Gao, X. Liu, J. Yin, Improved deep embedded clustering with local structure preservation, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2017, pp. 1753–1759.
[13] X. Yang, C. Deng, F. Zheng, J. Yan, W. Liu, Deep spectral clustering using dual autoencoder network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4066–4075.
[14] X. Peng, Z. Yu, Z. Yi, H. Tang, Constructing the L2-graph for robust subspace learning and subspace clustering, IEEE Trans. Cybern. 47 (4) (2016) 1053–1066.
[15] S. Wang, W. Guo, Sparse multigraph embedding for multimodal feature representation, IEEE Trans. Multimed. 19 (7) (2017) 1454–1466.
[16] W. Guo, Y. Shi, S. Wang, A unified scheme for distance metric learning and clustering via rank-reduced regression, IEEE Trans. Syst., Man, Cybern. 51 (8) (2021) 5218–5229.
[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[18] W. Shu, S. Wang, Q. Chen, Y. Hu, Z. Cai, R. Lin, Pathological image classification of breast cancer based on residual network and focal loss, in: Proceedings of the International Conference on Computer Science and Artificial Intelligence, 2019, pp. 211–214.
[19] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio, Contractive auto-encoders: explicit invariance during feature extraction, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, 2011, pp. 833–840.
[20] L. Yang, N.-M. Cheung, J. Li, J. Fang, Deep clustering by gaussian mixture variational autoencoders with graph embedding, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6440–6449.
[21] J. Fan, C. Yang, M. Udell, Robust non-linear matrix factorization for dictionary learning, denoising, and clustering, IEEE Trans. Signal Process. 69 (2021) 1755–1770.
[22] K.K. Bharti, P.K. Singh, Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering, Expert Syst. Appl. 42 (6) (2015) 3105–3114.
[23] P. Zhu, W. Zhu, Q. Hu, C. Zhang, W. Zuo, Subspace clustering guided unsupervised feature selection, Pattern Recognit. 66 (2017) 364–374.
[24] X. Yang, C. Deng, X. Liu, F. Nie, New l21-norm relaxation of multi-way graph cut for clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018, pp. 4374–4381.
[25] Y.-M. Xu, C.-D. Wang, J.-H. Lai, Weighted multi-view clustering with feature selection, Pattern Recognit. 53 (2016) 25–35.
[26] X. Yang, C. Deng, K. Wei, J. Yan, W. Liu, Adversarial learning for robust deep clustering, in: Advances in Neural Information Processing Systems, 33, 2020, pp. 9098–9108.
[27] Z. Dang, C. Deng, X. Yang, K. Wei, H. Huang, Nearest neighbor matching for deep clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13693–13702.
[28] Y. Wang, X. Zhao, X. Hu, Y. Li, K. Huang, Focal boundary guided salient object detection, IEEE Trans. Image Process. 28 (6) (2019) 2813–2824.
[29] D. Cai, X. He, X. Wang, H. Bao, J. Han, Locality preserving nonnegative matrix factorization, in: Proceedings of the International Joint Conference on Artificial Intelligence, 9, 2009, pp. 1010–1015.
[30] G.E. Hinton, R.S. Zemel, Autoencoders, minimum description length and Helmholtz free energy, in: Advances in Neural Information Processing Systems, 6, 1994, pp. 3–10.
[31] A. Gogna, A. Majumdar, R. Ward, Semi-supervised stacked label consistent autoencoder for reconstruction and analysis of biomedical signals, IEEE Trans. Biomed. Eng. 64 (9) (2016) 2196–2205.
[32] J. Zhao, M. Mathieu, R. Goroshin, Y. Lecun, Stacked what-where auto-encoders, arXiv preprint arXiv:1506.02351(2015).
[33] Z. Jiang, Y. Zheng, H. Tan, B. Tang, H. Zhou, Variational deep embedding: an unsupervised and generative approach to clustering, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2017, pp. 1965–1972.
[34] J. Yang, D. Parikh, D. Batra, Joint unsupervised learning of deep representations and image clusters, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 5147–5156.
[35] Y. Ren, K. Hu, X. Dai, L. Pan, S.C. Hoi, Z. Xu, Semi-supervised deep embedded clustering, Neurocomputing 325 (2019) 121–130.
[36] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, J.T. Zhou, Deep clustering with sample-assignment invariance prior, IEEE Trans. Neural Netw. Learn. Syst. 31 (11) (2020) 4857–4868.
[37] S. Mukherjee, H. Asnani, E. Lin, S. Kannan, Clustergan: latent space clustering in generative adversarial networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 33, 2019, pp. 4610–4617.
[38] Y. Opochinsky, S.E. Chazan, S. Gannot, J. Goldberger, K-autoencoders deep clustering, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2020, pp. 4037–4041.
[39] L. Yang, W. Fan, N. Bouguila, Clustering analysis via deep generative models with mixture models, IEEE Trans. Neural Netw. Learn. Syst. (2020), doi:10.1109/TNNLS.2020.3027761.
[40] X. Ji, J.F. Henriques, A. Vedaldi, Invariant information clustering for unsupervised image classification and segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9865–9874.
[41] C. Niu, J. Zhang, G. Wang, J. Liang, Gatcluster: self-supervised gaussian-attention network for image clustering, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 735–751.

Jinyu Cai received the B.S. degree in computer science and technology from Fuzhou University, Fuzhou, China, in 2018. He is currently pursuing the Ph.D. degree with the College of Computer and Data Science, Fuzhou University. His research interests include machine learning, computer vision and pattern recognition.

Shiping Wang received the Ph.D. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China in 2014. He is currently a Full Professor with the College of Computer and Data Science, Fuzhou University. His research interests include machine learning and computer vision.

Chaoyang Xu received the M.S. degree from College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, in 2009. He is currently working in School of Information Engineering, Putian University, Putian 351100, China; His research interests include deep learning and computer vision.

Wenzhong Guo received the Ph.D. degree in communication and information system from Fuzhou University in 2010. He is currently a Full Professor with the College of Mathematics and Computer Science, Fuzhou University. His research interests include cloud computing, mobile computing, and evolutionary computation. Currently, he leads the Network Computing and Intelligent Information Processing Laboratory, which is a Key Laboratory of Fujian Province, China.