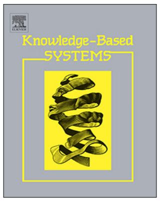
Knowledge-Based Systems 197 (2020) 105841

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



Deep density-based image clustering

Yazhou Ren [a,b,*], Ni Wang [a], Mingxia Li [a], Zenglin Xu [c]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[b] Institute of Electronic and Information Engineering of UESTC in Guangdong, Dongguan 523808, China
[c] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

ARTICLE INFO

Article history:
Received 11 August 2019
Received in revised form 4 January 2020
Accepted 29 March 2020
Available online 11 April 2020

Keywords:
Deep clustering
Density-based clustering
Feature learning

ABSTRACT

Recently, deep clustering, which is able to perform feature learning that favors clustering tasks via deep neural networks, has achieved remarkable performance in image clustering applications. However, the existing deep clustering algorithms generally need the number of clusters in advance, which is usually unknown in real-world tasks. In addition, the initial cluster centers in the learned feature space are generated by *k*-means. This only works well on spherical clusters and probably leads to unstable clustering results. In this paper, we propose a two-stage deep density-based image clustering (DDC) framework to address these issues. The first stage is to train a deep convolutional autoencoder (CAE) to extract low-dimensional feature representations from high-dimensional image data, and then apply t-SNE to further reduce the data to a 2-dimensional space favoring density-based clustering algorithms. In the second stage, we propose a novel density-based clustering technique for the 2-dimensional embedded data to automatically recognize an appropriate number of clusters with arbitrary shapes. Concretely, a number of local clusters are generated to capture the local structures of clusters, and then are merged via their density relationship to form the final clustering result. Experiments demonstrate that the proposed DDC achieves comparable or even better clustering performance than state-of-the-art deep clustering methods, even though the number of clusters is not given.

1. Introduction

Image clustering is one of the extensively exploited topics in machine learning and has many applications in a wide range of fields, including image retrieval [1,2] and annotation [3]. It seeks to partition images into clusters according to a similarity measure, such that similar images are grouped in the same cluster and images which are dissimilar from each other are grouped into different clusters. A number of traditional clustering methods have been proposed in the past decades, such as partitional clustering [4,5], hierarchical clustering [6], density-based clustering [7–11], distribution-based clustering [12], clustering based on non-negative matrix factorization (NMF) [13–15], etc. These methods usually fail to clustering image data sets which are with high dimensionality. The main reason is that reliable similarity measures are hard to obtain in the high dimensional space.

To mitigate this issue, a normal method is to first reduce the dimensionality of data via feature selection or feature extraction techniques, and then conduct clustering in the lower dimensional space. Another way is to consider clustering and feature learning together in the clustering framework, such as Torre et al. performs *k*-means clustering and linear discriminant analysis jointly [16]. However, these shallow models are typically with limited representation power and thus their improvement on image clustering performance is not significant.

Recently, deep clustering methods, which perform feature learning by applying deep neural networks (DNN) and conduct clustering in the latent learned feature space, have shown impressive performance in image clustering tasks and have attracted people's increasing attentions [17–28]. Despite the huge success, most of the existing deep clustering methods actually apply a partitional clustering, e.g., *k*-means clustering in the latent learned feature space. This brings the following drawbacks: (1) The number of clusters must be given in advance, which is usually unknown in practical clustering tasks. (2) The partitional clustering techniques can only find spherical clusters and perform worse on irregular clusters or imbalanced data. (3) The *k*-means like clustering methods have randomness, probably leading to unstable clustering results.

Some methods have been proposed to estimate the number of clusters in deep clustering models [29–31]. However, these methods do not consider the local information of clusters, and do not consider that points with different densities should play different roles in density-based clustering technique. Thus, the performance of these methods is still not satisfied and two questions

[*] Corresponding author at: School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.
E-mail address: yazhou.ren@uestc.edu.cn (Y. Ren).

https://doi.org/10.1016/j.knosys.2020.105841
0950-7051/© 2020 Elsevier B.V. All rights reserved.

are normally raising: *(1) How deep clustering methods effectively find appropriate number of clusters with irregular shape when the number of clusters is not known a-prior? (2) Do we really need to refine the deep neural networks with the initial cluster assignment?*

In this paper, we aim to answer these two questions and propose a novel effective deep density-based clustering (DDC) method for images. Specifically, DDC first learns deep feature representation of data via a deep autoencoder. Second, t-SNE [32] is adopted to further reduce the learned features to a 2-dimensional space while preserving the pairwise similarity of data instances. Finally, we develop a novel density-based clustering method which considers both the local structures of clusters and importance of instances to generate the final clustering results. The source code of the proposed DDC is available at https://github.com/Yazhou-Ren/DDC.

The contributions of this work are stated as below:

- We propose an effective density-based technique for deep clustering which can automatically find appropriate number of image clusters with arbitrary shapes. We first reduce the original data to a 2-dimensional space and then develop a novel density-based clustering method for the learned data.
- DDC is with good cluster visualization and interpretability. Its properties are theoretically and empirically analyzed. Its efficiency and robustness to parameter setting are also empirically verified.
- Extensive experiments are conducted to show that DDC becomes the new state-of-the-art deep clustering method on various image clusters discovering tasks when the number of clusters is unknown.

2. Related work

2.1. Deep clustering

Due to the good representation ability, deep neural networks (DNN) have gained impressive achievements in various types of machine learning and computer vision applications [33–35]. Most of the DNN methods focus on supervised problems in which the label information is known. In recent several years, people pay increasing attentions to adopting DNN in unsupervised learning tasks and a number of deep clustering methods have been proposed.

One kind of deep clustering methods divide the clustering procedure into two stages, i.e., feature learning and clustering. They first perform feature learning via DNN and then apply clustering algorithms in the learned space [19,20,36,37]. The other kind of deep clustering methods incorporate the abovementioned two stages into one framework. Song et al. [38] refine the autoencoder such that data representations in the learned space are close to their affiliated cluster centers. Xie et al. [21] propose deep embedded clustering (DEC) to jointly learn the cluster assignment and the feature representations. Ren et al. [39] propose semi-supervised deep embedded clustering to enhance the performance of DEC by using pairwise constraints. Yang et al. [23] and Chang et al. [17] apply convolutional neural networks (CNN) for exploring image clusters. Guo et al. [40] improve DEC with local structure preservation. Guo et al. [18] use data augmentation in the DEC framework and achieve state-of-the-art clustering performance on several image data sets.

2.2. Density-based clustering

The key advantage of density-based clustering is that the number of clusters is not needed and clusters with arbitrary shape can be found. Over the past decades, many density-based clustering methods have been developed. DBSCAN [7] defines a cluster with points from continuous high-density regions and treats those points in low-density regions as outliers or noises. Inspired by this popular algorithm, a lot of density-based clustering methods have been designed, such as OPTICS [41], DENCLUE [42], DESCRY [43], and others [44–47]. DenPeak (clustering by fast search and find of density peaks) [48] is another immensely popular density-based clustering method, which assumes that cluster centers locate in regions with higher density and the distances among different centers should be relatively large. Some improvements of DenPeak have also been made [49–51]. These methods described above are applied in the original feature space. Thus, their performance for grouping images which are with high dimensionality is not satisfied due to the limited representation ability.

Most recently, several deep clustering methods [29–31] which seek to address the issue of estimating the number of clusters have been proposed, i.e., DDC-UF (deep density clustering of unconstrained faces) [29], DCC (deep continuous clustering) [30], and DED (deep embedding determination) [31]. However, these methods ignore the local structures in each cluster, and do not allow points to play different roles according to their densities. By contrast, the proposed DDC takes into both the local information of clusters and importance of points account and achieves significant improvements on clustering performance.

3. Deep density-based image clustering

This section presents the proposed deep density-based image clustering (DDC) in detail. Let $\mathcal{X} = \{x_i \in \mathbb{R}^D\}_{i=1}^n$ denote the image data set, where n is number of data points and D is the dimensionality. DDC aims at grouping $\mathcal{X}$ into an appropriate number of disjoint clusters without any prior knowledge such as the number of clusters and label information. DDC is a two-stage deep clustering model which contains two main steps, i.e., deep feature learning which nonlinearly transfers the original features to a low dimensional space, and density-based clustering which automatically recognizes an appropriate number of clusters with shapes in the latent space.

3.1. Deep feature learning

As deep clustering methods generally do, we adopt deep autoencoder to initialize the feature transformation due to its excellent representation ability. An autoencoder is consisted of two parts: the encoder $h = f_\Theta(x)$ (maps each data point x to a learned representation h) and the decoder $x' = g_\Omega(h)$ (transfers data from the learned feature space to the original one). Here, the feature dimensionality of h is d. Θ and Ω denote the parameters of the encoder and decoder, respectively. In this paper, we use the denoising autoencoder [52] that solves the following problem:

$$\arg\min_{\Theta,\Omega} \frac{1}{n}\sum_{i=1}^{n} \|x_i - g_\Omega(f_\Theta(\tilde{x}_i))\|_2^2 \tag{1}$$

where $\tilde{x}$ is a corrupted copy of x by adding noises, e.g., adding Gaussian noise or randomly setting a portion of input data to 0. We use the stacked autoencoder (SAE) [53] in this work, in which each layer is a denoising autoencoder trained to reconstruct the previous layer's output. For image clustering, we adopt the deep convolutional autoencoder (CAE) in the experiments, whose structure will be stated in Section 3.3.

In [18], the data augmentation (DA) technique is used in the training process of deep autoencoder and has achieved significant improvements of clustering performance. The resulting optimization model is:

$$\arg\min_{\Theta,\Omega} \frac{1}{n}\sum_{i=1}^{n} \|\bar{x}_i - g_\Omega(f_\Theta(\bar{x}_i))\|_2^2 \tag{2}$$

where $\bar{x}_i = T_{rand}(x_i)$ denotes the random transformation[1] of x_i.

When the training of deep autoencoder (solving Eq. (1) or Eq. (2)) is finished, we observe the feature representations $\mathcal{H} = \{h_i = f_\Theta(x_i) \in \mathbb{R}^d\}_{i=1}^n$. For visualization and better fitting the designed density-based clustering algorithm, we further reduce data $\mathcal{H}$ to a 2-dimensional space $\mathcal{Z} = \{z_i \in \mathbb{R}^2\}_{i=1}^n$ by using t-SNE [32] which owns good preservation ability of pairwise similarities.

t-SNE is a dimensionality reduction method which can visualize high-dimensional data in a 2-dimensional space. Firstly, t-SNE defines the joint probability p_{ij} of data points h_i and h_j as:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n} \tag{3}$$

where

$$p_{j|i} = \frac{\exp(-\|h_i - h_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|h_i - h_k\|^2/2\sigma_i^2)} \tag{4}$$

Here, σ_i is a parameter for h_i. Secondly, the joint probability q_{ij} of z_i and z_j in the learned 2-dimensional space is calculated as:

$$q_{ij} = \frac{(1 + \|z_i - z_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|z_k - z_l\|^2)^{-1}} \tag{5}$$

Both p_{ii} and q_{ii} are set to 0. Then, t-SNE seeks to minimize the Kullback–Leibler divergence between the two joint probability distributions P and Q:

$$KL(P \parallel Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{6}$$

When all the 2-dimensional data points $\{z_i \in \mathbb{R}^2\}_{i=1}^n$ are obtained, we develop a novel density-based clustering in the embedded space $\mathcal{Z}$ as below.

3.2. Density-based clustering

We propose a novel density-based clustering method to obtain an appropriate partition of data $\mathcal{Z} = \{z_i \in \mathbb{R}^2\}_{i=1}^n$ in the 2-dimensional feature space when the number of clusters is unavailable.

3.2.1. Local clusters generation

DDC shares two fundamental definitions (i.e., ρ_i and δ_i of point z_i) with DenPeak [48]. Concretely, DDC defines the density of ρ_i of point z_i via a Gaussian kernel:

$$\rho_i = \sum_{z_j \in \mathcal{Z} \setminus \{z_i\}} \exp\left(-(\frac{d_{ij}}{d_c})^2\right) \tag{7}$$

where d_{ij} is the Euclidean distance between points z_i and z_j, and d_c is the cutoff distance that need to be predefined. A higher value of ρ_i means a higher density of point z_i. δ_i of point z_i denotes the minimum Euclidean distance between z_i and those points whose densities are larger than z_i. That is,

$$\delta_i = \min_{j:\rho_j > \rho_i}(d_{ij}) \tag{8}$$

For the point with the highest density, its ρ is set to the maximum of pairwise distances. DenPeak simply chooses several points with the highest ρ and δ values as cluster centers. Different from DenPeak, we consider those points with relatively large ρ and δ values as local cluster centers. The corresponding definition is given in Definition 1.

Definition 1 (*Local Cluster Centers*)**.**Those points satisfying the following condition are defined as local cluster centers:

$$\delta_i > d_c \quad and \quad \rho_j > \bar{\rho} \tag{9}$$

where $\bar{\rho} = \frac{1}{n}\sum_{j=1}^n \rho_j$ is the average density of all the points $\{z_i\}_{i=1}^n$.

It is easy to verify that a local cluster center z_i owns the largest density in its d_c-neighborhood, i.e., a circle with z_i and d_c as the center and radius, respectively. When all the local cluster centers are obtained, we assign each remaining point to the cluster as its nearest neighbor of higher density. Then, a set of local clusters are found and will be used to generate the final clustering. To analyze the characteristic of local cluster centers, the following two theorems are stated.

Theorem 1. *A local cluster center z_i owns the largest density value ρ_i locally in its d_c-neighborhood.*

Proof. We use 'proof by contradiction' method to prove the theorem. For a local cluster center z_i, assume that there exists a point z_j in the d_c-neighborhood of z_i satisfying $\rho_j > \rho_i$. Then, $\delta_i \leq d_c$ holds according to Eq. (8). This actually contradicts Eq. (9) in Definition 1. Thus, the assumption is wrong and the theorem is proved. □

Theorem 2. *The distance of two local cluster centers with different densities is at least d_c.*

Proof. Suppose z_i and z_j are two local cluster centers with $\rho_i \neq \rho_j$. We assume the distance $d_{ij} < d_c$, then z_i and z_j are in the d_c-neighborhoods of each other. Since z_i is a local cluster center, it owns the highest density in its d_c-neighborhood. Thus, $\rho_i \geq \rho_j$. z_j is also a local cluster center. Similarly, we have $\rho_j \geq \rho_i$. Thus, $\rho_i = \rho_j$. This contradicts the condition of the theorem. □

Thus, the distance of two local clusters is smaller than d_c only when they have the same density and Eq. (9) holds at the same time. In real tasks, this situation extremely rarely occurs. As a consequence, Theorems 1 and 2 indicate two important properties of local cluster centers: (1) Each local center is with the highest density locally. (2) The selected cluster centers are not too close to each other, preventing a huge number of cluster centers from being selected.

3.2.2. Merging local clusters

Suppose L local clusters $(\mathcal{C}^{(1)}, \mathcal{C}^{(2)}, \ldots, \mathcal{C}^{(L)})$ are obtained, they will be merged to form the final clustering result. First, we define core and border points in Definition 2.

Definition 2 (*Core and Border Points of a Cluster*)**.**Suppose a point z_i is from local cluster $\mathcal{C}^{(k)}$, it is defined as a core point if the following condition holds:

$$\rho_i > \bar{\rho}^{(k)} \tag{10}$$

where $\bar{\rho}^{(k)} = \frac{1}{n_k}\sum_{z_j \in \mathcal{C}^{(k)}} \rho_j$ is the average density of all the points in $\mathcal{C}^{(k)}$ and n_k is the number of points in $\mathcal{C}^{(k)}$. Otherwise, z_i is considered as a border point.

Definition 2 indicates that whether a point is a core or border point depends on its own density and the average density of the local cluster to which this point belongs. Generally, the core points of a cluster locate in the central regions, while the border points place in the boundary of areas with lower density.

Then, we define connectivity of clusters in Definitions 3 and 4.

[1] As in [18], we randomly shift for at most 3 pixels in each direction and randomly rotate for at most 10°.

Algorithm 1 Deep Density-based Image Clustering (DDC).

```
Input:  Image data set $\mathcal{X}$; Cutoff distance $d_c$.
Output: The final clustering result.
Stage 1 → Deep feature learning
Train a deep autoencoder via Eq. (1) or (2).
Transform $\mathcal{X}$ to lower feature representations $\mathcal{H}$ via the encoder $f_\Theta(\cdot)$.
Map $\mathcal{H}$ to a 2-dimensional data set $\mathcal{Z}$ via t-SNE.
Stage 2 → Density-based clustering
\\ Local clusters generation
for each point $z_i$ in $\mathcal{Z}$ do
    Compute $\rho_i$ and $\delta_i$ via Eqs. (7) and (8).
end for
Choose local cluster centers via Eq. (9).
Assign the remaining points and observe local clusters $\mathcal{C}^{(1)}, \mathcal{C}^{(2)}, \ldots, \mathcal{C}^{(L)}$.
\\ Generate the final clustering result
Define core and border points via Eq. (10).
Merge all the density connectable local clusters via Eqs. (11) and (12).
Return the final clustering result.
```

Definition 3 (*Density Directly-connectable of Clusters*)**.**A local cluster $\mathcal{C}^{(k)}$ is density directly-connectable from a local cluster $\mathcal{C}^{(l)}$ if:

$$\exists \text{ core points } z_i \in \mathcal{C}^{(k)} \text{ and } z_j \in \mathcal{C}^{(l)}, \text{ such that } d_{ij} < d_c. \tag{11}$$

Definition 4 (*Density Connectable of Clusters*)**.**A local cluster $\mathcal{C}^{(k)}$ is density-connectable to a local cluster $\mathcal{C}^{(l)}$ if:

$$\exists \text{ a path } \mathcal{C}^{(k)} = \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m = \mathcal{C}^{(l)} \tag{12}$$

where cluster $\mathcal{C}_j$ is density directly-connectable from cluster $\mathcal{C}_{j-1}$ ($j = 2, \ldots, m$) and m is the path length.

It is easy to verify that both density directly-connectable and density connectable are symmetric. Finally, all the density-connectable local clusters are merged and the final clustering result is provided. When two local clusters are merged, the cluster center with higher density becomes the center of the new merged cluster.

According to Definitions 3 and 4, two clusters are merged only when their central areas are very close to each other. This ensures the new merged cluster also has continuous high-density areas.

The pseudo-code of the proposed DDC is summarized in Algorithm 1.

3.3. Implementation

According to different optimization problems, DDC provides two specific algorithms:

(1) DDC: Use CAE and solve Eq. (1).
(2) DDC-DA: Use CAE and solve Eq. (2) in which data augmentation is adopted.

As in [18] and [31], the structure of CAE is always set to $\text{Conv}^5_{32} \to \text{Conv}^5_{64} \to \text{Conv}^3_{128} \to \text{Fc}_{10} \to \text{Conv}^3_{128} \to \text{Conv}^5_{64} \to \text{Conv}^5_{32}$. Here, Conv^5_{32} represents a convolutional layer with 32 filters and a 5 × 5 kernel. The stride is always set to 2. Fc_{10} denotes the full connected layer with 10 neurons. In convolutional autoencoders, all the internal layers except for the input, embedding, and output layers are activated by ReLU function. The structures of autoencoders also indicate that the dimensionality of learned representations $\mathcal{H}$ is 10.

Table 1
Image data sets used in the experiments.

Data set	# examples	# classes	Image size
MNIST	70000	10	28 × 28
MNIST-test	10000	10	28 × 28
USPS	9298	10	16 × 16
Fashion	10000	10	28 × 28
LetterA-J	10000	10	28 × 28

Given the embedded 2-dimensional data $\mathcal{Z}$, DDC has only one parameter (d_c) needed to be set. We set the value of d_c according to data $\mathcal{Z}$ itself. Concretely, we compute $\bar{d}$ as the average value of all pairwise distances in $\mathcal{Z}$. Then, set $d_c = \bar{d} \times ratio$. If *ratio* is extremely large, a small number of clusters will be found by DDC. If *ratio* is extremely small, a large number of clusters will be detected. However, we will empirically verify that DDC achieves stable performance in a wide range of *ratio*. The default value of *ratio* is 0.1.

3.4. Relations to exiting methods

DBSCAN [7] and DenPeak [48] are two worldwide popular density-based clustering methods. They are applied in the original feature space, while the proposed DDC works in the 2-dimensional embedded space. Besides, DBSCAN is sensitive to the parameters and tends to merge clusters with overlapping areas [7,9]. These shortcomings prevent its successful use in image clustering tasks.

DenPeak assumes that each cluster has only one center, leading to the following disadvantages: (1) In real applications, multiple centers/modes usually coexist in one cluster. Thus, DenPeak typically loses information of local structures of a cluster. (2) It is difficult for DenPeak to select a suitable number of clusters because usually a number of (which is much larger than the ground-truth number of clusters) points with high ρ and δ values can be considered as candidates of cluster centers. To address these issues, DDC firstly selects all the potential cluster centers to obtain the local clusters, and then aggregates all density connectable cluster to form the final clustering result. An illustration exhibiting the different behaviors of DenPeak and DDC is given in Figs. 1 and 2. Here, several data sets with irregular shapes of clusters are used, i.e., Twomoon, Flame, and t4.[2] DCC is directly applied on the 2-dimensional data without using CAE and t-SNE. The *ratio* of DDC is 0.1. DenPeak follows the parameter setting described in Section 4.3.

DED [31] is a recently proposed deep clustering model that transforms the original data via DNN to a 2-dimensional feature space that favors the density-based clustering algorithm. However, DED directly applies DenPeak on the 2-dimensional data, thereby inheriting the disadvantages of DenPeak.

4. Experimental setup

This section describes the tested image data sets, comparing methods, parameter settings, and evaluation measures.

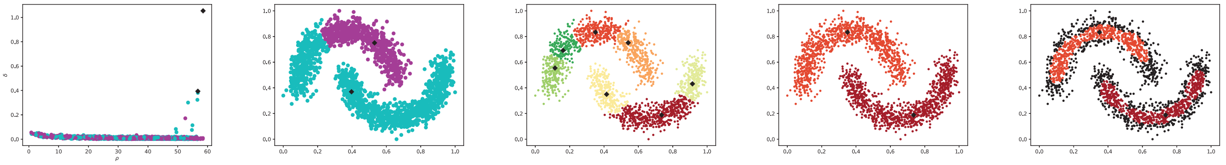
4.1. Image data sets

Five popular image data sets are used to assess the performance of comparing methods.

The MNIST data base[3] consists of 70000 handwritten digits of 28 × 28 pixel size from 10 categories (digits 0–9). The MNIST-test data set only contains the test set of MNIST, with 10000

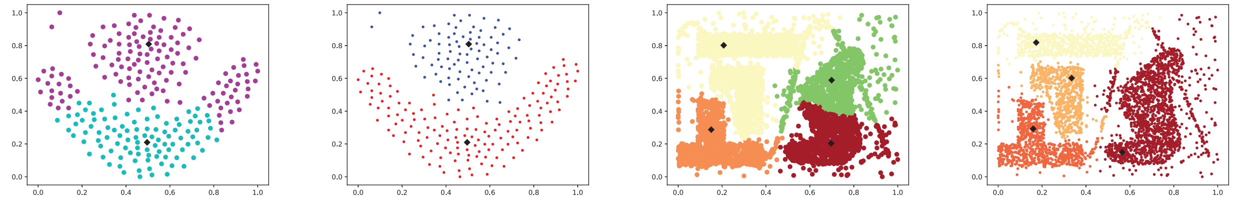
[2] http://cs.joensuu.fi/sipu/datasets/

[3] http://yann.lecun.com/exdb/mnist/



(a) Decision graph (b) Result of DenPeak (c) Initial result (d) Final result (e) Border points

Fig. 1. Twomoon: Clustering performance comparison of DenPeak and DDC. The Twomoon data set has 2000 points from two classes. (a): The decision graph of DenPeak. (b): The final result of DenPeak. (c): Initial local clusters of DDC. (d): The final result of DDC. (e): The border points detected by DDC are plotted as black points. The center of each cluster is highlighted with black '♦'. Points with the same color are from the same cluster. As shown in (a), a number of points with high ρ and δ values can be considered as centers and it is hard for DenPeak to choose an appropriate number of clusters. Even it is told that 2 clusters exist, the result of DenPeak is still not satisfied, as (b) shows. By contrast, DDC first generate a relatively large number of local cluster centers and then merge them to form the final clustering result. Compared (c) with (e), we find that two clusters are typically merged if there exists core points that are from both clusters and are close to each other. It is shown in (e) that border points generally locate around the boundary of each real cluster, while core points locate in central areas.



(a) Result of DenPeak (b) Result of DDC (c) Result of DenPeak (d) Result of DDC

Fig. 2. Clustering results of DenPeak and DDC on Flame and t4 data sets. (a) and (b) correspond to the Flame data set. (c) and (d) show the results on t4. DenPeak is told to select the true number of clusters. Due to loss information of local structures, DenPeak fails to find suitable clusters (as shown in (a) and (c)). In contrast, DDC performs perfectly on these two data sets. Even when noisy data exist (as exhibited in (d)), DDC can still automatically recognize the 4 irregular clusters.

Table 2
Results of the comparing methods. In each column, the best two results are highlighted in boldface. The results marked by '*' are excerpted from the papers. '–' denotes the results are unavailable from the papers or codes, and '- -' means 'out of memory' when applying.

	MNIST		MNIST-test		USPS		Fashion		LetterA-J	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
k-means	0.485	0.470	0.563	0.510	0.611	0.607	0.554	0.512	0.354	0.309
DBSCAN	- -	- -	0.114	0	0.167	0	0.100	0	0.100	0
DenPeak	- -	- -	0.357	0.399	0.390	0.433	0.344	0.398	0.300	0.211
DEC	0.849	0.816	0.856	0.830	0.758	0.769	0.591	0.618	0.407	0.374
IDEC	0.881*	0.867*	0.846	0.802	0.759	0.777	0.523	0.600	0.381	0.318
DCN	0.830*	0.810*	0.802*	0.786*	0.688*	0.683*	–	–	–	–
JULE	0.964*	0.913*	0.961*	0.915*	0.950*	0.913*	–	–	–	–
DEPICT	0.965*	0.917*	0.963*	0.915*	0.964*	0.927*	–	–	–	–
ClusterGAN	0.950*	0.890*	–	–	–	–	–	–	–	–
DWSC	0.948*	0.889*	–	–	–	–	–	–	–	–
DKM	0.840*	0.796*	–	–	0.757*	0.776*	–	–	–	–
VaDE	0.945*	0.876*	0.287*	0.287*	0.566*	0.512*	–	–	–	–
DCC	0.963*	–	–	–	–	–	–	–	–	–
DED	- -	- -	0.690	0.818	0.781	0.855	0.473	0.617	0.371	0.440
ConvDEC	0.940	0.916	0.861	0.847	0.784	0.820	0.514	0.588	0.517	0.536
ConvDEC-DA	**0.985**	**0.961**	0.955	**0.949**	**0.970**	**0.953**	0.570	0.632	0.571	**0.608**
DDC	0.965	0.932	**0.965**	0.916	0.967	0.918	**0.619**	**0.682**	**0.573**	0.546
DDC-DA	**0.969**	**0.941**	**0.970**	**0.927**	**0.977**	**0.939**	**0.609**	**0.661**	**0.691**	**0.629**

images. The USPS data set[4] is collected from handwritten digits from envelopes by the U.S. postal service. It contains 9298 grayscale images with size 16 × 16. Fashion [54] is a data set comprising 28 × 28 gray images of 70000 fashion products from 10 categories. Its test set with 10000 images are used in our experiments. The LetterA-J data set[5] is consisted of more than 500k 28 × 28 grayscale images of English letters from A to J. We randomly select 10000 images from its uncleaned subset as test set.

The summary of all data sets is shown in Table 1. The features of each data set are scaled to [0, 1].

4.2. Evaluation measures

Clustering accuracy (ACC) and normalized mutual information (NMI) are used to estimate the performance of comparing algorithms. Their values are both in [0,1]. A higher value of ACC or NMI indicates a better clustering performance.

4.3. Comparing methods

We compare the proposed DDC with both shallow clustering methods and deep ones. Shallow baselines are *k*-means [4], DBSCAN [7], and DenPeak [48]. Deep methods based on both full connected and convolutional autoencoders are compared, including DEC (deep embedded clustering) [21], IDEC (improved

[4] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html

[5] https://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html

Table 3
The average number of detected clusters.

Data set	DDC	DDC-DA
MNIST	10.8 ± 0.4	10.7 ± 0.5
MNIST-test	10.0 ± 0.0	10.0 ± 0.0
USPS	10.0 ± 0.0	10.0 ± 0.0
Fashion	10.5 ± 0.8	10.2 ± 0.8
LetterA-J	9.1 ± 0.8	10.1 ± 0.9

DEC with local structure preservation) [40], DCN (deep clustering network) [22], JULE (joint unsupervised learning for image clustering) [23], DEPICT (deep embedded regularized clustering) [24], ClusterGAN [25], DWSC (deep weighted k-subspace clustering) [26], DKM (deep k-means) [27], VaDE (variational deep embedding) [28], DCC (deep continuous clustering) [30], DED (deep embedding determination) [31], DEC-DA (DEC with data augmentation) [18].

Among all the comparing methods, DBSCAN, DenPeak, DCC, DED, and the proposed DCC do not need the number of clusters in advance. For all other methods, the number of clusters is set to the ground-truth number of categories. When applying DBSCAN, the 4-th nearest neighbor distances are computed w.r.t. the entire data, and parameter *Eps* is set to the median of those values. The *MinPts* value of DBSCAN is always set to 4. For DenPeak, the Gaussian kernel is used and d_c is set such that the average number of points in d_c-neighborhood is approximately 1% × n. To give DenPeak and DED an advantage, the detected number of clusters is set to the true number of classes according to the decision graph. So far, given the ground-truth number of clusters, ConvDEC-DA achieves state-of-the-art clustering performance in image clustering [18]. We compare ConvDEC-DA and its version without using DA in our experiments.

The reported ACC and NMI values are either excerpted from the original papers, or are the average values of running the released code with corresponding suggested parameters for 10 independent trials.

5. Results and analysis

5.1. Results on real image data

Table 2 gives the clustering results of comparing methods measured by ACC and NMI. In each column, the best two results are highlighted in boldface. From Table 2 we have the following observations: (1) The shallow models generally perform worse than deep clustering methods. DBSCAN works the worst mainly because it is hard to choose suitable parameters in high dimensional space. (2) Data augmentation (DA) can improve the clustering performance. Except for two methods using DA (i.e., ConvDEC-DA and DDC-DA), our DDC always achieves the highest ACC and NMI values. (3) Our DDC-DA always achieves one of the best two clustering results, even the number of clusters is not given. Even given the true number of clusters, DED still performs much worse than DDC and DDC-DA. (4) We also find that ConvDEC-DA sometimes performs unstably. For instance, it can usually obtain a high ACC value ($>$0.98) on MNIST-test, but it performs worse (ACC $<$0.84) occasionally on this data set. This might be caused by the bad initial cluster centers provided by k-means in the learned feature space. By contrast, our DDC and DDC-DA are more stable with small standard deviations.

The average number of clusters detected by our DDC and DDC-DA as well as the corresponding standard deviations are given in Table 3. From Table 3 we find that our methods can always find the correct numbers of categories on MNIST-test, and USPS. On MNIST, Fashion and LetterA-J, the recognized numbers of clusters are slightly different from the true values. These indicate the capability of the proposed DDC framework of automatically recognizing reasonable numbers of clusters.

5.2. Sensitivity analysis

This section tests the sensitivity of DCC w.r.t. the parameter *ratio* on MNIST-test and USPS data sets. The tested range is $[0.05, 0.16]$. Both ACC and NMI values of DDC-DA are reported in Fig. 3, from which we can observe that our method achieves stably excellent performance in a wide range of *ratio*. When applying the DDC methods in real clustering applications, the default value of *ratio* is recommended to be set to 0.1.

5.3. Runtime analysis

We compare our method with DEC-DA [18] because these two models use the same CAE structure and DEC-DA has been proved to be efficient compared with other existing deep clustering methods. The experiments are tested on a server with 32 GB RAM and 2 T P100 GPUs. Concretely, the runtimes of our DDC-DA on MNIST-test and USPS are 737 and 583 s, respectively. Those of ConvDEC-DA are 798 and 436 s, respectively. DDC-DA needs time to estimate the density ρ and δ for each point. ConvDEC-DA needs to refine the CAE with initial cluster centers. Thus, these two methods show competitive performance in terms of efficiency.

6. Discussion

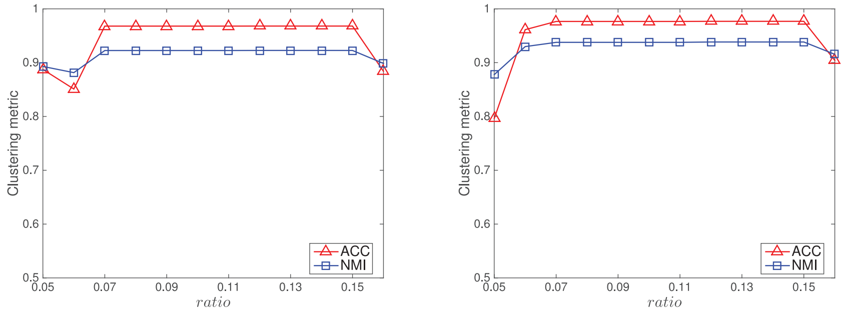
We also conduct experiments to directly use t-SNE to reduce the original data to the 2-dimensional space and then apply the proposed density-based clustering technique. The clustering results are much worse than our DDC methods. The main reason is that CAE can transform the original data to a lower dimensional space in which the intrinsic local structures are preserved. It is better to further reduce the lower dimensional representations to a 2-dimensional space rather than extracting from the original high dimensional data. As a consequence, DED [31] and our DDC make use of both CAE and t-SNE to obtain the 2-dimensional representations that favor the density-based clustering.

Now, let us come back to the question raised in Section 1: Is it really needed to refine the deep autoencoder with the initial cluster assignment? To answer this question, we first visualize the clustering results on MNIST-test and LetterA-J in the embedded 2-dimensional space of DDC-DA in Figs. 4 and 5, respectively. For data whose clusters are well separated (as shown in Fig. 4(a)), those centroid-based clustering methods, such as ConvDEC-DA, which depends greatly on the initial selection of cluster centers, needs to refine the CAE iteratively to achieve satisfied results. By contrast, our DDC can output remarkable performance without refinement even when several clusters in the middle area have overlapped areas.

For data in which many points from different categories mess together (as shown in the middle area of Fig. 5(a)), the refinement of ConvDEC-DA cannot separate the messed points correctly, neither does our DDC. If this happens and no additional information is given, the effectiveness of refining autoencoder is not significant for both centroid-based and density-based clustering. In our opinion, one needs prior information (e.g., pairwise constraints) or knowledge transferred from related tasks to handle this situation.

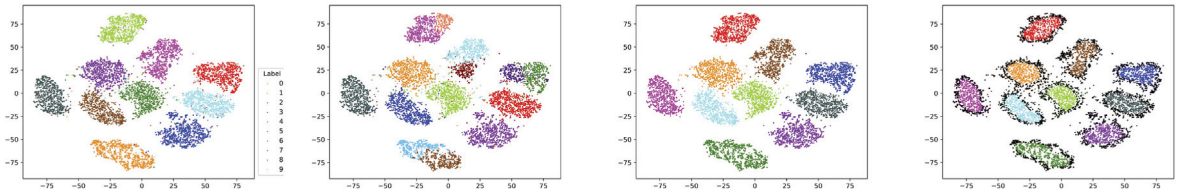
7. Conclusion and future work

We propose a novel deep density-based clustering (DDC) method for image clustering. It is well known that for high-dimensional data such as images, it is difficult to obtain satisfied performance by applying clustering methods in the original space of image data. So in DDC, first, we use CAE with good representation ability to extract 10-dimensional features from the original



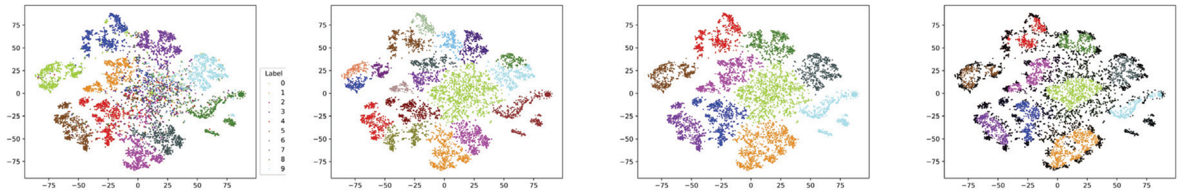
(a) MNIST-test (b) USPS

Fig. 3. Sensitivity analysis of parameter *ratio* (ACC and NMI).



(a) Ground truth labels (b) Initial result (c) Final result (d) Border points

Fig. 4. Visualization of DDC-DA on MNIST-test. (a) The ground truth labels of the embedded 2-dimensional data. (b) The initial result of DDC-DA. (c) The final result of DDC-DA. (d) The border points detected by DDC-DA.



(a) Ground truth labels (b) Initial result (c) Final result (d) Border points

Fig. 5. Visualization of DDC-DA on LetterA-J. (a) The ground truth labels of the embedded 2-dimensional data. (b) The initial result of DDC-DA. (c) The final result of DDC-DA. (d) The border points detected by DDC-DA.

data. After this, t-SNE is used to reduce the 10-dimensional data to a 2-dimensional space, which favors our density-based clustering. DDC consider both the local information of clusters and the importance of points in the clustering process. It is empirically proved to be the new state-of-the-art deep clustering method when the number of clusters is not given. Its efficiency and robustness are also verified. An interesting future work is to exploit semi-supervised learning and transfer learning into deep density-based clustering.

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.

CRediT authorship contribution statement

Yazhou Ren: Conceptualization, Methodology, Formal analysis, Resources, Writing - original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Ni Wang:** Methodology, Software, Validation, Investigation, Data curation, Writing - original draft, Visualization. **Mingxia Li:** Methodology, Software, Validation, Investigation, Writing - original draft, Visualization. **Zenglin Xu:** Resources, Writing - review & editing, Funding acquisition.

Acknowledgments

This paper was in part supported by the National Natural Science Foundation of China (Nos. 61806043, 61572111, and 61832001), the Project funded by China Postdoctoral Science Foundation (No. 2016M602674), and the Guangdong Basic and Applied Basic Research Foundation (No. 2020A1515011002).

References

[1] Y. Chen, J.Z. Wang, R. Krovetz, CLUE: cluster-based retrieval of images by unsupervised learning, IEEE Trans. Image Process. 14 (8) (2005) 1187–1201.

[2] P. Xie, E.P. Xing, Integrating image clustering and codebook learning, in: AAAI, 2015, pp. 1903–1909.

[3] J. Li, J.Z. Wang, Real-time computerized annotation of pictures, TPAMI 30 (6) (2008) 985–1002.

[4] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1967, pp. 281–297.

[5] Y. Ren, U. Kamath, C. Domeniconi, Z. Xu, Parallel boosted clustering, Neurocomputing 351 (2019) 87–100.

[6] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: A review, ACM Comput. Surv. 31 (3) (1999) 264–323.

[7] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: KDD, 1996, pp. 226–231.

[8] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, TPAMI 24 (5) (2002) 603–619.

[9] Y. Ren, U. Kamath, C. Domeniconi, G. Zhang, Boosted mean shift clustering, in: ECML-PKDD, 2014, pp. 646–661.

[10] Y. Ren, C. Domeniconi, G. Zhang, G. Yu, A weighted adaptive mean shift clustering algorithm, in: SDM, 2014, pp. 794–802.

[11] Y. Ren, X. Hu, K. Shi, G. Yu, D. Yao, Z. Xu, Semi-supervised denpeak clustering with pairwise constraints, in: Proceedings of the 15th Pacific Rim International Conference on Artificial Intelligence, 2018, pp. 837–850.

[12] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006, pp. 430–439.

[13] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: NIPS, MIT Press, 2001, pp. 556–562.

[14] S. Huang, Z. Xu, J. Lv, Adaptive local structure learning for document co-clustering, Knowl.-Based Syst. 148 (2018) 74–84.

[15] S. Huang, P. Zhao, Y. Ren, T. Li, Z. Xu, Self-paced and soft-weighted nonnegative matrix factorization for data representation, Knowl.-Based Syst. 164 (2019) 29–37.

[16] F.D.l. Torre, T. Kanade, Discriminative cluster analysis, in: ICML, 2006, pp. 241–248.

[17] J. Chang, L. Wang, G. Meng, S. Xiang, C. Pan, Deep adaptive image clustering, in: CVPR, 2017, pp. 5879–5887.

[18] X. Guo, E. Zhu, X. Liu, J. Yin, Deep embedded clustering with data augmentation, in: ACML, 2018, pp. 550–565.

[19] X. Peng, S. Xiao, J. Feng, W.Y. Yau, Z. Yi, Deep subspace clustering with sparsity prior, in: IJCAI, 2016, pp. 1925–1931.

[20] F. Tian, B. Gao, Q. Cui, E. Chen, T.-Y. Liu, Learning deep representations for graph clustering, in: AAAI, 2014, pp. 1293–1299.

[21] J. Xie, R.B. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: ICML, 2016, pp. 478–487.

[22] B. Yang, X. Fu, N.D. Sidiropoulos, M. Hong, Towards K-means-friendly spaces: Simultaneous deep learning and clustering, in: ICML, 2017, pp. 3861–3870.

[23] J. Yang, D. Parikh, D. Batra, Joint unsupervised learning of deep representations and image clusters, in: CVPR, 2016, pp. 5147–5156.

[24] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, H. Huang, Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, in: ICCV, 2017, pp. 5736–5745.

[25] S. Mukherjee, H. Asnani, E. Lin, S. Kannan, ClusterGAN: Latent space clustering in generative adversarial networks, AAAI (2019) 4610–4617.

[26] W. Huang, M. Yin, J. Li, S. Xie, Deep clustering via weighted *k*-subspace network, IEEE Signal Process. Lett. 26 (11) (2019) 1628–1632.

[27] M.M. Fard, T. Thonet, E. Gaussier, Deep *k*-means: Jointly clustering with *k*-means and learning representations, 2018, pp. 1–14, arXiv preprint arXiv:1806.10069.

[28] Z. Jiang, Y. Zheng, H. Tan, B. Tang, H. Zhou, Variational deep embedding: An unsupervised and generative approach to clustering, in: IJCAI, 2017, pp. 1965–1972.

[29] W.-A. Lin, J.-C. Chen, C.D. Castillo, R. Chellappa, Deep density clustering of unconstrained faces, in: CVPR, 2018, pp. 8128–8137.

[30] S.A. Shah, V. Koltun, Deep continuous clustering, 2018, pp. 1–11, arXiv preprint arXiv:1803.01449.

[31] Y. Wang, E. Zhu, Q. Liu, Y. Chen, J. Yin, Exploration of human activities using sensing data via deep embedded determination, in: Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications, 2018, pp. 473–484.

[32] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, JMLR 9 (2008) 2579–2605.

[33] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.

[34] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, TPAMI 35 (8) (2013) 1798–1828.

[35] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. (2006) 1527–1554.

[36] G. Chen, Deep learning with nonparametric clustering, 2015, pp. 1–14, arXiv preprint arXiv:1501.03084.

[37] M. Shao, S. Li, Z. Ding, Y. Fu, Deep linear coding for fast graph clustering, in: IJCAI, 2015, pp. 3798–3804.

[38] C. Song, F. Liu, Y. Huang, L. Wang, T. Tan, Auto-encoder based data clustering, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Springer, 2013, pp. 117–124.

[39] Y. Ren, K. Hu, X. Dai, L. Pan, S.C. Hoi, Z. Xu, Semi-supervised deep embedded clustering, Neurocomputing 325 (2019) 121–130.

[40] X. Guo, L. Gao, X. Liu, J. Yin, Improved deep embedded clustering with local structure preservation, in: IJCAI, 2017, pp. 1573–1759.

[41] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, OPTICS: ordering points to identify the clustering structure, in: SIGMOD, ACM, 1999, pp. 49–60.

[42] A. Hinneburg, D.A. Keim, et al., An efficient approach to clustering in large multimedia databases with noise, in: KDD, vol. 98, 1998, pp. 58–65.

[43] F. Angiulli, C. Pizzuti, M. Ruffolo, DESCRY: a density based clustering algorithm for very large data sets, in: Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2004, pp. 203–210.

[44] Q. Du, Z. Dong, C. Huang, F. Ren, Density-based clustering with geographical background constraints using a semantic expression model, ISPRS Int. J. Geo-Inf. 5 (5) (2016) 72.

[45] Y. Gu, X. Ye, F. Zhang, Z. Du, R. Liu, L. Yu, A parallel varied density-based clustering algorithm with optimized data partition, J. Spatial Sci. (2017) 1–22.

[46] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, An efficient and scalable density-based clustering algorithm for datasets with complex structures, Neurocomputing 171 (2016) 9–22.

[47] S.T. Mai, X. He, J. Feng, C. Plant, C. Böhm, Anytime density-based clustering of complex data, Knowl. Inf. Syst. 45 (2) (2015) 319–355.

[48] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.

[49] Y. Liu, Z. Ma, F. Yu, Adaptive density peak clustering based on k-nearest neighbors with aggregating strategy, Knowl.-Based Syst. 133 (2017) 208–220.

[50] R. Mehmood, S. El-Ashram, R. Bie, H. Dawood, A. Kos, Clustering by fast search and merge of local density peaks for gene expression microarray data, Sci. Rep. 7 (2017) 45602.

[51] J. Xu, G. Wang, W. Deng, DenPEHC: Density peak based efficient hierarchical clustering, Inform. Sci. 373 (2016) 200–218.

[52] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: ICML, 2008, pp. 1096–1103.

[53] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, JMLR 11 (2010) 3371–3408.

[54] H. Xiao, K. Rasul, R. Vollgraf, Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017, pp. 1–6, arXiv preprint arXiv:1708.07747.