Knowledge-Based Systems 252 (2022) 109346

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Active deep image clustering

Bicheng Sun [a], Peng Zhou [a,b,*], Liang Du [c], Xuejun Li [a]

[a] School of Computer Science and Technology, Anhui University, Hefei 230601, China
[b] The State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[c] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China

ARTICLE INFO

Article history:
Received 13 December 2021
Received in revised form 24 June 2022
Accepted 24 June 2022
Available online 1 July 2022

Keywords:
Clustering
Deep clustering
Active clustering

ABSTRACT

Deep clustering has attracted increasingly more attention in recent years. However, due to the absence of labels, deep clustering sometimes still provides unreliable clustering results. Although semi-supervised deep clustering can alleviate this problem to some extent by involving few human annotations, we observe that the performance of semi-supervised clustering highly depends on the selection of data for human labeling, but unfortunately, the supervised information selection is still a tough problem as traditional semi-supervised methods pay no attention to it. To tackle this problem, in this paper, we propose a novel deep active clustering method, which can actively select the key data for human labeling and apply the human annotations to improve the deep clustering. Different from conventional semi-supervised deep clustering methods which use fixed pre-given supervised information, we design a simple yet effective strategy to select the informative and uncertain data for querying annotation, which is beneficial to the clustering task. Furthermore, we integrate deep representation learning, clustering, and data selection into a unified framework, so that each task can be boosted by each other. Finally, we conduct extensive experiments on benchmark data sets by comparing it with some state-of-the-art deep clustering methods and semi-supervised clustering methods. The experimental results show that our active clustering methods can outperform both the unsupervised and semi-supervised clustering methods, demonstrating the effectiveness of the proposed method. The codes of this paper are released in https://doctor-nobody.github.io/codes/ADC_codes.zip.

1. Introduction

Clustering is a fundamental problem in unsupervised learning, and many classical clustering methods have been proposed in recent decades, such as kmeans [1], spectral clustering [2], and subspace clustering [3]. These methods aim to partition data into several clusters, such that the data in the same cluster will be close to each other and those in different clusters will be far apart from each other. Since the conventional methods often partition data in the original feature space, which may not reveal the intrinsic structure of data, the performance may be limited.

To address this issue, deep clustering has been proposed [4–9]. Deep clustering applies a deep neural network to learn an embedding representation, which can better characterize the intrinsic structure of data, and then partition data with the learned representation. For example, Xie et al. proposed the Deep Embedded Clustering (DEC), which applied an auto-encoder to obtain the latent representation and then used the Kullback–Leibler (KL) divergence minimization to obtain the final clustering results [4]; Yang et al. adopted the Convolutional Neural Network (CNN) to extract the latent embedding of images and performed clustering method on such embedding [5]. Due to the strong representation capacity of the deep neural networks, deep clustering often achieves a better performance. However, deep clustering is an unsupervised method after all, and it may also be misled by some noisy or difficult data due to the absence of the labels.

One natural way to address this issue is to obtain some supervised information for clustering, leading to the semi-supervised deep clustering [10–14]. In many real-world applications, the label of each data is difficult to obtain, but the pairwise relation of data is much easier to access. For example, in the face recognition task, we often do not know the name (i.e., the label) of the person in the image, but we can easily tell whether the two persons in the two images are the same person. Therefore, many semi-supervised methods apply the pairwise constraints (i.e., whether the two data are in the same class) as the supervised information which will be helpful to the clustering. For example, Ren et al. extended DEC to semi-supervised clustering by using the must-link and cannot-link constraints [10]; Vilhagra et al. proposed a

* Corresponding author at: School of Computer Science and Technology, Anhui University, Hefei 230601, China.
E-mail addresses: e19301144@stu.ahu.edu.cn (B. Sun), zhoupeng@ahu.edu.cn (P. Zhou), duliang@sxu.edu.com (L. Du), xjli@ahu.edu.cn (X. Li).

https://doi.org/10.1016/j.knosys.2022.109346
0950-7051/

Siamese network [15] using the pairwise constraints for text data clustering [13]. However, the performance of semi-supervised clustering often highly depends on the quality of the pairwise constraints, but unfortunately, how to appropriately select such pairwise constraints itself is a tough and challenging task.

To tackle this problem, in this paper, we propose an innovative Active Deep Clustering (ADC) method, which automatically selects the key pairs of data for labeling and applies these labeled pairs to improve the deep clustering. Different from the conventional methods which use fixed pre-given pairwise constraints and pay no attention to the constraints selection, we design a simple yet effective strategy to select the pairwise constraints. To select the key data which are helpful for clustering, we evaluate each pair by fully considering both the *uncertain* and *informative* properties. On one hand, the uncertain pairs often contain some difficult data which may mislead the clustering model. If we obtain the experts' annotations on these data, they can alleviate the misleading caused by the difficult data and thus improve the clustering performance. On the other hand, the goal of clustering is to find out which data are in the same cluster, or equivalently speaking, which data have the must-link constraints. Hence, must-link constraints are more informative than cannot-link constraints in clustering tasks. By finding out these uncertain and informative pairs for querying experts' annotations, we can obtain a more reliable clustering result. Since we try to apply these pairwise constraints, naturally, we first use the Siamese network to learn the representation. Moreover, in this paper, we focus on image clustering, and thus we use CNN as the backbone of the Siamese network. To obtain the clustering results and select the key pairs, we plug a clustering layer in the Siamese network, which can reveal the clustering structure of the data. Then, we use the aforementioned strategy to select key pairs for annotation according to the results of the clustering layer. We seamlessly integrate the representation learning, clustering, and the constraints active selection into a unified framework, so that the three tasks can be boosted by each other. At last, we obtain the final clustering results by the clustering layer.

It is worthy to highlight the contributions of this paper here:

- We propose a novel active deep clustering framework that simultaneously learns the representation, does clustering, and selects data for labeling.
- We design a simple yet effective strategy to select the informative and uncertain pairs for labeling, which is beneficial to our deep clustering method.
- The experiments on benchmark data sets show that the proposed method outperforms not only the unsupervised deep clustering methods but also the state-of-the-art semi-supervised deep clustering methods.

2. Related work

In this section, we briefly introduce some related works including deep clustering, semi-supervised deep clustering, and active learning.

2.1. Deep clustering

Clustering aims to partition data into multiple clusters, so that the similar data are in the same cluster and the dissimilar data are in different clusters. Since clustering does not need any labels, it attracts much attention in past decades [16–24]. In recent years, with deep learning [25] being applied to various domains and achieving promising performance, some works consider applying deep learning to clustering, leading to deep clustering.

The basic idea of deep clustering is to use a deep neural network to learn an embedding of the original data and do clustering on such learned embedding. Among various deep architectures, one simple yet effective neural network is the auto-encoder. Therefore, many methods applied auto-encoder to extract the latent embedding for clustering. For example, Yang et al. learned the embedding with auto-encoder followed by fuzzy c-means to obtain the final clustering result [26]; Yang et al. applied auto-encoder to learn a kmeans-friendly embedding for clustering [27]; Ji et al. combined the auto-encoder and the subspace clustering, leading to a deep subspace clustering method [28]; Fard et al. jointly learned the latent representations with auto-encoder and did kmeans clustering [29]; Lv et al. proposed deep subspace clustering method with the pseudo-labels [30].

Since the convolutional neural network has demonstrated promising performance in many tasks, especially in image processing tasks, CNN has also been used in deep clustering. For example, Yang et al. extended CNN to a recurrent framework to obtain the embedding and applied agglomerative clustering to partition the data [5]; Li et al. designed a convolutional auto-encoder to extract the latent representation for image clustering [31]; Guérin et al. proposed multiple pretrained CNN for image clustering [32]; Lin et al. used density clustering to partition the data on the latent representation learned by CNN [33]; Caron et al. proposed an end-to-end deep clustering method with CNN [34]; Li et al. proposed contrastive clustering which applied CNN to extract the latent representation of both the data and clusters [35]; most recently, Liu et al. constructed multiple graphs from the multiple views of data and proposed a new graph convolutional networks (GCN) based clustering method, which captured the stationary diffusion state of the multiple graphs [36].

Another famous unsupervised deep architecture is the generative model like Generative Adversarial Networks (GAN) or Variational Auto-Encoder (VAE). Some work applied the generative model to clustering. For example, Dilokthanakul et al. and Jiang et al. used VAE for clustering [37,38]; Chen et al. developed an interpretable representation learning for clustering by GAN [39]; Yu et al. mixed multiple GANs for clustering [40]; Zhou et al. proposed a deep adversarial subspace clustering method [41]; Mukherjee et al. proposed ClusterGAN which applied GAN to clustering [42]. Different from GAN and VAE, some methods tried to generate pseudo-labels for semantic clustering [43,44]. For example, Gansbeke et al. generated the pseudo-labels by mining nearest neighbors in the latent embedding space [43]; Park et al. obtained the pseudo-labels by an off-the-shelf unsupervised clustering method, and obtain the final clustering results with robust learning [44]. Since these methods generate pseudo-labels with high quality, they achieve state-of-the-art performance. Different from these methods which focus on how to generate the high-quality pseudo-labels, this paper concentrates on how to select the key data for querying the annotations from human experts. Since the labels used in our method are obtained from human experts, they may be more reliable than those pseudo-labels.

2.2. Semi-supervised deep clustering

Although deep clustering achieves better performance than conventional clustering methods, since they do not have any guidance of human annotation, they may still be misled by some difficult or unreliable data. To address this problem, some semi-supervised deep clustering methods are proposed [10,12,13,45–47]. For example, Shukla et al. proposed a semi-supervised deep kmeans method, which needs the class labels of some instances [12].

However, in clustering tasks, it may be difficult to obtain the class labels of data. Sometimes human experts even do not know the label space of data. Therefore, one more appropriate way is to obtain the pairwise constraints. In more detail, the supervised information contains some must-link and cannot-link constraints, i.e., if two data belong to the same class, they have a must-link; and if two data belong to different classes, they have a cannot-link. For example, Fogel et al. used an auto-encoder to learn the embedding and applied the pairwise constraints to construct a pairwise loss on the embedding [45]; Zhang et al. proposed a unified framework which can handle various kinds of constraints [46]; Ohi et al. designed a convolutional auto-encoder with pairwise loss to learn the latent representation for clustering [47].

Although semi-supervised deep clustering used the supervised information to alleviate the unreliability of the data to some extent, the performance highly depends on the selection of the supervised constraints. Unfortunately, in the semi-supervised clustering, they assume that the supervised constraints are pre-given and do not consider how to select the supervised constraints. In this paper, we apply active learning to tackle this problem.

2.3. Active learning

Active learning selects informative data for human annotation when handling unlabeled data [48]. It aims to train a classifier that has a good generalization performance with only few selected labeled data. One famous setting of active learning is batch mode active learning, which is also the one we focus on in this paper.

Batch mode active learning selects a batch of data for labeling in each iteration [49–55]. Given a data set with n instances $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, batch mode active learning divides it into two disjoint sets: labeled set $\mathcal{L}$ and unlabeled set $\mathcal{U}$, where $\mathcal{L} \cup \mathcal{U} = \mathcal{X}$ and $\mathcal{L} \cap \mathcal{U} = \varnothing$. In $\mathcal{L}$, each instance has been labeled by human, and in $\mathcal{U}$ all instances are unlabeled. Given a batch size k, batch mode active learning iteratively selects a batch of data $\mathcal{S} \subset \mathcal{U}$ where $|\mathcal{S}| = k$ ($|\cdot|$ is the number of instances in a set) for labeling until there is no budget.

Different batch mode active learning methods often select batches based on different strategies. For example, Hoi et al. selected batches with Fisher information matrix [49]; Chattopadhyay et al. proposed an active learning method based on marginal probability distribution matching [50]; Wang et al. applied α-relative Pearson divergence to select batches [51]. Most batch mode active learning methods are designed for classification tasks, and thus they select instances for querying their labels. However, in this paper, we plug the active learning into the deep clustering method, where we select a pair of data $(\mathbf{x}_i, \mathbf{x}_j)$ for querying whether they belong to the same cluster. In clustering tasks, since the label space is often unknown, our scheme for labeling with must-link and cannot-link is simpler and more practical.

3. Active deep clustering

In this section, we introduce our active deep clustering in more detail. The key to active clustering is to answer the following two questions: (1) How to select key data for annotation? (2) How to use the human annotation to do clustering? To answer these two questions, we propose the ADC framework which is shown in Fig. 1. We first consider the second question. With the human annotation, we first use a Siamese network [15] to learn the representation of the data and apply a clustering layer to obtain the clustering result. To answer the first question, according to the clustering result, we design a strategy to select key pairs of data for annotation and use the annotation to guide the Siamese network training in turn. Therefore, in our method, we integrate the representation learning, clustering, and constraints active selection into a unified framework, so that each part can be boosted by each other. In the following, we will introduce each part in more detail, respectively.

3.1. Representation learning

Inspired by metric learning [56], we need to map the original data into a new semantic latent space, so that in this latent space, the data in the same cluster are close and the data in different clusters are far apart from each other. To achieve this, given a data instance $\mathbf{x}$, we need to learn the latent representation $f(\mathbf{x}; \boldsymbol{\theta})$, where $f(\cdot; \boldsymbol{\theta})$ is the map function to map $\mathbf{x}$ into such latent space, and $\boldsymbol{\theta}$ is the set of learned parameters in the map function f. In the latent space, we wish that, given two instances $\mathbf{x}_i$ and $\mathbf{x}_j$, if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster, i.e., there is a must-link between them, $f(\mathbf{x}_i; \boldsymbol{\theta})$ and $f(\mathbf{x}_j; \boldsymbol{\theta})$ should be as close to each other as possible; and if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different clusters, i.e., there is a cannot-link constraint between them, $f(\mathbf{x}_i; \boldsymbol{\theta})$ and $f(\mathbf{x}_j; \boldsymbol{\theta})$ should be far apart from each other.

In this subsection, we assume that we already have a must-link set $\mathcal{M}$ and a cannot-link set $\mathcal{C}$. In the first iteration, for the efficiency consideration, we randomly select k pairs for human annotation to obtain the initial $\mathcal{M}$ and $\mathcal{C}$. In the following iterations, $\mathcal{M}$ and $\mathcal{C}$ are constructed with the constraints active selection method, which will be introduced in more detail in Section 3.3. In this paper, we use the similarity metric $\|f(\mathbf{x}_i; \boldsymbol{\theta}) - f(\mathbf{x}_j; \boldsymbol{\theta})\|_2$, which is the Euclidean distance in the latent space. To this end, we should optimize the following two objective functions, in the cases that the pair is a must-link constraint and a cannot-link constraint, respectively:

$$\begin{cases} \min_{\boldsymbol{\theta}} \quad \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}} \quad \|f(\mathbf{x}_i; \boldsymbol{\theta}) - f(\mathbf{x}_j; \boldsymbol{\theta})\|_2^2, \\ \max_{\boldsymbol{\theta}} \quad \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} \quad \|f(\mathbf{x}_i; \boldsymbol{\theta}) - f(\mathbf{x}_j; \boldsymbol{\theta})\|_2^2. \end{cases} \tag{1}$$

In the cannot-link case, instead of maximizing $\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}} \|f(\mathbf{x}_i; \boldsymbol{\theta}) - f(\mathbf{x}_j; \boldsymbol{\theta})\|_2^2$ directly by the second objective function, we wish to maximize the margin of different clusters to improve the generalizability of the model. To fulfill this, inspired by [13], we introduce a pre-defined margin $\delta > 0$, and wish the distance of two data in cannot-link should be larger than δ, or we impose a penalty on it. More formally, we wish to minimize $\max(\delta - \|f(\mathbf{x}_i; \boldsymbol{\theta}) - f(\mathbf{x}_j; \boldsymbol{\theta})\|_2, 0)^2$, which means if the distance is larger than δ, there is no penalty and otherwise, the penalty is the square of the difference between the distance and the margin (something like the square of hinge loss). Then, by denoting an indicator y_{ij} such that $y_{ij} = 1$ if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$ and $y_{ij} = 0$ if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$, we can combine these two objectives into a unified objective function:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \mathcal{L}_1 = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}\cup\mathcal{C}} \Big(& y_{ij} \|f(\mathbf{x}_i; \boldsymbol{\theta}) - f(\mathbf{x}_j; \boldsymbol{\theta})\|_2^2 \\ & + (1 - y_{ij}) \max(\delta - \|f(\mathbf{x}_i; \boldsymbol{\theta}) - f(\mathbf{x}_j; \boldsymbol{\theta})\|_2, 0)^2 \Big) \end{aligned} \tag{2}$$

Then, we define the form of the map function $f(.; \boldsymbol{\theta})$. Because the same map function $f(.; \boldsymbol{\theta})$ with the same $\boldsymbol{\theta}$ is used to process both must-link and cannot-link constraints, one natural option for realizing such a similarity metric is to use the Siamese network [15]. Since we focus on image clustering, we use the twin CNN modules whose weights are shared as the backbone to extract the representation. Here, we use a simple yet effective backbone, whose structure is shown in Fig. 2, which consists of several convolutional layers and pooling layers and at last is followed by a full connection layer.

We first select a pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M} \cup \mathcal{C}$ to construct a triplet $(\mathbf{x}_i, \mathbf{x}_j, y_{ij})$. Then we feed $\mathbf{x}_i$ and $\mathbf{x}_j$ into the two shared-weight CNN modules shown in Fig. 2. $f(\mathbf{x}_i; \boldsymbol{\theta})$ and $f(\mathbf{x}_j; \boldsymbol{\theta})$ are the output of the CNN module with inputs $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, where $\boldsymbol{\theta}$ is the set of weight parameters in the CNN. The Siamese network learns the parameter $\boldsymbol{\theta}$ by minimizing $\mathcal{L}_1$ in Eq. (2).

By minimizing $\mathcal{L}_1$ w.r.t. the network parameters $\boldsymbol{\theta}$, we can obtain an initial Siamese network that makes the latent representations of data in the same cluster be similar and makes the representations of data in different clusters be dissimilar.

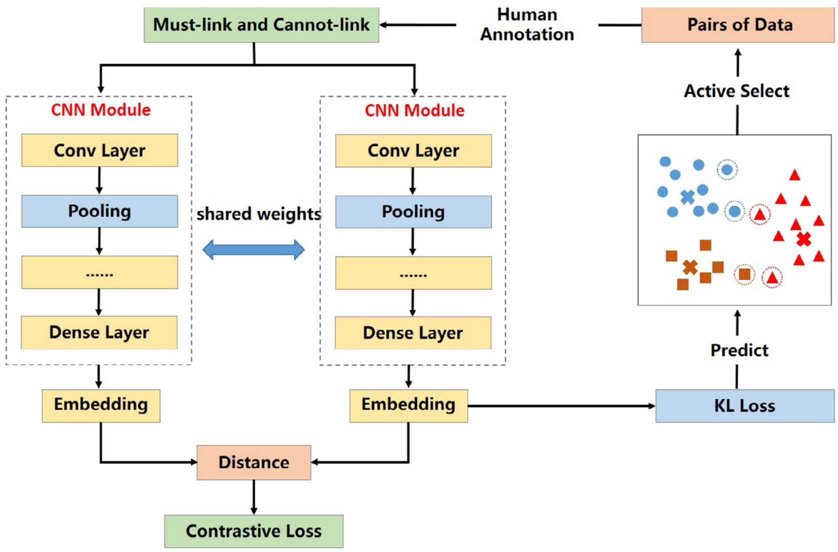


Fig. 1. The framework of ADC. The framework contains three parts. The first part is the representation learning module with contrastive loss. It is a Siamese network that contains two CNN modules with shared weights. The second part is a clustering layer with KL divergence loss. After the Siamese network we obtain the latent representation of all data and then we feed these latent representations into the clustering layer to obtain the clustering results. The third part is the constraints active selection module, which selects pairs for annotation according to the information generated in the clustering layer.

Fig. 2. The backbone of the CNN module used in the Siamese network.

3.2. Clustering layer

Given a pre-defined number of clusters c, inspired by DEC [4], we also simultaneously fine-tune the parameters $\boldsymbol{\theta}$ in the Siamese network and the c cluster centers $\{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_c\}$. Firstly, we initialize the c cluster centers by running kmeans on the outputs $f(\mathbf{x}_i; \boldsymbol{\theta})$ of the initial Siamese network.

Then following DEC [4], we use the Student's t-distribution to measure the similarity between the latent representation $f(\mathbf{x}_i; \boldsymbol{\theta})$ and the centroid $\boldsymbol{\mu}_j$:

$$q_{ij} = \frac{(1 + \|f(\mathbf{x}_i; \boldsymbol{\theta}) - \boldsymbol{\mu}_j\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'=1}^{c}(1 + \|f(\mathbf{x}_i; \boldsymbol{\theta}) - \boldsymbol{\mu}_j'\|_2^2/\alpha)^{-\frac{\alpha+1}{2}}}. \tag{3}$$

q_{ij} can be regarded as a soft assignment of each instance. To obtain a harder cluster assignment, we introduce a sharper auxiliary distribution p_{ij} of the assignment:

$$p_{ij} = \frac{q_{ij}^2 / \sum_{i=1}^{n} q_{ij}}{\sum_{j'=1}^{c}(q_{ij'}^2 / \sum_{i=1}^{n} q_{ij'})}. \tag{4}$$

Then, we wish the two distribution p_{ij} and q_{ij} be as closed to each other as possible. To this end, we minimize the KL divergence between them as follows:

$$\mathcal{L}_2 = \sum_{i=1}^{n}\sum_{j=1}^{c} p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{5}$$

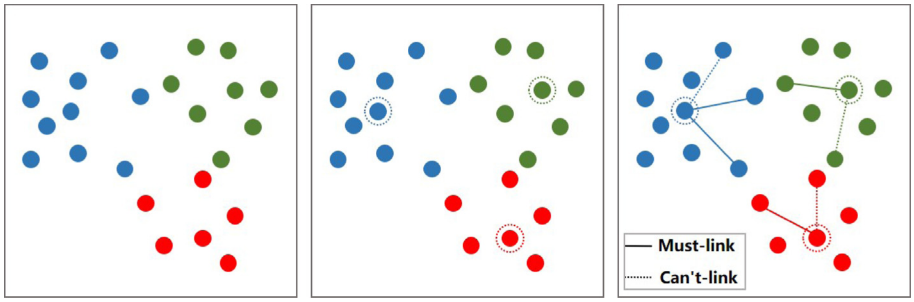
Then we minimize $\boldsymbol{\theta}$ and $\boldsymbol{\mu}_i$ by stochastic gradient descent (SGD) with computing the gradients $\frac{\partial \mathcal{L}_2}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \mathcal{L}_2}{\partial \boldsymbol{\mu}_i}$. After the convergence, we obtain the clustering result of $\mathbf{x}_i$ as $\text{argmax}_j\, q_{ij}$.

3.3. Constraints active selection

After obtaining the distribution q_{ij}, for the ith instance, many conventional clustering methods only use the largest one q_{im} for clustering, where $m = \text{argmax}_j\, q_{ij}$, whereas they discard all other $q_{ij'}$ where $j' \neq m$. However, when we take a closer look at those q_{ij}, we observe that although we do not use other $q_{ij'}$ for clustering directly, they are still very useful for us to select the pairs for querying human labeling.

Intuitively, to achieve a better clustering performance, we observe that the ideal pairs $(\mathbf{x}_i, \mathbf{x}_j)$ for querying annotation should satisfy the following properties:

- *Uncertain*. If the model can easily determine whether $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same cluster, there is no need for querying human annotation. Therefore we want to select the uncertain pairs for labeling, and the obtained human annotations can be most helpful for our clustering.
- *Informative*. It is easy to verify that there are much more cannot-link constraints than the must-link ones. However, the goal of clustering is to determine which instances are in the same cluster. Therefore, must-link constraints are much more informative than cannot-link constraints and



(a) Original clustering results. (b) Find the instance in the center of the cluster ($\mathbf{x}_{p_1}$). (c) Find the most uncertain instances for querying human annotation.

Fig. 3. An illustration of constraints active selection. (a) shows the original clustering results, where the three clusters are denoted by blue, green and red circles, respectively. (b) shows the $\mathbf{x}_{p_1}$ of each cluster, which is the instance in the center of each cluster. These instances are in the circle marked with blue-dotted, green-dotted and red-dotted lines, respectively. (c) finds the most uncertain instances in each cluster, which are the instances farthest away from the center. Then, together with $\mathbf{x}_{p_1}$, they form pairs for querying human annotations. The solid line denotes the must-link and the dashed line denotes the cannot-link constructed by human annotations.

thus they are much more helpful for clustering than cannot-link constraints.

According to the above observations, we design a strategy to select the uncertain and informative pairs for labeling by using all assignments q_{ij}. In more detail, given the budget of each batch k, at each time of selection, we distribute the budget to each cluster according to the number of instances in each cluster obtained by the previous clustering step. Assume that there are c clusters and the c clusters contains $n_1, \ldots, n_c$ instances, respectively, where $\sum_{p=1}^{c} n_p = n$. Then, we distribute $b_p = [k * \frac{n_p}{\sum_{p=1}^{c} n_p}]$ queries to the pth cluster, i.e., we select b_p pairs from the pth cluster for annotation, where $[x]$ is the largest integer which is no larger than x. Since the data in a pair are selected in the same cluster, we prefer to select the potential must-link constraints which satisfy the *informative* property.

Now, we focus on one of the cluster and we take the pth cluster π_p as an example. In this cluster, we wish to select some uncertain pairs for annotation. Suppose there are n_p instances in the pth cluster. We first sort the n_p instances in the descend order of q_{ip} as $\pi_p = \{\mathbf{x}_{p_1}, \ldots, \mathbf{x}_{p_{n_p}}\}$ where $q_{p_1 p} \geq \cdots \geq q_{p_{n_p} p}$. According to the previous subsection, we know that q_{ip} can be viewed as a similarity of instances $\mathbf{x}_i$ and the cluster center $\boldsymbol{\mu}_p$, i.e., the larger the q_{ip} is, the closer $\mathbf{x}_i$ is to the center. Therefore, we can select the *uncertain* pairs with q_{ip}.

Specifically, $\mathbf{x}_{p_1}$ has the largest q value, which means $\mathbf{x}_{p_1}$ may stay in the center of the cluster. Similarly, $\mathbf{x}_{p_{n_p}}$ has the smallest q value, which means it may be in the boundary of the cluster. Naturally, we can select $(\mathbf{x}_{p_1}, \mathbf{x}_{p_{n_p}})$ as an uncertain yet informative pair for annotation. Since we need to select b_p pairs for annotation, we can select $\{(\mathbf{x}_{p_1}, \mathbf{x}_{p_{n_p - b_p + 1}}), \ldots, (\mathbf{x}_{p_1}, \mathbf{x}_{p_{n_p}})\}$ for querying the human labels. In the next time of selection, we also sort the instances in the cluster according to the new q_{ip} and select the first one and the last b_p ones to form the b_p pairs. Notice that if a pair has been selected for annotation in previous iterations, we should skip it to avoid the waste of budget. Fig. 3 shows an illustration of this strategy.

The above strategy can select the informative yet uncertain pairs for annotation. However, in each iteration, for all candidate pairs, we should check whether they have been selected in previous iterations. With the process of iterations, this operation is time-consuming, because it needs to traverse the previously selected sets many times.

To tackle this problem, we propose a simpler yet effective method to avoid checking whether a pair has been selected before. Supposing the times of selection pairs (or equivalently speaking, the number of iterations) is T, we first equally divide the original data set $\mathcal{X}$ into T disjoint subsets $\mathcal{X}_1, \ldots, \mathcal{X}_T$, where $\mathcal{X}_1 \cup \mathcal{X}_2 \cup \cdots \cup \mathcal{X}_T = \mathcal{X}$ and $\mathcal{X}_i \cap \mathcal{X}_j = \varnothing$ for each $1 \leq i \neq j \leq T$. Then, in the tth iteration ($1 \leq t \leq T$), for the pth cluster π_p, we construct the new cluster set of the pth cluster as $\pi_p^{(t)} = \pi_p \cap \mathcal{X}_t$, and perform the selection operation introduced before on the new constructed cluster set $\pi_p^{(t)}$. Since $\mathcal{X}_i \cap \mathcal{X}_j = \varnothing$ and $\pi_p \cap \pi_q = \varnothing$, we have $\pi_p^{(i)} \cap \pi_q^{(j)} = \varnothing$ for all $i \neq j$ or $p \neq q$, and thus in each iteration, the selected pairs will not be repeated.

Algorithm 1 Constraints Active Selection

Input: Data set $\mathcal{X}$, p_{ij} and q_{ij} calculated by clustering layer, T subsets of $\mathcal{X}$: $\mathcal{X}_1, \cdots, \mathcal{X}_T$, the budget of each batch k, the number of the current iteration t.
Output: The selected pairs set $\mathcal{S}^{(t)}$ which contains k pairs for querying human annotation.

```
Partition X into c clusters π_1, ···, π_c according to q_ij values.
for p = 1, ···, c do
    Construct π_p^(t) = π_p ∩ X_t and compute the number of instances in it as n_p = |π_p^(t)|, where |·| is the number of instances in a set.
end for
Distribute the budget to each cluster: b_p = [k * n_p / Σ_{p=1}^{c} n_p].
for p = 1, ···, c do
    Sort π_p^(t) in the descend order of q_ip as π_p^(t) = {x_{p_1}, ···, x_{p_{n_p}}}.
    Obtain the selected set of the p-th cluster as S_p = {(x_{p_1}, x_{p_{n_p−b_p+1}}), ···, (x_{p_1}, x_{p_{n_p}})}.
end for
Obtain the selected set of the t-th iteration as S^(t) = S_1 ∪ S_2 ∪ ··· ∪ S_c for human annotation.
```

Algorithm 1 summarizes the process of the constraints selection in the tth iteration. After obtaining $\mathcal{S}^{(t)}$, we query the pairs in $\mathcal{S}^{(t)}$ for human annotations to construct the must-link constraint set $\mathcal{M}$ and the cannot-link constraint set $\mathcal{C}$. Then, we apply the $\mathcal{M}$ and $\mathcal{C}$ to train the Siamese network introduced in Section 3.1 for representation learning.

3.4. Algorithm and implementation details

The whole algorithm of our ADC is summarized in Algorithm 2. In our backbone CNN module shown in Fig. 2, for all convolution

Table 1
Information of the data sets.

	# of images	Size of images	# of classes
USPS	9298	16 × 16	10
MNIST	70000	28 × 28	10
Fashion MNIST	70000	28 × 28	10

layers, RELU is used as the active function, the size of the convolution kernel is 3 × 3, and the step size is 1. For the first three pooling layers, we use the 2 × 2 max pooling with step size 2. For the last pooling layer, we use the 6 × 6 average pooling with step size 2. The dimension of the output embedding is 10.

Algorithm 2 Active Deep Clustering

Input: Data set $\mathcal{X}$, the budget of each batch k, the numbers of iterations T.
Output: Clustering results.

for $t = 1, \cdots, T$ **do**
  if $t = 1$ **then**
    Randomly select k pairs from $\mathcal{X}$ for annotation and construct the initial $\mathcal{M}$ and $\mathcal{C}$.
  else
    Select the pairs for human annotation with Algorithm 1.
    Construct $\mathcal{M}$ and $\mathcal{C}$ with human annotations.
  end if
  Apply $\mathcal{M}$ and $\mathcal{C}$ to train the Siamese network with the contrastive loss function Eq. (2).
  Obtain the embedding of all instances with the trained Siamese network.
  Fine-tune the Siamese network with the clustering loss function Eq. (5).
  Obtain the q_{ij} for all instances.
end for
Obtain the final clustering results with $\mathrm{argmin}_j\, q_{ij}$.

When we train the Siamese network with pairwise constraints, the margin δ in Eq. (2) is fixed to 1, the batch size is 64 and the epoch is 100. We use RMSProp as the optimizer with a learning rate of 0.001. In the clustering layer, we fix the α in Eq. (3) as 1. When we use the clustering layer to fine-tune the network, we use the SGD with a learning rate of 0.001 as the optimizer.

4. Experiments

In this section, we compare our ADC with some state-of-the-art unsupervised and semi-supervised methods on benchmark data sets.

4.1. Data sets

We conduct experiments on three popular image data sets:

- **USPS.**[1] This is a data set, which contains 9298 handwritten digital images from the United States Postal Service. The size of each image is 16 × 16, and the images are categorized into 10 classes.
- **MNIST** [57]. This is also a benchmark data set of handwritten images, which contains 70000 images in 10 categories. The size of each image is 28 × 28.
- **Fashion MNIST** [58]. Fashion MNIST data set contains 70000 fashion product images with the size 28 × 28, and the images are also categorized in 10 classes.

The statistical information of each data set is summarized in Table 1.

4.2. Experimental setup

To demonstrate the effectiveness of the proposed method, we compare it with some classical clustering methods, deep clustering methods, and semi-supervised clustering methods. The classical clustering methods include kmeans [1] and spectral clustering (SC) [16]. The deep clustering methods include the following algorithms:

- **DEC** [4], which applies the auto-encoder to learn the embedding representations for clustering.
- **JULE** [5], which uses CNN in a recurrent framework for feature learning and clustering.
- **IDEC** [6], which improves DEC by combining the reconstruction loss and the clustering loss.
- **DSC** [59], which is a deep spectral clustering via dual auto-encoder.
- **ASPC-DA** [60], which is a self-paced deep clustering method with data augmentation.
- **GCML** [61], which performs multi-manifold learning and clustering by preserving geometric structure.
- **SCAN** [43], which is a deep semantic clustering method via generating pseudo-labels.
- **RUC** [44], which improves SCAN by applying robust learning to obtain the final clustering result.

Besides these unsupervised methods, we also compare with the following semi-supervised deep clustering methods:

- **SDEC** [10], which extends the DEC architecture to the semi-supervised clustering by introducing the pairwise constraints.
- **AutoEmbedder** [47], which introduces pairwise constraints to downsample high-dimensional data for deep clustering.
- **DCC** [14], which is a method using pairwise constraints in the proposed deep constrained clustering framework.
- **ADC-random**, which is a degenerated version of our ADC as an ablation study. To show the effectiveness of the constraints active selection strategy, in ADC-random, we replace the constraints active selection strategy in our ADC with the random selection, which randomly selects k pairs for annotation.

In our method, we set the number of batches for annotation $T = 5$ and the budget of each batch is $k = n/25$, where n is the number of instances in each data set. Therefore, the whole number of constraints we use is $T * k = n/5$. All hyper-parameters in our method are fixed for all data sets as introduced in Section 3.4. For all methods on all data sets, the number of clusters is set to be the true number of classes of each data set. For the compared semi-supervised methods SDEC, AutoEmbedder, and DCC, we run their codes with totally $n/5$ randomly selected constraints, which are the same number as ours for a fair comparison. For ADC-random, the setting is the same as ADC. To evaluate the performance of each method, we use three standard evaluation metrics including Accuracy (ACC), Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI). For ACC, NMI and ARI, the higher the value is, the better the clustering performance is.

All the experiments are conducted on a PC with an Intel Core i7-8750H CPU@2.20 GHz and an NVIDIA GTX 1050Ti GPU.

4.3. Experimental results

Table 2 shows the ACC and NMI results of our ADC and other compared methods on all data sets. From Table 2, we find that our method outperforms not only the state-of-the-art unsupervised deep clustering methods but also the state-of-the-art semi-supervised deep clustering methods. It well demonstrates the

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html.

Table 2
Clustering results of all methods.

Methods	USPS			MNIST			Fashion MNIST		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
kmeans [1]	0.668	0.627	0.546	0.532	0.500	0.366	0.474	0.512	0.349
SC [16]	0.649	0.794	0.638	0.656	0.731	0.511	0.508	0.575	0.368
DEC [4]	0.682	0.643	0.605	0.842	0.834	0.782	0.558	0.562	0.503
JULE [5]	0.950[a]	0.913[a]	–	0.964[a]	0.913[a]	–	0.563[a]	0.608[a]	–
IDEC [6]	0.725	0.677	0.575	0.847	0.823	0.768	0.611	0.623	0.467
DSC [59]	0.869[a]	0.857[a]	–	0.978[a]	0.941[a]	–	0.662[a]	0.645[a]	–
ASPC-DA [60]	0.982[a]	0.951[a]	–	0.988[a]	0.966[a]	–	0.591[a]	0.654[a]	–
GCML [61]	0.980[a]	0.946[a]	–	0.958[a]	0.902[a]	–	0.710[a]	0.685[a]	–
SCAN [43]	0.808	0.846	0.765	0.964	0.933	0.925	0.754	0.747	0.660
RUC [44]	0.907	0.924	0.893	0.973	0.948	0.942	0.748	0.720	0.638
SDEC [10]	0.738	0.727	0.639	0.808	0.767	0.697	0.579	0.623	0.455
AutoEmbedder [47]	0.628	0.587	0.574	0.972	0.947	0.950	0.853	0.761	0.701
DCC [14]	0.782	0.725	0.616	0.975	0.933	0.913	0.780	0.742	0.641
ADC-random	0.955	0.896	0.928	0.984	0.955	0.965	0.701	0.720	0.712
ADC	**0.988**	**0.965**	**0.984**	**0.996**	**0.987**	**0.993**	**0.880**	**0.820**	**0.785**

[a]denotes the results are those reported in their papers and other results are the reproduced results by us with the released source codes.

Table 3
Running time of all methods (Sec.).

Methods	USPS	MNIST	Fashion MNIST
DEC	932	2362	5216
IDEC	762	2056	4975
SCAN	28440	39600	46800
RUC	21600	97200	111600
SDEC	836	5188	4580
AutoEmbedder	873	7380	7400
DCC	2159	6167	7205
ADC	543	6220	7083

Table 4
Ablation study: running time compared with ADC-check (Sec.).

Methods	USPS	MNIST	Fashion MNIST
ADC-check	565	6429	7552
ADC	543	6220	7083

effectiveness and superiority of the proposed method. Especially on the Fashion MNIST data set, which may be the most difficult one because the accuracy of most methods is lower than 0.7, our method achieves a 3.2% and 7.8% improvements compared with the second-best semi-supervised method on ACC and NMI, respectively. The reason may be that, in this difficult data set, there are many difficult or unreliable data. Unsupervised methods, such as DEC and JULE, do not have any supervised information and thus these methods may be easily misled by those difficult or unreliable data. Although the semi-supervised methods, such as SDEC and DCC, use some supervised information, they do not pay any attention to selecting the supervised information and use randomly selected information, the improvement caused by the supervised information is limited. The proposed ADC is an active method, which can find key data, i.e., the ones which satisfy the uncertain and informative properties, for annotation. The carefully selected supervised information is more helpful for clustering. Noticed that, compared with our degenerated version ADC-random, whose constraints are selected randomly and may be useless for clustering, ADC achieves a better performance. This well demonstrates the motivation of our active schema.

To further show the effectiveness of the constraints active selection, we show the ACC and NMI performance of ADC and the semi-supervised deep methods with different numbers of constraints in Fig. 4. From Fig. 4, we find that the performance of ADC improves with the increase of the number of selection pairs. Moreover, when compared with other semi-supervised deep methods and ADC-random, ADC outperforms them most of the time, and ADC can achieve the best results of other methods with fewer constraints than they use. It shows that our constraints active selection strategy is more effective than the random selection, which demonstrates the superiority and necessity of the strategy.

Table 3 shows the running time of ADC and compared methods. From Table 3, we find that ADC is comparable with some mainstream semi-supervised methods.

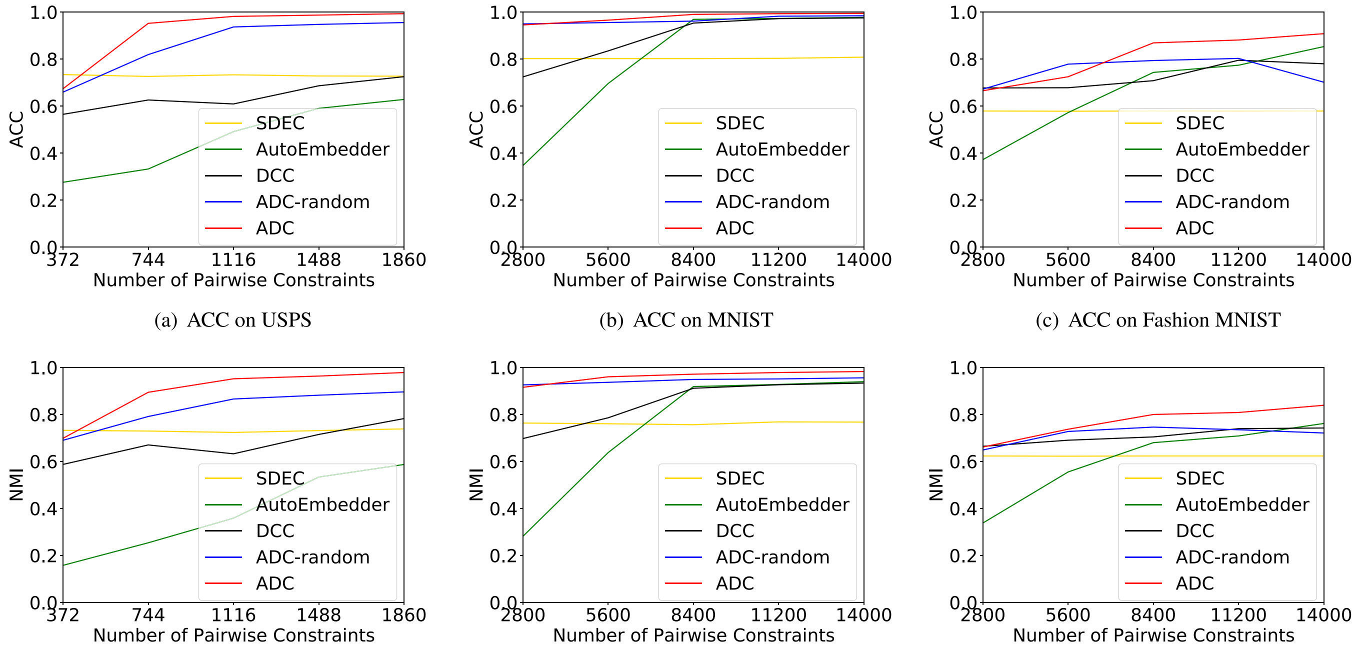
4.4. Ablation study

In Section 3.3, we propose a strategy to avoid checking whether a pair has been selected before. In this section, we compare the running time of ADC with the version that checks each pair (denoted as ADC-check) to see whether the strategy can reduce the time. Table 4 shows the comparison results. It can be seen that the strategy can indeed save time compared with ADC-check. Notice that, the only difference between ADC-check and ADC is whether to check the pairs or not, and the representation learning and clustering of the two methods remain the same. Therefore, space may still exist to further speed up the proposed method. In the future, we will study how to further reduce the time and space complexity.

Moreover, in our strategy, we select pairs based on the similarity to the cluster centers. One more natural strategy is to select the data pairs with the largest distance in the same cluster, as they should be the most uncertain pairs. We denote this strategy as ADC-Uncertain and compare ADC with it on the benchmark data sets. The comparison results are shown in Table 5. It can be seen that our strategy is better than selecting the most uncertain pairs. In our active clustering framework, we wish to select the data pairs which are *uncertain* and *informative*. ADC-Uncertain selects pairs with the largest distance in the same cluster, which are the most uncertain pairs. However, it has a large probability that the selected pairs by ADC-Uncertain are cannot-link pairs. As introduced before, must-link constraints are much more informative than cannot-link constraints and thus the excessive cannot-link constraints do not have too much value for clustering. Therefore, our strategy based on the distance to the cluster center is more appropriate for semi-supervised clustering. It is a better trade-off between uncertain property and informative property.

4.5. Visualization results

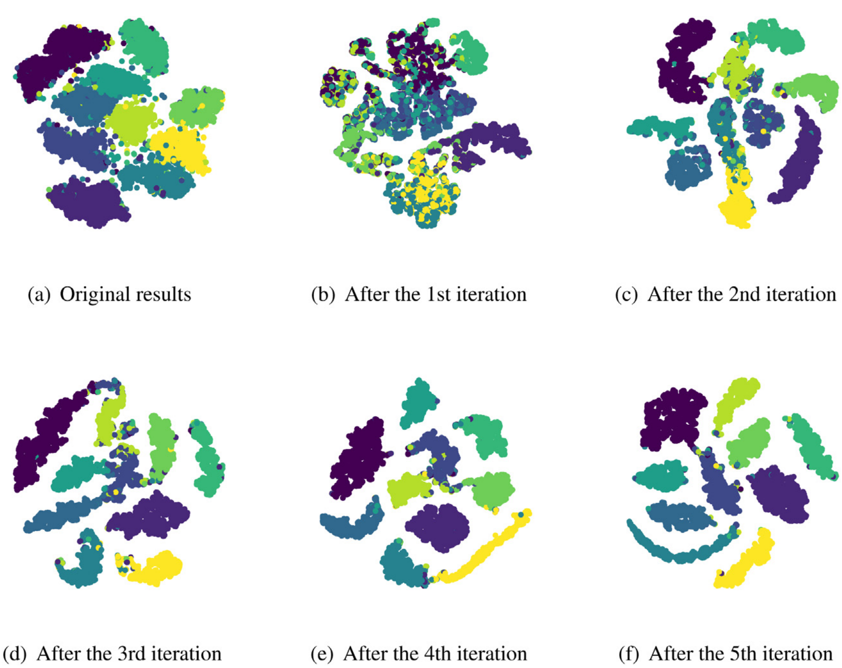
To show the effectiveness of the learned representation of our backbone network, we show the t-SNE [62] visualization results



(a) ACC on USPS (b) ACC on MNIST (c) ACC on Fashion MNIST

(d) NMI on USPS (e) NMI on MNIST (f) NMI on Fashion MNIST

Fig. 4. Clustering results of ADC and the semi-supervised deep methods on different numbers of selected pairs.



(a) Original results (b) After the 1st iteration (c) After the 2nd iteration

(d) After the 3rd iteration (e) After the 4th iteration (f) After the 5th iteration

Fig. 5. t-SNE results on USPS with each iteration.

Table 5
Ablation Study: comparison with ADC-Uncertain.

Methods	USPS			MNIST			Fashion MNIST		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
ADC-Uncertain	0.962	0.903	0.921	0.978	0.942	0.953	0.789	0.732	0.662
ADC	**0.988**	**0.965**	**0.984**	**0.996**	**0.987**	**0.993**	**0.880**	**0.820**	**0.785**

of the original data and the representation learned from each iteration. Fig. 5 shows the t-SNE visualization results of USPS data set. The results on other data sets are similar. From Fig. 5, we find that, after each iteration, the learned representation displays an increasingly clearer clustering structure with the iteration, which demonstrates that the backbone network together with the

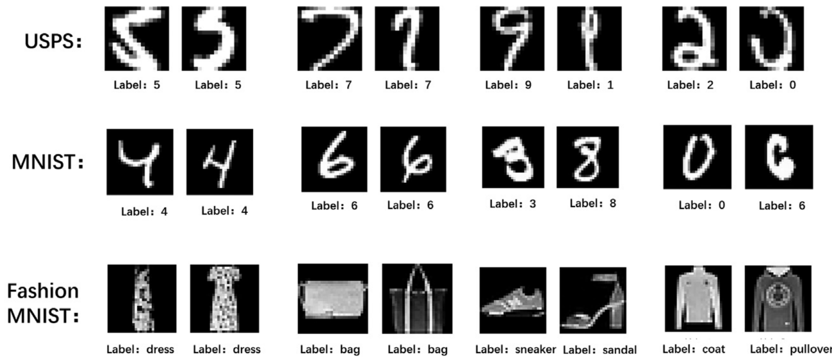


Fig. 6. Some example pairs ADC selects for querying human annotation.

Table 6
Clustering results on STL-10.

Metrics	kmeans	SC	DEC	JULE	IDEC	SCAN	RUC	SDEC	AutoEmbedder	DCC	ADC	ADC-Res
ACC	0.192	0.159	0.359[a]	0.277[a]	0.370[a]	0.767[a]	**0.866**[a]	0.389[a]	0.376	0.184	0.603	0.651
NMI	0.125	0.098	0.276[a]	0.182[a]	0.325[a]	**0.680**[a]	–	0.328[a]	0.250	0.106	0.522	0.581
ARI	0.061	0.048	0.186[a]	0.164[a]	0.189[a]	**0.616**[a]	–	0.206[a]	0.179	0.104	0.426	0.463

[a]denotes the results are those reported in their papers and other results are the reproduced results by us with the released source codes.

human annotation can indeed lead to a better representation for clustering.

Fig. 6 shows the example pairs selected by our ADC on the three data sets. On each data set, we show 2 must-link pairs (the first 2 pairs in each row) and 2 cannot-link pairs (the last 2 pairs in each row) in Fig. 6, respectively. We can find that, in the must-link pairs, the two images often look very different, and in the cannot-link pairs, the two images often look similar. Therefore, these pairs are the difficult ones which may mislead the clustering algorithm and if we obtain the human annotations of these pairs, the annotations can help the clustering algorithm to handle them correctly. This is in line with our motivation for active clustering, i.e., we apply the human annotations of some difficult or unreliable data to alleviate their side-effect and improve the performance of clustering.

4.6. Experiments on STL-10 data set

We also conduct experiments on a more difficult data set STL-10 [63]. STL-10 data set contains 13000 high-resolution color images which are in 10 categories. The size of each image is 96×96. Table 6 shows the clustering results of compared unsupervised and semi-supervised methods and our ADC on STL-10 data set. Since some methods (e.g. RUC and SCAN) use ResNet [64] as their backbones, we also replace the CNN in ADC with ResNet, which is denoted as ADC-Res.

From Table 6, we can find that ADC outperforms some conventional unsupervised and semi-supervised methods. Compared with the original ADC, ADC-Res improves the performance. However, it is not better than the state-of-the-art deep clustering methods SCAN and RUC. Their methods apply pseudo-labels to guide the clustering. Since pseudo-labels do not cost manpower, they can generate much more pseudo-labels for clustering. Therefore, in the future, we can further consider how to combine the few human annotations and a lot of pseudo-labels in active deep clustering. On one hand, a few human annotations can provide more precise supervised information; and on the other hand, a large number of pseudo-labels can propagate the supervised information to as many unlabeled data as possible.

5. Conclusion

This paper proposed a novel active deep clustering method for image clustering. To improve the reliability of the clustering result, in our method, we actively selected the uncertain and informative data for querying human annotations. After obtaining the annotations, which were in the form of must-link and cannot-link constraints, we applied them to train a Siamese network to learn the representations which can preserve such constraints. Moreover, we also adopted a clustering loss to make sure that representations had a clear clustering structure. We integrated the representation learning, clustering, and the constraints active selection into a unified framework to keep them being boosted by each other. The extensive experiments shew that the proposed active method outperformed the state-of-the-art deep clustering methods and semi-supervised methods which demonstrated the effectiveness and superiority of the proposed method.

Although the proposed method is simple yet effective, there still exist many other criteria in active learning besides the informative and uncertainty. In the future, to tackle some specific tasks, we can try some other criteria in our active selection strategy. For example, we can select the pairs which are more easily to propagate the label information to other unlabeled data, such that we can use as few annotations as possible to achieve a relatively good clustering performance.

CRediT authorship contribution statement

Bicheng Sun: Methodology, Software, Writing – original draft. **Peng Zhou:** Conceptualization, Methodology, Writing – original draft, Supervision. **Liang Du:** Methodology, Writing – review & editing. **Xuejun Li:** Writing – review & editing, Funding acquisition.

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 supported by the National Natural Science Foundation of China grants 62176001, 61806003, 61976129, and 6197 2001.

References

[1] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1, (14) Oakland, CA, USA, 1967, pp. 281–297.
[2] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in Neural Information Processing Systems, 2002, pp. 849–856.
[3] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, SIGKDD Explor. 6 (1) (2004) 90–105.
[4] J. Xie, R.B. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: M. Balcan, K.Q. Weinberger (Eds.), Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, in: JMLR Workshop and Conference Proceedings, vol.48, JMLR.org, 2016, pp. 478–487.
[5] J. Yang, D. Parikh, D. Batra, Joint unsupervised learning of deep representations and image clusters, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, IEEE Computer Society, 2016, pp. 5147–5156.
[6] X. Guo, L. Gao, X. Liu, J. Yin, Improved deep embedded clustering with local structure preservation, in: C. Sierra (Ed.), Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, ijcai.org, 2017, pp. 1753–1759.
[7] Y. Yan, H. Hao, B. Xu, J. Zhao, F. Shen, Image clustering via deep embedded dimensionality reduction and probability-based triplet loss, IEEE Trans. Image Process 29 (2020) 5652–5661.
[8] X. Peng, J. Feng, J.T. Zhou, Y. Lei, S. Yan, Deep subspace clustering, IEEE Trans. Neural Netw. Learn. Syst. 31 (12) (2020) 5509–5521.
[9] S. Huang, Z. Kang, Z. Xu, Q. Liu, Robust deep k-means: An effective and simple method for data clustering, Pattern Recognit. 117 (2021) 107996.
[10] Y. Ren, K. Hu, X. Dai, L. Pan, S.C.H. Hoi, Z. Xu, Semi-supervised deep embedded clustering, Neurocomputing 325 (2019) 121–130.
[11] S. Fogel, H. Averbuch-Elor, D. Cohen-Or, J. Goldberger, Clustering-driven deep embedding with pairwise constraints, IEEE Comput. Graph. Appl. 39 (4) (2019) 16–27.
[12] A. Shukla, G.S. Cheema, S. Anand, Semi-supervised clustering with neural networks, in: 6th IEEE International Conference on Multimedia Big Data, BigMM 2020, New Delhi, India, September 24-26, 2020, IEEE, 2020, pp. 152–161.
[13] L.A. Vilhagra, E.R. Fernandes, B.M. aes Nogueira, TextCSN: a semi-supervised approach for text clustering using pairwise constraints and convolutional siamese network, in: C. Hung, T. Cerný, D. Shin, A. Bechini (Eds.), SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing, Online Event, [Brno, Czech Republic], March 30 - April 3, 2020, ACM, 2020, pp. 1135–1142.
[14] H. Zhang, T. Zhan, S. Basu, I. Davidson, A framework for deep constrained clustering, Data Min. Knowl. Discov. 35 (2) (2021) 593–620.
[15] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, 20-26 June 2005, San Diego, CA, USA, IEEE Computer Society, 2005, pp. 539–546.
[16] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[17] P. Zhou, Y. Shen, L. Du, F. Ye, X. Li, Incremental multi-view spectral clustering, Knowl. Based Syst. 174 (2019) 73–86.
[18] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, W. Gao, Late fusion incomplete multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. 41 (10) (2019) 2410–2423.
[19] P. Zhou, L. Du, X. Li, Self-paced consensus clustering with bipartite graph, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, ijcai.org, 2020, pp. 2133–2139.
[20] P. Zhou, J. Chen, L. Du, X. Li, Balanced spectral feature selection, IEEE Trans. Cybern. (2022) 1–13, http://dx.doi.org/10.1109/TCYB.2022.3160244.
[21] P. Zhou, L. Du, X. Liu, Y. Shen, M. Fan, X. Li, Self-paced clustering ensemble, IEEE Trans. Neural Netw. Learn. Syst. 32 (4) (2021) 1497–1511.
[22] Z. Kang, C. Peng, Q. Cheng, X. Liu, X. Peng, Z. Xu, L. Tian, Structured graph learning for clustering and semi-supervised classification, Pattern Recognit. 110 (2021) 107627.
[23] Z. Huang, J.T. Zhou, H. Zhu, C. Zhang, J. Lv, X. Peng, Deep spectral representation learning from multi-view data, IEEE Trans. Image Process. 30 (2021) 5352–5362.
[24] P. Zhou, X. Wang, L. Du, X. Li, Clustering ensemble via structured hypergraph learning, Inf. Fusion 78 (2022) 171–179.
[25] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[26] Q. Yang, H. Wang, T. Li, Y. Yang, Deep belief networks oriented clustering, in: 2015 10th International Conference on Intelligent Systems and Knowledge Engineering, ISKE, IEEE, 2015, pp. 58–65.
[27] B. Yang, X. Fu, N.D. Sidiropoulos, M. Hong, Towards K-means-friendly spaces: Simultaneous deep learning and clustering, in: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, in: Proceedings of Machine Learning Research, vol.70, PMLR, 2017, pp. 3861–3870.
[28] P. Ji, T. Zhang, H. Li, M. Salzmann, I.D. Reid, Deep subspace clustering networks, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 24–33.
[29] M.M. Fard, T. Thonet, E. Gaussier, Deep k-means: Jointly clustering with k-means and learning representations, Pattern Recognit. Lett. 138 (2020) 185–192.
[30] J. Lv, Z. Kang, X. Lu, Z. Xu, Pseudo-supervised deep subspace clustering, IEEE Trans. Image Process. 30 (2021) 5252–5263.
[31] F. Li, H. Qiao, B. Zhang, Discriminatively boosted image clustering with fully convolutional auto-encoders, Pattern Recognit. 83 (2018) 161–173.
[32] J. Guérin, B. Boots, Improving image clustering with multiple pretrained CNN feature extractors, in: British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018, BMVA Press, 2018, p. 51.
[33] W. Lin, J. Chen, C.D. Castillo, R. Chellappa, Deep density clustering of unconstrained faces, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 8128–8137.
[34] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV, in: Lecture Notes in Computer Science, vol.11218, Springer, 2018, pp. 139–156.
[35] Y. Li, P. Hu, J.Z. Liu, D. Peng, J.T. Zhou, X. Peng, Contrastive clustering, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, AAAI Press, 2021, pp. 8547–8555.
[36] C. Liu, Z. Liao, Y. Ma, K. Zhan, Stationary diffusion state neural estimation for multiview clustering, in: AAAI, 2022.
[37] N. Dilokthanakul, P.A. Mediano, M. Garnelo, M.C. Lee, H. Salimbeni, K. Arulkumaran, M. Shanahan, Deep unsupervised clustering with gaussian mixture variational autoencoders, 2016, arXiv preprint arXiv:1611.02648.
[38] Z. Jiang, Y. Zheng, H. Tan, B. Tang, H. Zhou, Variational deep embedding: An unsupervised and generative approach to clustering, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, ijcai.org, 2017, pp. 1965–1972.
[39] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets, in: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016, pp. 2172–2180.
[40] Y. Yu, W. Zhou, Mixture of GANs for clustering, in: J. Lang (Ed.), Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ijcai.org, 2018, pp. 3047–3053.
[41] P. Zhou, Y. Hou, J. Feng, Deep adversarial subspace clustering, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 1596–1604.
[42] S. Mukherjee, H. Asnani, E. Lin, S. Kannan, ClusterGAN: Latent space clustering in generative adversarial networks, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, AAAI Press, 2019, pp. 4610–4617.
[43] W.V. Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, L.V. Gool, SCAN: learning to classify images without labels, in: Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part X, in: Lecture Notes in Computer Science, vol.12355, Springer, 2020, pp. 268–285.
[44] S. Park, S. Han, S. Kim, D. Kim, S. Park, S. Hong, M. Cha, Improving unsupervised image clustering with robust learning, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, June 19-25, 2021, Computer Vision Foundation / IEEE, 2021, pp. 12278–12287.
[45] S. Fogel, H. Averbuch-Elor, D. Cohen-Or, J. Goldberger, Clustering-driven deep embedding with pairwise constraints, IEEE Comput. Grap. Appl. 39 (4) (2019) 16–27.

[46] H. Zhang, S. Basu, I. Davidson, A framework for deep constrained clustering - algorithms and advances, in: U. Brefeld, E. Fromont, A. Hotho, A.J. Knobbe, M.H. Maathuis, C. Robardet (Eds.), Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, WÜRzburg, Germany, September 16-20, 2019, Proceedings, Part I, in: Lecture Notes in Computer Science, vol.11906, Springer, 2019, pp. 57–72.

[47] A.Q. Ohi, M.F. Mridha, F.B. Safir, M.A. Hamid, M.M. Monowar, AutoEmbedder: A semi-supervised DNN embedding system for clustering, Knowl. Based Syst. 204 (2020) 106190.

[48] B. Settles, Active learning literature survey, Tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.

[49] S.C. Hoi, R. Jin, J. Zhu, M.R. Lyu, Batch mode active learning and its application to medical image classification, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 417–424.

[50] R. Chattopadhyay, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, J. Ye, Batch mode active sampling based on marginal probability distribution matching, ACM Trans. Knowl. Discov. Data (TKDD) 7 (3) (2013) 13.

[51] H. Wang, L. Du, P. Zhou, L. Shi, Y.-D. Shen, Convex batch mode active sampling via α-relative pearson divergence, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 3045–3051.

[52] H. Wang, L. Du, P. Zhou, L. Shi, Y. Qian, Y. Shen, Experimental design with multiple kernels, in: 2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14-17, 2015, IEEE Computer Society, 2015, pp. 419–428.

[53] Y. Yan, S.-J. Huang, Cost-effective active learning for hierarchical multi-label classification, in: IJCAI, 2018, pp. 2962–2968.

[54] H. Wang, R. Zhou, Y.-D. Shen, Bounding uncertainty for active batch selection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 5240–5247.

[55] C. Li, H. Ma, Z. Kang, Y. Yuan, X. Zhang, G. Wang, On deep unsupervised active learning, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, ijcai.org, 2020, pp. 2626–2632.

[56] E.P. Xing, A.Y. Ng, M.I. Jordan, S.J. Russell, Distance metric learning with application to clustering with side-information, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems 15 Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada, MIT Press, 2002, pp. 505–512.

[57] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[58] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017, CoRR abs/1708.07747, arXiv:1708.07747.

[59] X. Yang, C. Deng, F. Zheng, J. Yan, W. Liu, Deep spectral clustering using dual autoencoder network, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 4066–4075.

[60] X. Guo, X. Liu, E. Zhu, X. Zhu, M. Li, X. Xu, J. Yin, Adaptive self-paced deep clustering with data augmentation, IEEE Trans. Knowl. Data Eng. 32 (9) (2020) 1680–1693.

[61] L. Wu, Z. Liu, Z. Zang, J. Xia, S. Li, S.Z. Li, Generalized clustering and multi-manifold learning with geometric structure preservation, 2021, CoRR abs/2009.09590, arXiv:2009.09590.

[62] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (86) (2008) 2579–2605.

[63] A. Coates, A.Y. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: G.J. Gordon, D.B. Dunson, M. Dudík (Eds.), Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011, in: JMLR Proceedings, vol.15, JMLR.org, 2011, pp. 215–223.

[64] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, IEEE Computer Society, 2016, pp. 770–778.