1-2019

Semi-supervised deep embedded clustering

Yazhou REN
University of Electronic Science and Technology of China

Kangrong HU
University of Electronic Science and Technology of China

Xinyi DAI
University of Electronic Science and Technology of China

Lili PAN
University of Electronic Science and Technology of China

Steven C. H. HOI
Singapore Management University, CHHOI@smu.edu.sg

See next page for additional authors

DOI: https://doi.org/10.1016/j.neucom.2018.10.016

Citation

REN, Yazhou; HU, Kangrong; DAI, Xinyi; PAN, Lili; HOI, Steven C. H.; and XU, Zenglin. Semi-supervised deep embedded clustering. (2019). *Neurocomputing*. 325, 121-130. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/4188

Published in Neurocomputing
Volume 325, 24 January 2019, Pages 121-130.
https://doi.org/10.1016/j.neucom.2018.10.016

Semi-supervised deep embedded clustering

Yazhou Ren[a,*], Kangrong Hu[a], Xinyi Dai[a], Lili Pan[b], Steven C.H. Hoi[c], Zenglin Xu[a]

[a] *SMILE Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731 China*
[b] *School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731 China*
[c] *School of Information Systems, Singapore Management University, 81 Victoria Street, 188065 Singapore*

Keywords:
Semi-supervised learning
Deep embedded clustering
Pairwise constraints

ABSTRACT

Clustering is an important topic in machine learning and data mining. Recently, deep clustering, which learns feature representations for clustering tasks using deep neural networks, has attracted increasing attention for various clustering applications. Deep embedded clustering (DEC) is one of the state-of-the-art deep clustering methods. However, DEC does not make use of prior knowledge to guide the learning process. In this paper, we propose a new scheme of semi-supervised deep embedded clustering (SDEC) to overcome this limitation. Concretely, SDEC learns feature representations that favor the clustering tasks and performs clustering assignments simultaneously. In contrast to DEC, SDEC incorporates pairwise constraints in the feature learning process such that data samples belonging to the same cluster are close to each other and data samples belonging to different clusters are far away from each other in the learned feature space. Extensive experiments on real benchmark data sets validate the effectiveness and robustness of the proposed method.

1. Introduction

Clustering is one of very extensively studied topics in artificial intelligence and enjoys a wide range of applications, ranging from document analysis [1,2], regional science [3], image retrieval [4–6], annotation [7], segmentation [8], to network analysis [9–11]. In the past few decades, many clustering algorithms have been proposed, including k-means [12], hierarchical clustering [13], DBSCAN [14], Gaussian mixture model [15], non-negative matrix factorization based clustering methods [16–19], mean shift clustering [20–22], consensus clustering [23–26], graph-based clustering [27,28], and so on. Despite being studied extensively, the performance of traditional clustering methods generally deteriorates with high dimensional data due to unreliable similarity metrics, a phenomenon known as the curse of dimensionality.

To mitigate the curse of dimensionality, a common way is to transform data from a high dimensional feature space to a lower one by applying dimension reduction techniques like principle component analysis (PCA) or feature selection methods [29,30]. Then, clustering is performed in the lower dimensional feature space. However, this scheme ignores the interconnection between features learning and clustering. To address this issue, the work [31] proposes to perform clustering and feature learning simultaneously by integrating k-means and linear discriminant analysis (LDA) into a joint framework. Nevertheless, the representation ability of features learned by these shallow models is limited [32].

In recent years, deep neural networks (DNN) that own better representation ability have been broadly applied in many machine learning tasks [33–37]. Lately, some work has been done to successfully apply deep neural networks in clustering tasks [38–42]. The resulting model is called *deep clustering*. Peng et al. [38] and Tian et al. [39] divide the deep clustering into two phases, i.e., feature transformation using DNN and clustering. In contrast, feature mapping and clustering are jointly learned in [40]. Xie et al. [40] propose deep embedded clustering (DEC) to learn a mapping from the high original feature space to a lower-dimensional one in which an effective objective is optimized. Yang et al. [41] and Chang et al. [42] make use of deep convolutional neural network (CNN) for image data clustering.

Traditional clustering methods refer to unsupervised settings. But, in many real machine learning and computer vision tasks, we know some prior knowledge such as pairwise constraints a-prior. There are typically two kinds of pairwise constraints: must-link constraints and cannot-link constraints. Must-link constraints specify that two instances are known to be in the same cluster in advance, while cannot-link constraints indicates the corresponding two instances belong to different clusters. Semi-supervised learning can use these constraints to improve the learning ability and has produced a huge impact over various machine learning applications [43]. Lately, a number of semi-supervised clustering (SSC)

[*] Corresponding author.
E-mail address: yazhou.ren@uestc.edu.cn (Y. Ren).

methods which take advantage of pairwise constraints have been developed [44–47]. Obviously, despite its success in clustering, DEC is not able to make use of such prior information to guide the clustering process and to further enhance the clustering performance.

To address this issue, we propose semi-supervised deep embedded clustering (SDEC) that incorporates semi-supervised information in DEC to further improve its effectiveness. By integrating pairwise constraints, SDEC considerably improves the quality of clustering results over DEC. Specifically, SDEC makes use of pairwise constraints in the feature learning process such that data samples from the same cluster are enforced to be close to each other and data samples from different clusters are enforced to be far away from each other in the learned feature space where the final cluster assignment is conducted.

The contributions of this paper are summarized below:

- We propose a new semi-supervised clustering scheme SDEC which simultaneously learns feature transformation and cluster assignment jointly by integrating with pairwise constraints. A joint objective considering both the unlabeled data and prior information is developed.
- By leveraging the prior knowledge of pairwise constraints, SDEC significantly improves the clustering performance of the state-of-the-art DEC. The proposed method is also robust to the choice of parameters.
- SDEC can address the curse of dimensionality and is effective in clustering high-dimensional data. Experimental results on real image and document data sets demonstrate its effectiveness and robustness.

The rest of this paper is organized as follows: Section 2 reviews related work and Section 3 introduces the proposed method. Sections 4 and 5 present the experimental settings and the empirical results, respectively. Conclusion and future work are provided in Section 6.

2. Related work

2.1. Deep clustering

Deep clustering is a new category of clustering that has arisen in recent years. Inspired by the similarity between eigendecomposition in spectral methods and auto-encoder [48] in learning lower-dimensional representation, Tian et al. [39] were the first to introduce deep neural network in the field of clustering, which simply combines a nonlinear embedding of the original graph and k-means algorithm in the embedding space. Chen [49] learns representation using a deep belief network and then runs nonparametric maximum margin clustering in the feature space. Shao et al. [50] proposed a linear coder which can be stacked and layerwise trained to learn feature representation for graph clustering. Peng et al. [38] incorporate structure prior in the representation learning via auto-encoder so that local and global subspace structure can be obtained. Then k-means is applied in the learned space to get the final clustering result. Law et al. [51] proposed a deep supervised clustering metric learning method to learn data representation, given the ground-truth partition.

These algorithms mentioned above show commonplace in a two-stage procedure. They first learn representations in a low dimensional feature space, and then run clustering algorithm on the embedding space. In contract, Song et al. [52] embedded an objective of clustering into the auto-encoder model such that the data representations in the embedding space are close to their corresponding cluster centers. Xie et al. [40] proposed a framework called deep embedded clustering (DEC) which jointly learns feature representation and cluster assignments. Guo et al. [32] improved the DEC framework and proposed improved deep embedded clustering (IDEC) with local structure preservation. Since DEC is unsupervised, it cannot use prior information to guide the learning process. In this paper, we develop a semi-supervised version of DEC to alleviate this problem.

2.2. Semi-supervised clustering

Semi-supervised learning is a learning category falls between unsupervised learning and supervised learning, which utilizes both labeled and unlabeled data. There are generally three types of semi-supervised learning: semi-supervised classification [53–56], semi-supervised dimension reduction [57–59], and semi-supervised clustering. In semi-supervised clustering, pairwise must-link and cannot-link constraints are often used [44,46,60,61]. The constraints specify the relation of two data points in the data sets. [62,63] are two variants of k-means that incorporate pairwise constraints. Basu and Mooney [45] developed a hierarchical density based clustering algorithm under the semi-supervised setting. C-DBSCAN [64] extends DBSCAN in semi-supervised scenario, handling the situation when the clusters are diffuse, partially overlapping, connected by bridges or having very different densities. Yu et al. utilized the semi-supervised information in ensemble clustering techniques to further improve their performance [65–68].

Typical semi-supervised clustering methods work in the original feature space with worse representation ability. It is reasonable to do semi-supervised clustering with DNN to make SSC more powerful. Chen [69] extended semi-supervised clustering to deep feature learning, which performs semi-supervised maximum margin clustering on the learned features of DNN and iteratively updates parameters according to most violate constraints, proving that semi-supervised information do improve the deep representation for clustering.

3. Semi-supervised deep embedded clustering

This section elucidates the proposed semi-supervised deep embedded clustering (SDEC) with pairwise constraints. Consider a data set X of n unlabeled samples $\{x_i \in \mathbb{R}^d\}_{i=1}^n$ where d is the dimension. The set of initial must-link constraints is denoted by $ML = \{(x_i, x_j) : x_i$ and x_j belong to the same cluster$\}$ and the set of cannot-link constraints is $CL = \{(x_i, x_j) : x_i$ and x_j belong to different clusters$\}$, $1 \le i,\ j \le N$. The number of cluster K is chosen according to prior knowledge, each cluster is represented by a center $\mu_j, j = 1, \ldots, K$. We seek to find a nonlinear transformation f_θ: $X \to Z$ that maps the data from high-dimensional original space X to latent feature space Z. Here, θ represents the model parameters. The learned f_θ is expected to favor the clustering task and semi-supervised information. Our final goal is to obtain an appropriate partition of data in feature space Z by utilizing the unlabeled data and the user-specified pairwise constraints.

In general, our proposed SDEC has two key steps, i.e., parameter initialization via stacked auto-encoder (SAE) [48] and clustering with pairwise constraints.

3.1. Parameter initialization

We choose deep neural networks (DNN) to initialize the nonlinear transformation f_θ due to its better representation ability. Concretely, we initialize the DNN structure with SAE, the same as what DEC [40] does. Each layer of the network is a denoising auto-encoder [70] trained to reconstruct the previous layer's output after random corruption. After training we concatenate all the encoder and decoder layers together to form a deep auto-encoder [48]. Please refer to [40] for more details.

The encoder layers are exactly what we need as the initial mapping f_θ between the original feature space and the latent learned space. The embedded data points $\{z_i \in Z\}_{i=1}^n$ in the learned space Z are valid feature representations for the original input data samples. We then employ k-means clustering on them to obtain K initial centers $\{\mu_j\}_{j=1}^K$ in space Z.

3.2. Clustering with pairwise constraints

3.2.1. Minimization with KL divergence

DEC [40] makes use of the student's t-distribution [71] to measure the similarity between embedded point z_i and center μ_j as :

$$q_{ij} = \frac{(1 + \|z_i - \mu_j\|^2)^{-1}}{\sum_{j'}(1 + \|z_i - \mu_{j'}\|^2)^{-1}}, \tag{1}$$

where $z_i = f_\theta(x_i) \in Z$ corresponds to $x_i \in X$ after embedding, μ_j is the center of the jth cluster in the embedded space, and $\| \cdot \|$ denotes L2-norm. q_{ij} is considered as the probability of assigning data point i to cluster j, and $q_i = [q_{i1}, q_{i2}, \ldots, q_{iK}]^T$ is considered as a soft assignment of data point i.

The DEC model iteratively refines the cluster assignments by learning from their high confidence assignments. In each step, DEC matches the soft assignment Q to an auxiliary target distribution P, which is computed as:

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j'} q_{ij}^2/f_{j'}}, \tag{2}$$

where $f_j = \sum_i q_{ij}$. In DEC [40], KL divergence between the soft assignment Q and the target distribution P ($KL(P\|Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}$) is minimized to refine the nonlinear transformation f_θ, i.e., the deep neural network structure initialized by the encoder layers of SAE.

3.2.2. Minimization with pairwise constraints

The main contribution of DEC is the use of KL divergence. It uses data points with high confidence as supervision and makes points in each cluster distribute more densely. However, DEC cannot make use of user-specified pairwise constraints to guide the clustering procedure. To address this, we consider adding pairwise constraints to the objective of DEC to lead the direction of clustering and embedding.

Firstly, we define a matrix to describe pairwise constraints ML and CL as[1]:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{bmatrix}. \tag{3}$$

For must-link constraints, when x_i and x_k are assigned to the same cluster, $a_{ik} = 1$. If x_i and x_k satisfy cannot-link constraints, $a_{ik} = -1$. Other entities in this matrix are all zero.

The pairwise constraints specify whether a pair of data examples belong to the same class (must-link constraints) or different classes (cannot-link constraints). We expect that points with the same label should be closer to each other, while points from different classes are far away from each other in latent feature space. To this end, we define the objective of SDEC as:

$$L = KL(P\|Q) + \lambda \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^n a_{ik} \|z_i - z_k\|^2$$

$$= \underbrace{\sum_{i=1}^n \sum_{j=1}^K p_{ij} log \frac{p_{ij}}{q_{ij}}}_{L_u} + \lambda \underbrace{\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^n a_{ik} \|z_i - z_k\|^2}_{L_s}, \tag{4}$$

where n is the number of data points and λ is a trade-off parameter which is defined by the user. When $\lambda = 0$, SDEC degenerates into DEC. Actually, minimizing Eq. (4) can minimize the costs of violated constraints, thus being able to simultaneously learn feature representations and perform clustering assignments to favor the user-specified constraints.

As Eq. (4) shows, the overall loss function of the proposed SDEC can be divided into two parts, the unsupervised clustering loss L_u and the semi-supervised constraint loss L_s. L_u is the KL divergence loss between the soft assignments q_i and the auxiliary distribution p_i. L_u can learn the latent representations of original data that favor clustering tasks. The semi-supervised loss L_s denotes the consistency between the learned representation $\{z_i\}_{i=1}^n$ with the prior information A. Intuitively, if two points satisfy $(x_i, x_k) \in ML$, then $a_{ik} = 1$. To minimize Eq. (4), the distance between z_i and z_k will be small in the latent space Z. Similarly, if $(x_i, x_k) \in CL$, the distance $\|z_i - z_k\|^2$ will be large in space Z. As a consequence, SDEC not only learns good representation for clustering, but also makes points from the same class more close and points from different classes separate from each other. In this way, those points in between-cluster areas can be pulled more correctly and the inappropriate cluster assignments can be somehow corrected with the use of prior information. The framework of SDEC is shown in Fig. 1.

3.2.3. Optimization

We use the stochastic gradient descent (SGD) and backpropagation to optimize Eq. (4). It can be checked that the gradient of objective L w.r.t. feature-space embedding of each data point z_i can be computed as:

$$\frac{\partial L}{\partial z_i} = 2 \sum_{j=1}^K (1 + \|z_i - \mu_j\|^2)^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j)$$
$$+ \frac{2\lambda}{n} \sum_{k=1}^n a_{ik}(z_i - z_k). \tag{5}$$

The gradient of L w.r.t. each cluster center μ_j in space Z is calculated by:

$$\frac{\partial L}{\partial \mu_j} = -2 \sum_{i=1}^n (1 + \|z_i - \mu_j\|^2)^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j) \tag{6}$$

The proofs of Eqs. (5) and (6) are given in Theorems 1 and 2, respectively. During backpropagation, the gradients $\partial L/\partial z_i$ are passed down to update the DNN's parameter θ. The gradients $\partial L/\partial \mu_i$ are used to update the clustering centers $\{\mu_j\}_{j=1}^K$ via SGD. We stop the algorithm if less than *tol*% of points change their cluster assignments between two consecutive updates or the maximal number of iterations is reached. Then, the final clustering result is obtained.

The detailed procedure is summarized in Algorithm 1. As Algorithm 1 shows, SDEC updates the soft assignments every T iterations. As in [32,40], T is considered as batch size and is always set to 256 throughout the experiments. When updating the clustering assignments (Lines 10 and 11 of Algorithm 1), the ith point is assigned to cluster j with the highest q_{ij} value, i.e., $y_i \leftarrow \arg\max_j q_{ij}$.

3.3. Algorithm analysis

Theorem 1. *The gradient of objective L w.r.t. z_i is computed by Eq. (5).*

[1] The $n \times n$ matrix A is extremely sparse and is stored as a sparse matrix to save space in implementation.

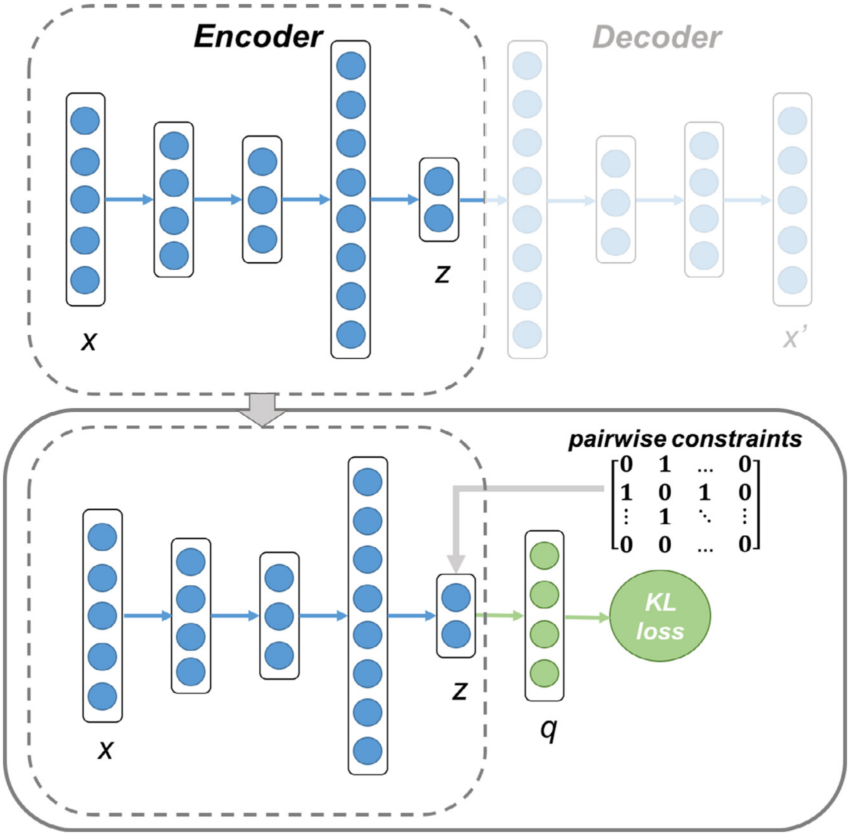


Fig. 1. The framework of SDEC. The upper part of the figure shows the parameter initialization of the algorithm. We use the encoder layers of a pretrained SAE to initialize the DNN structure. The box with solid line represents the learning process of SDEC. Pairwise constraints are added to the embedding layer Z to direct learning of feature representation. q denotes the soft assignment of each data point and is used to compute the KL divergence loss. SDEC takes advantage of both semi-supervised loss and KL divergence loss to update the parameters of DNN.

Algorithm 1 Semi-supervised deep embedded clustering.

```
Require: Data set $X$; coefficient $\lambda$; number of clusters $K$;
         pairwise constraints matrix $A$; update interval $T$;
         stopping threshold $tol\%$.
Ensure: Cluster assignments $\{y_i\}_{i=1}^n$; cluster centers $\{\mu_i\}_{i=1}^K$; deep
        mapping $f_\theta$.
Step 1 → Initialization with SAE
Pretrain SAE and obtain $K$ initial centers $\{\mu_j\}_{j=1}^K$ and cluster
    assignments $\{y_i\}_{i=1}^n$ by running $k$-means in the latent space $Z$.
Step 2 → Clustering with pairwise constraints
for $iter \in \{0, 1, \dots, MAXITER\}$ do
    Choose a batch of samples $S \subset X$.
    if $iter\%T == 0$ then
        $z_i \leftarrow f_\theta(x_i), \forall x_i \in X$.
        Compute all $q_{ij}$ values according to Eq. (1).
        Compute all $p_{ij}$ values according to Eq. (2).
        Save old assignments: $y_{old\,i} \leftarrow y_i$.
        Update label assignments: $y_i \leftarrow \arg\max_j q_{ij}$.
        if $(\sum_{i=1}^n y_{old\,i} \neq y_i)/n < tol\%$ then
            Stop training
        end if
    end if
    Update $\theta$ and $\{\mu_j\}_{j=1}^K$ via Eqs. (5) and (6).
end for
```

Proof. Objective L can be rewritten as:

$$L = \sum_{i=1}^{n}\sum_{j=1}^{K} p_{ij}\log\frac{p_{ij}}{q_{ij}} + \lambda\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n} a_{ik}\|z_i - z_k\|^2,$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{K}(p_{ij}\log p_{ij} - p_{ij}\log q_{ij}) + \lambda\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n} a_{ik}\|z_i - z_k\|^2. \quad (7)$$

Its gradient w.r.t. z_i is:

$$\begin{aligned}\frac{\partial L}{\partial z_i} &= \sum_{i=1}^{n}\sum_{j=1}^{K}(p_{ij}\log p_{ij} - p_{ij}\log q_{ij}) + \lambda\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{n} a_{ik}\|z_i - z_k\|^2, \\ &= -\sum_{j=1}^{K}\frac{\partial(p_{ij}\log q_{ij})}{\partial z_i} + \frac{2\lambda}{n}\sum_{k=1}^{n} a_{ik}(z_i - z_k), \\ &= -\sum_{j=1}^{K} p_{ij}\frac{\partial(\log q_{ij})}{\partial z_i} + \frac{2\lambda}{n}\sum_{k=1}^{n} a_{ik}(z_i - z_k). \end{aligned} \quad (8)$$

When updating z_i, p_{ij} is already computed and is considered as a constant number. Thus, Eq. (8) holds. We then compute:

$$\begin{aligned}\frac{\partial(\log q_{ij})}{\partial z_i} &= \frac{\partial\left(\log\frac{(1+\|z_i-\mu_j\|^2)^{-1}}{\sum_{j'}(1+\|z_i-\mu_{j'}\|^2)^{-1}}\right)}{\partial z_i}, \\ &= \frac{\partial(\log(1+\|z_i-\mu_j\|^2)^{-1})}{\partial z_i} \\ &\quad - \frac{\partial(\log\sum_{j'}(1+\|z_i-\mu_{j'}\|^2)^{-1})}{\partial z_i}, \\ &= -\frac{2(z_i-\mu_j)}{1+\|z_i-\mu_j\|^2} \\ &\quad +2\left(\sum_{j'}(z_i-\mu_{j'})(1+\|z_i-\mu_{j'}\|^2)^{-2}\right)\end{aligned}$$

$$\times \frac{1}{\sum_{j'}(1+\|z_i-\mu_{j'}\|^2)^{-1}}. \tag{9}$$

Let $S=\sum_{j'}(1+\|z_i-\mu_{j'}\|^2)^{-1}$, then $q_{ij}S=(1+\|z_i-\mu_j\|^2)^{-1}$. We can obtain:

$$\begin{aligned}\frac{\partial(\log q_{ij})}{\partial z_i} &= -2(z_i-\mu_j)(1+\|z_i-\mu_j\|^2)^{-2}\frac{1}{q_{ij}S}\\ &\quad +2\left(\sum_{j'}(z_i-\mu_{j'})(1+\|z_i-\mu_{j'}\|^2)^{-2}\right)\frac{1}{S}.\end{aligned} \tag{10}$$

Substituting Eq. (10) into Eq. (8), we have:

$$\begin{aligned}\frac{\partial L}{\partial z_i} &= 2\sum_{j=1}^{K} p_{ij}\Bigg\{(z_i-\mu_j)(1+\|z_i-\mu_j\|^2)^{-2}\frac{1}{q_{ij}S}\\ &\quad -\left[\sum_{j'}(z_i-\mu_{j'})(1+\|z_i-\mu_{j'}\|^2)^{-2}\right]\frac{1}{S}\Bigg\}\\ &\quad +\frac{2\lambda}{n}\sum_{k=1}^{n}a_{ik}(z_i-z_k),\\ &= 2\sum_{j=1}^{K} p_{ij}(z_i-\mu_j)(1+\|z_i-\mu_j\|^2)^{-2}\frac{1}{q_{ij}S}\\ &\quad -2\left[\sum_{j'}(z_i-\mu_{j'})(1+\|z_i-\mu_{j'}\|^2)^{-2}\right]\frac{1}{S}\\ &\quad +\frac{2\lambda}{n}\sum_{k=1}^{n}a_{ik}(z_i-z_k),\\ &= 2\sum_{j=1}^{K} p_{ij}(z_i-\mu_j)(1+\|z_i-\mu_j\|^2)^{-2}\frac{1}{q_{ij}S}\\ &\quad -2\left[\sum_{j'}(z_i-\mu_{j'})(1+\|z_i-\mu_{j'}\|^2)^{-2}\right]\frac{q_{ij'}}{q_{ij'}S}\\ &\quad +\frac{2\lambda}{n}\sum_{k=1}^{n}a_{ik}(z_i-z_k),\\ &= 2\sum_{j=1}^{K} p_{ij}(z_i-\mu_j)(1+\|z_i-\mu_j\|^2)^{-2}\frac{1}{q_{ij}S}\\ &\quad -2\left[\sum_{j}(z_i-\mu_{j})(1+\|z_i-\mu_{j}\|^2)^{-2}\right]\frac{q_{ij}}{q_{ij}S}\\ &\quad +\frac{2\lambda}{n}\sum_{k=1}^{n}a_{ik}(z_i-z_k),\\ &= 2\sum_{j=1}^{K} p_{ij}(z_i-\mu_j)(1+\|z_i-\mu_j\|^2)^{-1}\\ &\quad -2\sum_{j=1}^{K} q_{ij}(z_i-\mu_j)(1+\|z_i-\mu_j\|^2)^{-1}\\ &\quad +\frac{2\lambda}{n}\sum_{k=1}^{n}a_{ik}(z_i-z_k),\\ &= 2\sum_{j=1}^{K}(p_{ij}-q_{ij})(z_i-\mu_j)(1+\|z_i-\mu_j\|^2)^{-1}\\ &\quad +\frac{2\lambda}{n}\sum_{k=1}^{n}a_{ik}(z_i-z_k).\end{aligned} \tag{11}$$

The second '=' of Eq. (11) holds is because that $\sum_{j=1}^{K}p_{ij}=1$. □

Theorem 2. *The gradient of objective L w.r.t. μ_j is computed by Eq. (6).*

Proof. Similarly with Theorem 1, it is not hard to prove that $\frac{\partial L}{\partial \mu_j}$ can be computed by Eq. (6). From another view, this theorem can be proven by the observation of [40]. In [40], it is given that:

$$\frac{\partial L_u}{\partial \mu_j} = -2\sum_{i=1}^{n}(1+\|z_i-\mu_j\|^2)^{-1}\times(p_{ij}-q_{ij})(z_i-\mu_j). \tag{12}$$

Since L_s is independent with μ_j, $\frac{\partial L}{\partial \mu_j}=\frac{\partial L_u+L_s}{\partial \mu_j}=\frac{\partial L_u}{\partial \mu_j}$. □

Complexity analysis. The computational complexity of SDEC algorithm is $O(nD^2+nd_eK+n_cd_e)$, where d_e, n_c, and D are the dimension of embedding space, the number of total pairwise constraints, and the maximum number of neurons in hidden layers of DNN, respectively. In general, $K<d_e<D$ and $O(n)=O(n_c)$ hold. Thus, the complexity of SDEC is $O(nD^2)$, which is linear to the data size.

4. Experimental setup

In this section, we first introduce the data sets used in our experiments. Then we describe the implementation in detail, including experiment setup of our algorithm, comparing methods and evaluation metrics.

4.1. Data sets

We evaluate the proposed method on several popular data sets:

- USPS: The USPS data set[2] contains 9298 grayscale images, obtained from the scanning of handwritten digits from envelopes by the U.S. postal service.
- STL-10: The STL-10 data set[3] contains 13,000 color images with the size of 96-by-96. The images are categorized into 10 classes. As in [40], we also use the concatenation of HOG feature and a 8-by-8 color map as input.
- CIFAR-10: The CIFAR-10 data set[4] is consisted of 60,000 images labeled as 10 classes. Each class contains 6000 samples. We concatenate HOG feature and 8-by-8 color map to represent each picture, as same as STL-10.
- MNIST: The MNIST data set[5] consists of 70,000 handwritten digits of 28 × 28 pixel size. We treat each gray image as a 784 dimensional vector. Each dimension is centered and normalized.
- 20NG: 20NG is a subset of the 20 Newsgroups[6], which is a popular data base for document analysis. 20NG contains 3 subcategories of 20-Newsgroup, i.e, comp.graphics, rec.autos, and sci.crypt.

We preprocess all the data sets in the same way as DEC [40] and IDEC [32]. Concretely, we normalize all data sets such that $\frac{1}{d}\|x_i\|_2^2\approx 1$, for each example x_i with the dimension d. The summary of the data sets and image samples are shown in Table 1 and Fig. 2, respectively.

4.2. Experiment settings

As in DEC [40] and IDEC [32], the structure of encoder layers of SAE is set to d-500-500-2000-10 for all data sets, where d is

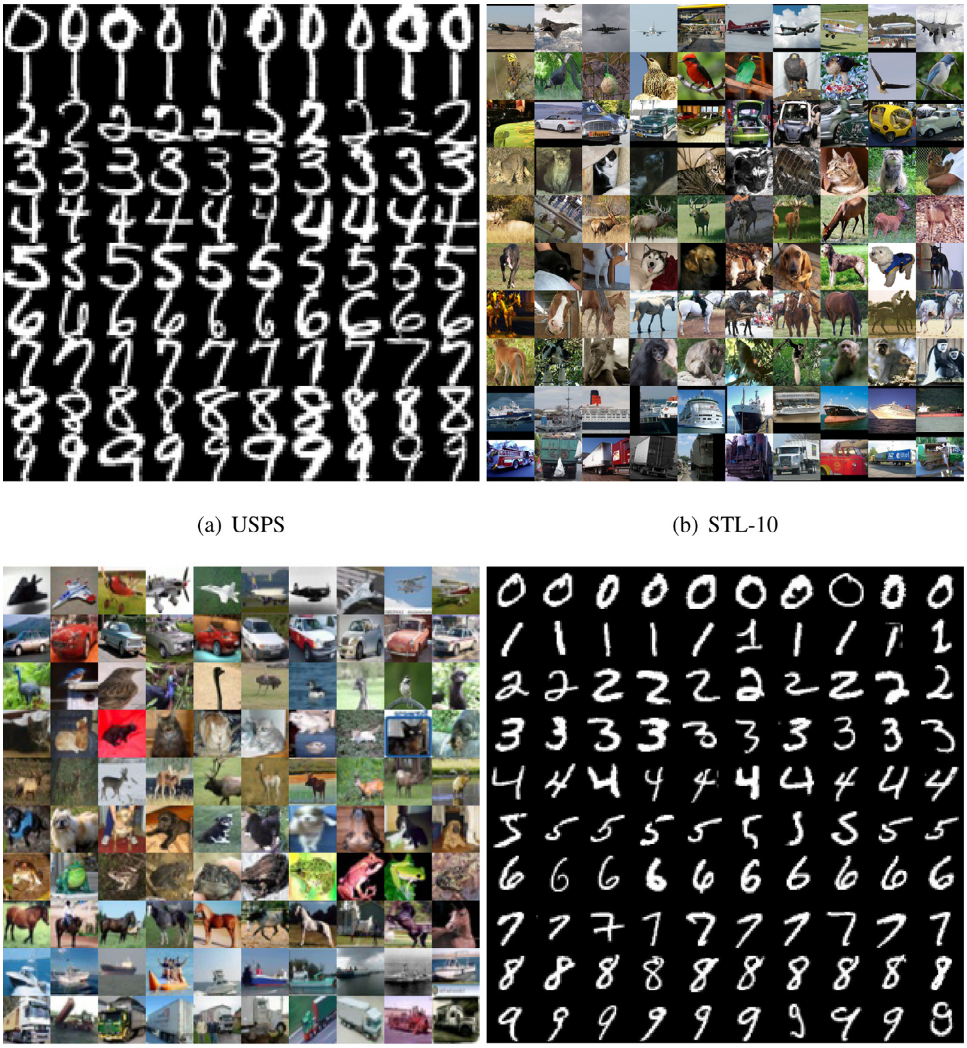
[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html.

[3] https://cs.stanford.edu/~acoates/stl10/.

[4] https://www.cs.toronto.edu/~kriz/cifar.html.

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

[6] http://qwone.com/~jason/20Newsgroups/.



(a) USPS

(b) STL-10

(c) CIFAR-10

(d) MNIST

Fig. 2. Image data sets. For each data set, each row represents one class and randomly shows ten image samples from this class.

Table 1
Data sets used in the experiments.

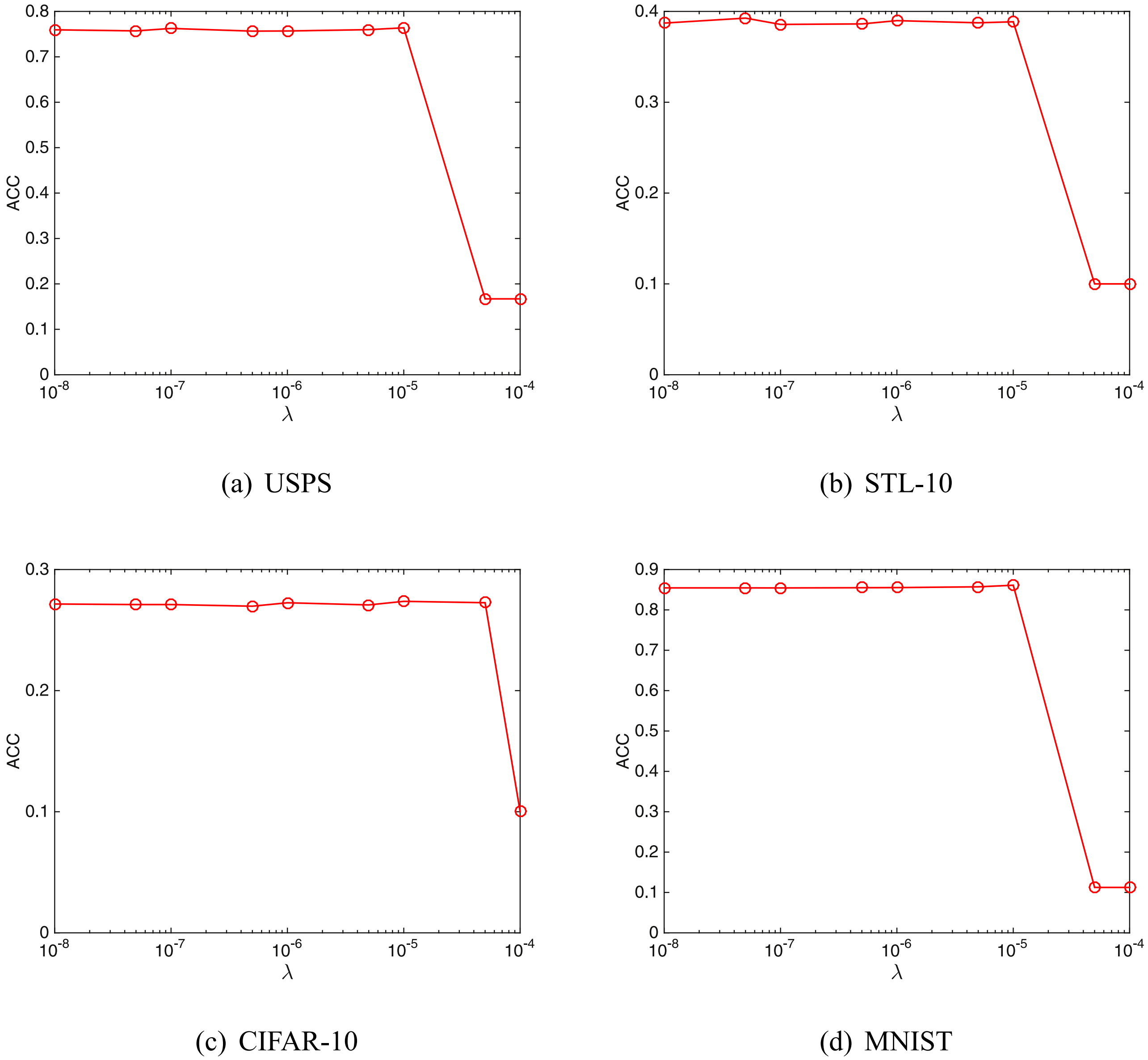
Data set	# Examples	# Classes	# Features
USPS	9298	10	256
STL-10	13,000	10	1428
CIFAR-10	60,000	10	180
MNIST	70,000	10	784
20NG	2965	3	7270

the dimension of input data. All layers are fully connected and all internal layers, except input, output and embedding layer, are activated by ReLU nonlinearity function. We pretrain and fine-tune the auto-encoder using the same parameter setting as DEC, to minimize the influence of parameter tuning.

For each data set, the pairwise constraint matrix A is generated randomly according to ground truth. We randomly select pairs of data points from the data sets: if two data points share the same label, we generate a must-link constraint. Otherwise, a cannot-link constraint is generated. The learning rate of SGD is 0.01. The convergence threshold *tol*% is set to 0.1%. For all algorithms, we set the number of clusters *K* to the number of ground truth categories. We independently run each algorithm 10 times and report the average results. *t*-test is used to assess the statistical significance of the results at 5% significance level.

To evaluate the effectiveness of our proposed algorithm (SDEC), we compare it with several benchmark algorithms. We first compare our algorithm with DEC [40] and IDEC [32]. The classic *k*-means algorithm [12] is applied in both the original and embedding feature spaces. Consider that SDEC is a semi-supervised clustering algorithm, we also perform *k*-means with the supervision of pairwise constraints [62]. The details of the comparing clustering methods are given in the following:

- *k*-means: Run *k*-means [12] algorithm in the original feature space.
- KM-cst (pairwise constrained *k*-means): *k*-means algorithm with pairwise constraints [62] is applied in the original feature space.
- AE+KM: Run *k*-means [12] algorithm in the latent feature space *Z* obtained from SAE. The SAE is pretrained and fine-tuned following the same setting with our method.
- AE+KM-cst: Apply pairwise constrained *k*-means [62] in the latent space *Z* learned by SAE.
- DEC: We use the authors' released code of DEC, with all the parameter settings the same as in [40].
- IDEC: IDEC [32] is an improved version of DEC with local structure preservation. We set the parameters the same as DEC and



(a) USPS

(b) STL-10

(c) CIFAR-10

(d) MNIST

Fig. 3. Sensitivity analysis of parameter λ of SDEC (ACC).

the hyper-parameter is set to the same value that is reported in [32].

4.3. Evaluation metric

To assess the performance of the comparing algorithms, we adopt clustering accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI) for evaluation. The values of NMI and ACC are both in [0,1], while the ARI values range in [−1,1]. For the three metrics, the higher the values are, the better the clustering results are.

5. Results and analysis

5.1. Results on real data

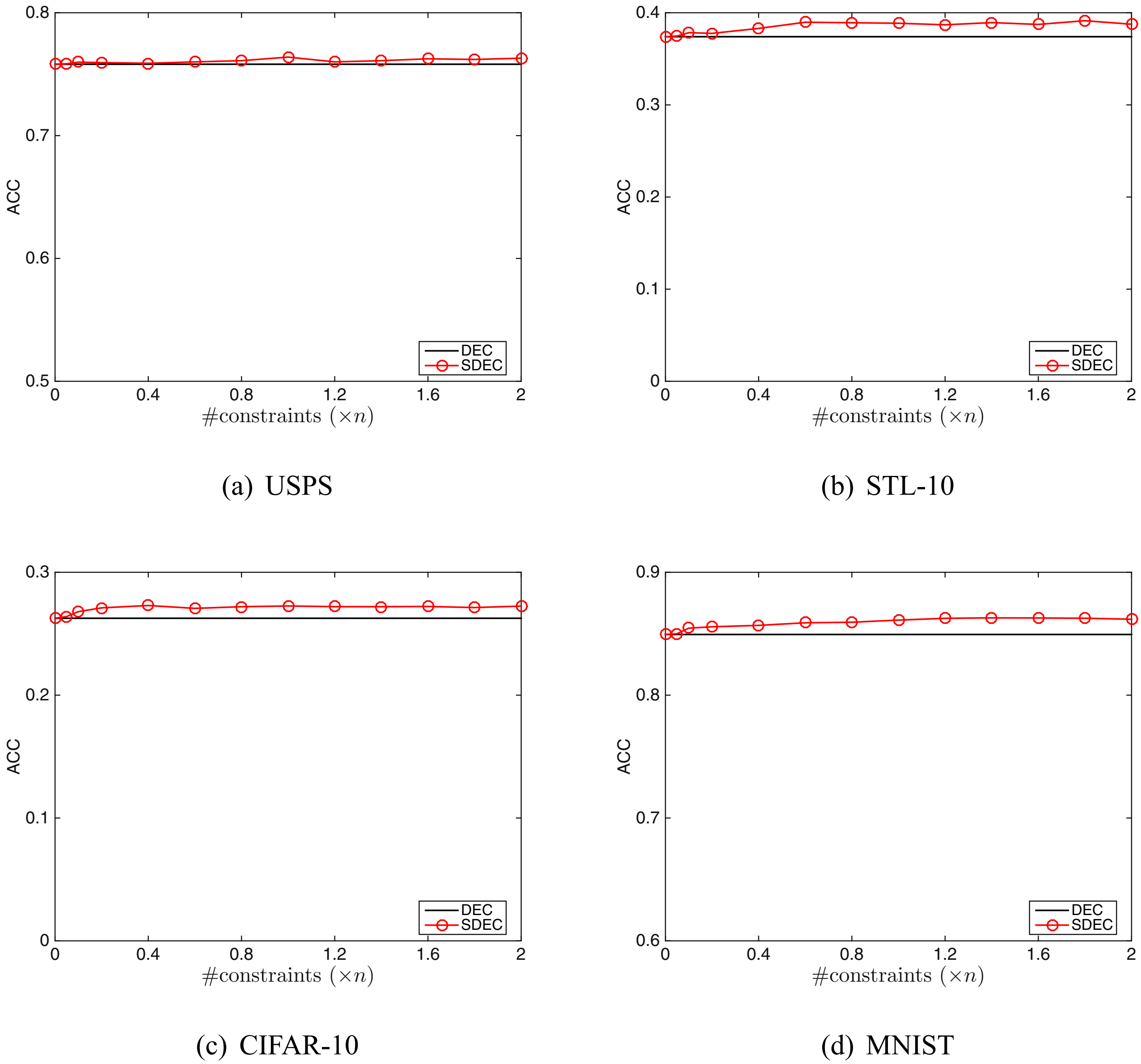
When applying semi-supervised methods on each data set, the number of total pairwise constraints is set to n (the number of data points). The parameter λ of SDEC is set to 10^{-5}. Tables 2–4 show the clustering results measured by ACC, NMI, and ARI, respectively. In each row, the best and comparable results are highlighted in boldface. To save space, the standard deviations (std) are not reported. In fact, the std values of SDEC are pretty small (i.e., SDEC obtains std values of 0.05%, 0.24%, 0.22%, 0.03%, and 0.03% on USPS, STL-10, CIFAR-10, MNIST, and 20NG, respectively).

Table 2
Clustering results measured by ACC (%).

Data	*k*-means	KM-cst	AE+KM	AE+KM-cst	DEC	IDEC	SDEC
USPS	65.67	68.18	70.28	71.87	75.81	75.86	**76.39**
STL-10	28.31	29.09	34.00	35.15	37.40	36.99	**38.86**
CIFAR-10	23.75	23.91	23.89	24.36	26.26	25.02	**27.26**
MNIST	52.98	54.27	74.09	75.98	84.94	83.85	**86.11**
20NG	33.77	33.89	40.81	47.71	50.11	53.63	**78.12**

Table 3
Clustering results measured by NMI (%).

Data	*k*-means	KM-cst	AE+KM	AE+KM-cst	DEC	IDEC	SDEC
USPS	62.00	63.94	66.38	67.29	76.91	**77.68**	**77.68**
STL-10	24.40	24.79	29.37	29.75	32.43	32.53	**32.84**
CIFAR-10	14.67	14.21	15.80	16.03	16.99	**17.27**	**17.20**
MNIST	49.74	50.47	72.26	73.09	81.60	77.89	**82.89**
20NG	0.54	2.27	18.62	25.59	45.36	44.45	**46.36**



(a) USPS

(b) STL-10

(c) CIFAR-10

(d) MNIST

Fig. 4. Sensitivity analysis of the number of pairwise constraints (ACC).

Table 4
Clustering results measured by ARI (%).

Data	*k*-means	KM-cst	AE+KM	AE+KM-cst	DEC	IDEC	SDEC
USPS	53.26	56.03	57.51	58.90	68.79	69.48	**69.86**
STL-10	11.92	12.33	16.30	16.95	19.49	18.85	**20.56**
CIFAR-10	6.98	6.93	6.69	6.87	9.21	7.71	**9.48**
MNIST	37.12	38.17	64.73	66.52	77.30	73.44	**79.20**
20NG	0.01	0.01	1.10	9.24	26.77	26.31	**44.78**

Several interesting observations can be obtained from these three tables: (1) As the tables show, the clustering performance of *k*-means (AE+KM) in the learned space is much better than *k*-means in the original data space, indicating that the great nonlinear representation power of deep neural network do favor the clustering tasks. (2) Three algorithms based on deep embedded clustering framework (i.e., DEC, IDEC, and SDEC), which jointly learns feature representation and cluster assignments, outperform AE+*k*-means (AE+KM), which means iteratively updating the feature learning according to the clustering assignments learns better feature representations for clustering. (3) KM-cst generally performs better than *k*-means both in the original space and in the embedding space. This shows incorporating pairwise information do improve clustering performance. (4) The proposed SDEC achieves the best performance and outperforms unsupervised deep embedded clustering algorithm DEC and IDEC. Specifically, SDEC improves upon DEC by a large margin on 20NG with ACC and ARI. This shows the usage of little prior knowledge like pairwise constraints can significantly enhance the clustering performance.

5.2. Sensitivity analysis

In this section we test the sensitivity of SDEC w.r.t. the parameters λ and the number of pairwise constraints on the four image data sets. We first analyze the sensitivity of the parameter λ of SDEC with setting the number of pairwise constraints to n (the size of data sets). Since the difference between the two parts of SDEC's objective is huge (e.g., in an independent run of applying SDEC on MNIST, the value of KL divergence loss L_u in Eq. (4) is 1.3×10^{-2}, while that of loss L_s is -6.7×10^{3}), λ should be set to a quite small value. The test range of λ is $[10^{-8}, 10^{-4}]$ and Fig. 3 gives the results. As showed in Fig. 3, SDEC performs stably in a wide range of λ. The performance of SDEC decreases sharply when λ is relatively big (e.g., $\lambda = 10^{-4}$). The main reason is that the semi-supervised loss L_s dominates in this case.

Then, we set $\lambda = 10^{-5}$ and test the sensitivity of the number of pairwise constraints n_c, which ranges from 0 to $2 \times n$. Fig. 4 gives the results. It can be seen that with the increasing of n_c, the performance of SDEC generally improves in the beginning. Then, SDEC performs stably in a wide range of n_c. This shows that the initial introduction of pairwise constraints into deep embedded clustering will lead to a significant increase of performance, and then the performance becomes stable which means enough prior information has been captured. This observation is generally consistent with semi-supervised learning literature.

6. Conclusion and future work

In this paper, we propose a semi-supervised deep embedded clustering (SDEC) model. SDEC incorporates pairwise constraints to guide the process of feature learning, ensuring that must-link examples are close and cannot-link examples are distinct in the learned feature space. Both KL divergence loss and semi-supervised loss are jointly optimized in the semi-supervised deep clustering framework to gain the deep representation for clustering. Extensive experiments on real image and document data sets demonstrate the effectiveness and robustness of SDEC. An interesting future direction is to exploit manifold constraints into deep embedded clustering.

Acknowledgments

This paper was in part supported by Grants from the Natural Science Foundation of China (Nos. 61806043, 61572111, and 61832001), two Projects funded by China Postdoctoral Science Foundation (Nos. 2016M602674 and 2017M623007), a 985 Project of UESTC (No. A1098531023601041), three Fundamental Research Funds for the Central Universities of China (Nos. ZYGX2016J078, ZYGX2016Z003, and ZYGX2016J034). This research is supported by the National Research Foundation Singapore under its AI Singapore Programme (No. AISG-RP-2018-001).

References

[1] H.K. Kim, H. Kim, S. Cho, Bag-of-concepts: comprehending document representation through clustering words in distributed representation, Neurocomputing 266 (2017) 336–352.

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

[3] Y. Ren, Big data clustering and its applications in regional science, in: L.A. Schintler, Z. Chen (Eds.), Big Data for Regional Science, Routledge, 2018, pp. 257–264.

[4] 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.

[5] S.G. Jacob Goldberger, H. Greenspan, Unsupervised image-set clustering using an information theoretic framework, IEEE Trans. Image Process. 15 (2) (2006) 449–458.

[6] P. Xie, E.P. Xing, Integrating image clustering and codebook learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015, pp. 1903–1909.

[7] J. Li, J.Z. Wang, Real-time computerized annotation of pictures, IEEE Trans. Pattern Anal. Mach. Intell. 30 (6) (2008) 985–1002.

[8] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.

[9] Z. Xu, F. Yan, Y.A. Qi, Bayesian nonparametric models for multiway data analysis, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2) (2015) 475–487.

[10] S. Zhe, Z. Xu, X. Chu, Y.A. Qi, Y. Park, Scalable nonparametric multiway data analysis, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS, San Diego, CA, USA, 2015, pp. 1125–1134. May 9–12.

[11] Z. Xu, B. Liu, S.Z.H. Bai, Z. Wang, J. Neville, Variational random function model for network modeling, IEEE Trans. Neural Netw. Learn. Syst. (2018), doi:10.1109/TNNLS.2018.2837667.

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

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

[14] 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: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.

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

[16] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Proceedings of the Advances in Neural Information Processing Systems, MIT Press, 2001, pp. 556–562.

[17] S. Huang, Y. Ren, Z. Xu, Robust multi-view data clustering with multi-view capped-norm k-means, Neurocomputing 311 (2018a) 197–208.

[18] S. Huang, Z. Kang, Z. Xu, Self-weighted multi-view clustering with soft capped norm, Knowl. Based Syst. 158 (2018b) 1–8.

[19] S. Huang, H. Wang, T. Li, T. Li, Z. Xu, Robust graph regularized nonnegative matrix factorization for clustering, Data Min. Knowl. Discov. 32 (2) (2018c) 483–503.

[20] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.

[21] Y. Ren, C. Domeniconi, G. Zhang, G. Yu, A weighted adaptive mean shift clustering algorithm, in: Proceedings of the SIAM International Conference on Data Mining, 2014a, pp. 794–802.

[22] Y. Ren, U. Kamath, C. Domeniconi, G. Zhang, Boosted mean shift clustering, in: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2014b, pp. 646–661.

[23] A. Strehl, J. Ghosh, Cluster ensembles - a knowledge reuse framework for combining multiple partitions, JMLR 3 (2002) 583–617.

[24] Y. Ren, C. Domeniconi, G. Zhang, G. Yu, Weighted-object ensemble clustering, in: Proceedings of the IEEE Thirteenth International Conference on Data Mining, IEEE, 2013, pp. 627–636.

[25] Y. Ren, C. Domeniconi, G. Zhang, G. Yu, Weighted-object ensemble clustering: methods and analysis, Knowl. Inf. Syst. 51 (2) (2017) 661–689.

[26] Z. Yu, X. Zhu, H. Wong, J. You, J. Zhang, G. Han, Distribution-based cluster structure selection, IEEE Trans. Cybern. 47 (11) (2017) 3554–3567.

[27] Z. Kang, C. Peng, Q. Cheng, Kernel-driven similarity learning, Neurocomputing 267 (2017) 210–219.

[28] Z. Kang, L. Wen, W. Chen, Z. Xu, Low-rank kernel learning for graph-based clustering, Knowl. Based Syst. (2018), doi:10.1016/j.knosys.2018.09.009.

[29] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Proceedings of the Advances in Neural Information Processing Systems, 2006, pp. 507–514.

[30] Y. Ren, G. Zhang, G. Yu, X. Li, Local and global structure preserving based feature selection, Neurocomputing 89 (2012) 147–157.

[31] F.D.l. Torre, T. Kanade, Discriminative cluster analysis, in: Proceedings of the International Conference on Machine Learning, 2006, pp. 241–248.

[32] X. Guo, L. Gao, X. Liu, J. Yin, Improved deep embedded clustering with local structure preservation, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2017, pp. 1573–1759.

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

[34] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[35] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.

[36] H. Liu, L. He, H. Bai, B. Dai, K. Bai, Z. Xu, Structured inference for recurrent hidden semi-Markov model, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI, Stockholm, Sweden, 2018, pp. 2447–2453. July 13–19.

[37] J. Ye, L. Wang, G. Li, D. Chen, S. Zhe, X. Chu, Z. Xu, Learning compact recurrent neural networks with block-term tensor decomposition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 9378–9387.

[38] X. Peng, S. Xiao, J. Feng, W.Y. Yau, Z. Yi, Deep subspace clustering with sparsity prior, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2016, pp. 1925–1931.

[39] F. Tian, B. Gao, Q. Cui, E. Chen, T.-Y. Liu, Learning deep representations for graph clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2014, pp. 1293–1299.

[40] J. Xie, R.B. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: Proceedings of the International Conference on Machine Learning, 2016, pp. 478–487.

[41] J. Yang, D. Parikh, D. Batra, Joint unsupervised learning of deep representations and image clusters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5147–5156.

[42] J. Chang, L. Wang, G. Meng, S. Xiang, C. Pan, Deep adaptive image clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5879–5887.

[43] X. Zhu, Semi-Supervised Learning Literature Survey, Technical Report, Computer Sciences, University of Wisconsin-Madison, 2005. http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.

[44] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., Constrained k-means clustering with background knowledge, in: Proceedings of the International Conference on Machine Learning, 1, 2001, pp. 577–584.

[45] S. Basu, R.J. Mooney, Semi-Supervised Clustering: Learning with Limited User Feedback, Technical Report, The University of Texas at Austin, 2003.

[46] M. Bilenko, S. Basu, R.J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in: Proceedings of the International Conference on Machine Learning, 2004, pp. 81–88.

[47] S. Basu, M. Bilenko, R.J. Mooney, A probabilistic framework for semi-supervised clustering, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2004, pp. 59–68.

[48] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the International Conference on Machine Learning, 2008, pp. 1096–1103.

[49] G. Chen, Deep learning with nonparametric clustering, Comput. Res. Repos. (2015) 1–14. arXiv: 1501.03084.

[50] M. Shao, S. Li, Z. Ding, Y. Fu, Deep linear coding for fast graph clustering, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2015, pp. 3798–3804.

[51] M.T. Law, R. Urtasun, R.S. Zemel, Deep spectral clustering learning, in: Proceedings of the Thirty-fourth International Conference on Machine Learning, 2017, pp. 1985–1994.

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

[53] O. Chapelle, A. Zien, Semi-supervised classification by low density separation, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2005, pp. 57–64.

[54] Z. Xu, R. Jin, J. Zhu, I. King, M.R. Lyu, Z. Yang, Adaptive regularization for transductive support vector machine, in: Proceedings of the Advances in Neural Information Processing Systems, 2009, pp. 2125–2133.

[55] K. Huang, Z. Xu, I. King, M.R. Lyu, Semi-supervised learning from general unlabeled data, in: Proceedings of the Eighth IEEE International Conference on Data Mining, 2008, pp. 273–282.

[56] Z. Xu, R. Jin, J. Zhu, I. King, M.R. Lyu, Efficient convex relaxation for transductive support vector machine, in: Proceedings of the Advances in Neural Information Processing Systems, 2007, pp. 1641–1648.

[57] Z. Xu, I. King, M.R. Lyu, R. Jin, Discriminative semi-supervised feature selection via manifold regularization, IEEE Trans. Neural Netw. 21 (7) (2010) 1033–1047.

[58] Y. Yu, G. Yu, X. Chen, Y. Ren, Semi-supervised multi-label linear discriminant analysis, in: Proceedings of International Conference on Neural Information Processing, 2017, pp. 688–698.

[59] Y. Ren, G. Zhang, G. Yu, Random subspace based semi-supervised feature selection, in: Proceedings of International Conference on Machine Learning and Cybernetics, 2011, pp. 113–118.

[60] D. Klein, S.D. Kamvar, C.D. Manning, From Instance-level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering, Technical Report, Stanford InfoLab, 2002. http://ilpubs.stanford.edu:8090/528/.

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

[62] S. Basu, A. Banerjee, R.J. Mooney, Active semi-supervision for pairwise constrained clustering., in: Proceedings of the SIAM International Conference on Data Mining, 2004, pp. 333–344.

[63] P. Bradley, K. Bennett, A. Demiriz, Constrained K-Means Clustering, Microsoft Research, Redmond, 2000.

[64] C. Ruiz, M. Spiliopoulou, E. Menasalvas, C-DBSCAN: density-based clustering with constraints, in: Proceedings of the International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing, Springer, 2007, pp. 216–223.

[65] Z. Yu, H. Chen, J. You, H. Wong, J. Liu, L. Li, G. Han, Double selection based semi-supervised clustering ensemble for tumor clustering from gene expression profiles, IEEE/ACM Trans. Comput. Biol. Bioinform. 11 (4) (2014) 727–740.

[66] Z. Yu, P. Luo, J. You, H. Wong, H. Leung, S. Wu, J. Zhang, G. Han, Incremental semi-supervised clustering ensemble for high dimensional data clustering, IEEE Trans. Knowl. Data Eng. 28 (3) (2016) 701–714.

[67] Z. Yu, Z. Kuang, J. Liu, H. Chen, J. Zhang, J. You, H. Wong, G. Han, Adaptive ensembling of semi-supervised clustering solutions, IEEE Trans. Knowl. Data Eng. 29 (8) (2017) 1577–1590.

[68] Z. Yu, P. Luo, J. Liu, J.J. You, H.S. Wong, G. Han, J. Zhang, Semi-supervised ensemble clustering based on selected constraint projection, IEEE Trans. Knowl. Data Eng. (2018) 1–14, doi:10.1109/TKDE.2018.2818729.

[69] G. Chen, Deep transductive semi-supervised maximum margin clustering, CoRR (2015) 1–14. arXiv: 1501.06237.

[70] 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, J. Mach. Learn. Res. 11 (2010) 3371–3408.

[71] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.