Hindawi
Scientific Programming
Volume 2021, Article ID 6613452, 12 pages
https://doi.org/10.1155/2021/6613452

Research Article

Semisupervised Deep Embedded Clustering with Adaptive Labels

Zhikui Chen, Chaojie Li, Jing Gao, Jianing Zhang, and Peng Li

School of Software Technology, Dalian University of Technology, Dalian 116620, China

Correspondence should be addressed to Jing Gao; gaojinghit@gmail.com

Received 29 October 2020; Revised 14 December 2020; Accepted 8 January 2021; Published 16 January 2021

Academic Editor: Boxiang Dong

Deep embedding clustering (DEC) attracts much attention due to its outperforming performance attributed to the end-to-end clustering. However, DEC cannot make use of small amount of a priori knowledge contained in data of increasing volume. To tackle this challenge, a semisupervised deep embedded clustering algorithm with adaptive labels is proposed to cluster those data in a semisupervised end-to-end manner on the basis of a little priori knowledge. Specifically, a deep semisupervised clustering network is designed based on the autoencoder paradigm and deep clustering, which well mine the clustering representation and clustering assignment by preventing the shift of labels in DEC. Then, to train parameters of the deep semisupervised clustering network, a back-propagation-based algorithm with adaptive labels is introduced based on the pretrain and fine-tune strategies. Finally, extensive experiments on representative datasets are conducted to evaluate the performance of the proposed method in terms of clustering accuracy and normalized mutual information. Results show the proposed method outperforms the state-of-the-art methods of DEC.

1. Introduction

Clustering, as one of the most important basic research methods in data mining and machine learning, plays an important role in pattern recognition, image retrieval, computer vision, social network analysis, natural language processing, and knowledge discovery [1]. It divides data samples into different categories in the pattern space by exploring potential distribution structures of data. In the past decades, many classical clustering algorithms have been proposed, such as K-means, DBSCAN, Gaussian mixture model, spectral clustering, nonnegative matrix factorization-based clustering, and graph-based clustering [2–5]. Recently, deep clustering has attracted much attention with the increasing collection of high-dimensional data. It can well alleviate the degradation of traditional clustering in the face of high-dimensional input data by learning low-dimensional representations of data. For example, Lv et al. [6] proposed a deep feature-based clustering by using a stacked autoencoder to extract deep text features. To further improve clustering performance on high-dimensional data, some deep end-to-end clustering methods were proposed, which merged deep neural networks into clustering. For instance, Xie et al. [7] proposed deep embedded clustering (DEC), which learns clustering features of data and divides data in a self-learning manner. Hong et al. [8] proposed mini-GCN, which can combine CNN and GCN to extract more distinctive features and overcome the high computational cost of GCN. Zhao et al. [9] separated view-specific irrelevant information from common features, eliminating the influence of useless information in the view.

Those above methods can well mine data patterns in an unsupervised manner, neglecting some prior knowledge in real data, which is represented by a small number of labelled data or pairwise constraints given by experts. Lately, a number of semisupervised clustering methods were proposed [10–13], utilizing both enough unlabelled data and some prior knowledge to improve clustering performance. For example, Hong et al. [14] proposed a semisupervised deep learning framework that can learn more discriminative information from a small-scale hyperspectral image and transfer it to the classification task of large-scale data. However, most of the current semisupervised clustering cannot use a priori knowledge in a strong-supervision manner because they do not use label information to directly guide the learning of cluster centres. Also, they cannot

cluster samples in a data-driven way of learning clustering centroids and clustering-specific representations.

To address those challenges, a new semisupervised joint learning framework is proposed, which jointly learns the feature embedding space and cluster assignment by integrating a small amount of label information in a joint optimization function.

In addition, the previous semisupervised clustering strategies cannot directly use the strong-supervised knowledge of data labels in the deep embedded clustering due to the label shift problem that clustering results are inconsistent with the actual labels of samples. In other words, those labelled samples of the same class are often scattered to the incorrect classes, and this incorrect supervised information destroys pattern structures of data, causing the degradation of deep embedded clustering.

To solve this challenge, a label adaptive strategy is introduced in this paper based on a voting mechanism. Through the label adaptive strategy, the shifted labels generated in the clustering process are projected as the winner label, ensuring that the labelled samples of the same cluster are always in one cluster in the clustering process. So, the proposed strategy can directly use the label loss to guide the clustering process via adjusting the cluster centres and learning clustering-specific representations. The method in this paper is improved on the basis of DEC and expanded to a semisupervised deep clustering method. The contributions of this paper are summarized as follows:

(i) A new semisupervised joint learning framework is proposed, which integrates a small amount label information to jointly learn the feature embedding space and the cluster assignment with the help of a joint optimization function.

(ii) A label adaptive strategy is introduced to correct the label shift of the clustering process. It can not only improve the utilization of label information, but also effectively avoid the potential degradation that the centroid of traditional deep clustering algorithm is dominated by the code network.

(iii) Extensive experiments on two image datasets and one text dataset are conducted, where the results prove that the proposed method greatly outperforms the state-of-the-art clustering methods.

The rest of this paper is organized as follows: we briefly review the related work in Section 2. Section 3 introduces the details of the proposed method. Section 4 introduces the back-propagation-based algorithm with adaptive labels based on the pretraining and fine-tuning strategies. Section 5 introduces the experimental details of this paper. Finally, the conclusions are presented.

2. Related Work

2.1. Unsupervised Clustering. Clustering has attracted a lot of attention and has been greatly developed for a long time. Many excellent clustering algorithms were proposed [15, 16]. For example, K-means is a classical unsupervised clustering algorithm aiming to minimize the sum of the distance between data points and centroids [2]. Fuzzy expectation maximization combines clustering, cluster number detection, and feature selection into an estimation problem to perform the clustering process [17]. Feature clustering hashing (FCH) is a hashing method based on feature clustering, which can generate lower dimensional data with balanced variance on the premise of maintaining similarity in the Euclidean space [18]. The above methods can be regarded as the clustering algorithm based on features. Distance metric learning with side information learns a distance measure that incorporates the given similarity pairs. Learning a Mahalanobis distance metric designs a new distance measurement function that can learn the Mahalanobis distance metric by forcibly adjusting the distance of a given instance and applying it to new data [19]. Bayesian discriminative fuzzy clustering (BDFC) designs a probabilistic method for unsupervised distance metric learning which can maximize the separability between different clusters in the projection space [20]. The above methods can be regarded as the clustering algorithm based on the distance metric learning. Constrained Laplacian rank (CLR) learns graph with k connected components (where k is the number of clusters) and adjusts the data graph as part of the clustering process [21]. Structure doubly stochastic (SDS) learns structured double random matrices by applying low-rank constraints on Laplace matrices of graphs [22]. Multiview spectral clustering is a novel multiview Markov chain clustering method which can utilize complementary information embedded in different views [23]. The above methods can be regarded as the clustering algorithm based on a graph. With the rise of deep learning, the introduction of deep neural network in clustering has received much attention. Deep clustering network (DCN) finds K-means-friendly clustering space through synchronous deep learning and clustering process [24]. Deep embedded clustering (DEC) uses an automatic encoder to complete the transformation of feature space [7]. Ingeniously, it can perform feature extraction and cluster assignment tasks simultaneously. This algorithm achieves good results and becomes a reference for the performance of new deep clustering algorithm. Improved deep embedded clustering (IDEC) improves clustering performance by preserving the local structure of data [25]. Colearning nonnegative correlated and uncorrelated features (CoUFC) [26] recognizes view-specific features and eliminates the influence of irrelevant information to obtain useful interview feature correlation.

There exists some prior information in many actual data, but the above unsupervised methods do not consider the information. In order to make full use of the label information, this paper proposes a new semisupervised joint learning framework, which integrates label information into deep clustering to jointly learn the data representations and the clustering assignment.

2.2. Semisupervised Clustering. Semisupervised clustering is one of the important research directions in the field of data mining. It can guide the clustering process and improve the quality of clustering by using prior knowledge such as paired constraints or a small amount of labelled data. Recently, the semisupervised clustering method has achieved fruitful

results. For instance, semisupervised kernel mean shift clustering (SKMS) maps data points to a high-dimensional kernel space in which constraints are imposed by linear transformation of the mapped points [27]. Semisupervised linear discriminant clustering (SLDC) combines k-means and linear discriminant analysis (LDA) to consider both the clustering and dimensionality reduction and finds the appropriate feature space by using soft LDA with unlabelled examples [28]. Semisupervised nonnegative matrix factorization (CPSNMF) propagates limited constraint information to the entire data set to obtain more supervisory information and utilizes this supervisory information to maintain the geometry of the data space [29]. Semisupervised graph-based clustering (SSGC) uses a graph of k-nearest neighbours and the local density measure of the similarity between vertexes to integrate the seed into the process of building the cluster, improving the quality of the cluster [30]. The above methods can be regarded as an extension of the traditional clustering algorithms by using label information or pairwise constraints. Relevant component analysis (RCA) is an efficient algorithm for learning Mahalanobis metrics by using a version of the constrained Fisher's linear discriminant [31]. Discriminative component analysis (DCA) learns the linear data transformation of the best Mahalanobis distance measurement with context information [32]. Information theoretic metric learning (ITML) uses a relationship between multivariate Gaussian distribution and Mahalanobis distance set to learn a new Mahalanobis distance function [33]. Bregman distance function learning (BKM) presents a new method for learning nonlinear distance functions with edge information, which is to use a nonparametric method similar to support vector machines to learn Bregman distance functions [34]. The above methods can be considered as exploring a new distance metric function by using constraint information. Still some research work is used to explore an integrated framework for semisupervised clustering. For example, the double affinity propagation-based cluster ensemble (AP^2C) integrates affinity propagation (AP) algorithm and normalized cut (Ncut) algorithm into cluster integration framework [35]. It can capture the relationship between attributes, find a group of representative attributes, and eliminate noise attributes. Semisupervised clustering with sequential constraints (SCSC) proposes an efficient dynamic semisupervised clustering framework [36]. It transforms the dynamic clustering process into a search problem on a feasible clustering space, which is defined as a convex shell generated by partitioning multiple sets. Hybrid semisupervised clustering ensemble (HSCE) proposes a semisupervised clustering ensemble framework that uses pairwise constraints or labelled data to generate different basic partitions by using constraint-based semisupervised clustering algorithm and metric-based semisupervised clustering algorithm, respectively, and then integrates these basic partitions into integration functions to obtain target clustering [37].

Traditional semisupervised clustering algorithms are mostly executed in the original space and have poor performance in the face of high-dimensional data. Therefore, it is necessary to enhance its expressiveness by using deep neural network. MDL-RS designs a general multimodal deep learning framework, which can well embed multiple fusion modules and break the performance bottleneck under single modality [38]. Deep transductive semisupervised maximum margin clustering uses labelled and unlabelled data under a given pair of constraints to learn the nonlinear mapping under the maximum margin framework for clustering analysis [39]. This work proves that the deep representation of the original does contribute to the improvement of clustering results. Semisupervised deep embedded clustering (SDEC) incorporates pairwise constraints in the process of feature learning, forcing data samples in the same cluster to be close to each other, and data samples of different clusters are far apart from each other [40].

However, due to the label shift problem, these semisupervised methods cannot directly use label information to guide the learning of cluster centres. Therefore, this paper designs a label adaptive strategy based on the voting mechanism to correct the transfer of labels in the clustering process, directly using the label loss to guide the clustering process and improve the clustering performance.

3. Semisupervised Deep Clustering with Adaptive Labels

In this section, a semisupervised deep embedded clustering algorithm with adaptive labels (Semi-DEC) is introduced to make full use of prior knowledge of a small number of labels. Semi-DEC is composed of a deep code network and a semisupervised embedding network, as shown in Figure 1. The former uses the encoder-decoder paradigm, transferring high-dimensional data into low-dimensional features. It can well address the curse of dimensionality in data. The latter mines knowledge patterns by dividing data into several groups. It can better consider prior knowledge by solving the shift of labels in clustering. The details of those two networks are introduced as follows.

3.1. The Deep Code Network. The deep code network aims to learn latent features of data in a low-dimensional space on the basis of the encoder-decoder network [41]. That is, it computes the hidden representations of data samples, reconstructs data samples from those hidden representations, and minimizes the loss between raw data and reconstructed data. Specifically, given a dataset of n points $X = \{x_i \in R^{d_1}\}_{i=1}^{n}$, where d_1 is the dimension of data, the deep code network learns hidden representations of data in the following form:

$$\overline{x} = \text{Dropout}(x), \tag{1}$$

$$h = g_e(W_e\overline{x} + b_e), \tag{2}$$

where Dropout is the random mapping function that sets some elements of each input to be 0 based on a given probability. $\overline{x}$ is the result of the random mapping of the input x. W_e and b_e are the weight and bias vectors, respectively, which represent the parameters of the encoder

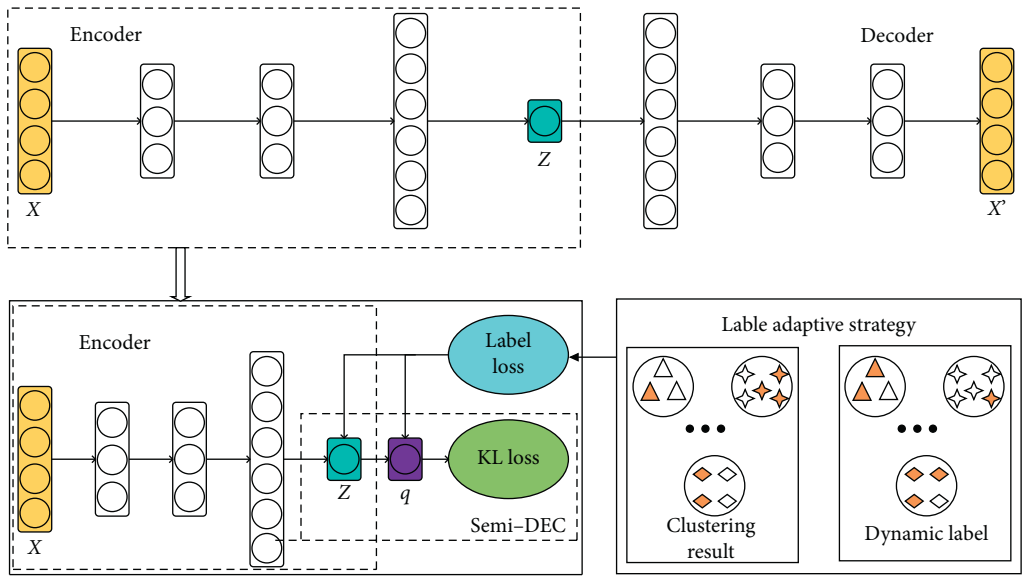

Figure 1: The architecture of the semisupervised deep embedding clustering algorithm with adaptive labels.

network. h is the hidden representation with g_e representing the encoder function.

After obtaining hidden representations of data samples, the deep code network decodes hidden representations by the reconstructing function as follows:

$$\overline{h} = \text{Dropout}(h), \tag{3}$$

$$t = g_d\left(W_d\overline{h} + b_d\right), \tag{4}$$

where $\overline{h}$ is the result of the random mapping of the hidden representation h. W_d and b_d are the weight and bias vectors of the decoder function. t is the reconstructed data, and g_d represents the decoder function.

Finally, the deep code network uses the mean squared error function to measure the loss between raw data and reconstructed data as follows:

$$\text{loss} = \|x - t\|_2^2, \tag{5}$$

where $\frac{2}{2}$ represents the mean squared error function. In Semi-DEC, the loss of the deep code network is used to pretrain parameters.

3.2. The Semisupervised Embedding Network. The semisupervised embedding network aims to divide data into several groups where the distances between samples of the same group are closer than those of different groups. The semisupervised embedding network consists of the unsupervised part that mines intrinsic patterns and the supervised part that uses the small amount of prior knowledge.

3.2.1. The Unsupervised Part. The unsupervised part of the semisupervised embedding network is measured by the KL divergence as follows:

$$L_1 = \text{KL}(P\|Q) = \sum_{i=1}^{n}\sum_{j=1}^{k} p_{ij}\log\frac{p_{ij}}{q_{ij}}, \tag{6}$$

where Q is the cluster assignment of the semisupervised embedding network and P is the target distribution. Given the hidden representations of n data samples $Z = \{z_i \in R^{d_2}\}_{i=1}^{n}$ (d_2 is the dimension of data in the embedding space) and k cluster centroids $\mu_j | j = 1, \ldots, k$, the cluster assignment of the semisupervised embedding network is expressed as

$$q_{ij} = \frac{\left(1 + \left\|z_i - \mu_j\right\|^2\right)^{-1}}{\sum_{j'=1}^{k}\left(1 + \left\|z_i - \mu_{j_\prime}\right\|^2\right)^{-1}}. \tag{7}$$

The target distribution is defined as follows:

$$p_{ij} = \frac{\left(q_{ij}^2 / \sum_{i=1}^{n} q_{ij}\right)}{\left(\sum_{j'=1}^{k} q_{ij}^2 / \sum_{i=1}^{n} q_{ij_\prime}\right)}, \tag{8}$$

where Q is measured by the student distribution and P is the square of Q, which strengthens the membership each sample.

3.2.2. The Supervised Part. The supervised part is introduced to address the shift of labels in the unsupervised part based on the small group of priori knowledge. It is measured by the soft-max loss function as follows:

$$L_2 = -\lambda\sum_{i=1}^{n} a_i y_i' \log q_i = -\lambda\sum_{i=1}^{n}\sum_{j=1}^{k} a_i y_{ij}' \log q_{ij}, \tag{9}$$

where y_i' represents the temporary correction label obtained through the label adaptive strategy, λ is a trade-off parameter to balance the influence of the label loss, q_i represents the label obtained by cluster assignment, and a_i is the sign that indicates whether there is a label of a certain sample and is expressed via

$$a_i = \begin{cases} 1, & y_i \text{ exists}, \\ 0, & \text{else}, \end{cases} \tag{10}$$

where y_i represents the true label of the sample.

Finally, the computation of the semisupervised embedding network is expressed as follows:

$$L = L_1 + L_2 = \sum_{i=1}^{n} \sum_{j=1}^{k} p_{ij} \log \frac{p_{ij}}{q_{ij}} - \lambda \sum_{i=1}^{n} \sum_{j=1}^{k} a_i y_{ij}' \log q_{ij}, \tag{11}$$

which can effectively merge the knowledge of a small number of labels into the unsupervised learning.

3.3. Optimization. We use the stochastic gradient descent (SGD) and back-propagation to optimize the loss function equation (11). It is worth noting that the parameters to be optimized have two parts: feature space embedded of each data point z_i and the cluster centres μ_j. The gradients of L with respect to embedded point z_i can be computed as

$$\frac{\partial L}{\partial z_i} = 2 \sum_{j=1}^{k} \left(1 + \left\|z_i - \mu_j\right\|^2\right)^{-1} \times \left(p_{ij} - q_{ij}\right)\left(z_i - \mu_j\right) - 2\lambda a_i \sum_{j=1}^{k} y_{ij}' \left(1 + \left\|z_i - \mu_j\right\|^2\right)^{-1} \times \left(1 - \frac{q_{ij}}{p_{ij}}\right)\left(z_i - \mu_j\right). \tag{12}$$

The gradients of L with respect to the cluster centre μ_j can be computed as

$$\frac{\partial L}{\partial \mu_j} = -2 \sum_{i=1}^{n} \left(1 + \left\|z_i - \mu_j\right\|^2\right)^{-1} \times \left(p_{ij} - q_{ij}\right)\left(z_i - \mu_j\right) + 2\lambda a_i \sum_{i=1}^{n} y_{ij}' \left(1 + \left\|z_i - \mu_j\right\|^2\right)^{-1} \times \left(1 - \frac{q_{ij}}{p_{ij}}\right)\left(z_i - \mu_j\right). \tag{13}$$

In the process of back propagation, the parameters $\{W_e, b_e\}$ in the deep code network are updated by passing down the gradient $(\partial L/\partial z_i)$. The cluster centre μ_j is updated by gradient $(\partial L/\partial \mu_j)$. The clustering process will be terminated when the cluster assignment between two consecutive iterations is less than tol % or the maximum number of training times is reached.

4. The Back-Propagation Algorithm of Semi-DEC

In this section, the back-propagation algorithm is introduced to train parameters of Semi-DEC. It is composed of two steps, i.e., the unsupervised pretraining step and the semisupervised fine-tuning step. The details of the back-propagation algorithm of Semi-DEC are introduced as follows.

4.1. The Unsupervised Pretraining Step. The unsupervised pretraining step uses the encoder-decoder paradigm to learn generalized features of data and adopts the K-means clustering to explore the centroids hidden in data.

Specifically, given a dataset of n points X and a deep encoder network of m layers, the unsupervised pretraining step models each layer of the deep encoder network as an autoencoder based on equations (1) to (4) to obtain the pretraining parameters of the deep code network. For example, each raw sample x_i in the dataset is input into the autoencoder of the 1st hidden layer, obtaining the hidden representation h_i which is input into the autoencoder of the 2nd hidden layer. After each hidden layer is initialized in the same way, the whole network is trained again in an end-to-end manner by minimizing the reconstruction loss.

Then, the raw data X are mapped into the latent feature space by the deep code network, getting the hidden representations Z. The K-means clustering is conducted on the hidden representations to get initial centroids.

4.2. The Semisupervised Fine-Tuning Step. After obtaining the pretrained deep code network and the initial centroids, Semi-DEC is trained in the semisupervised manner based on the loss function equation (11) to solve the shift of label in unsupervised learning. Specifically, given the raw data X, Semi-DEC constructs the label sign list of samples as defined in equation (10). Then, suppose the number of labelled samples is v, it gathers statistics of the distribution of data which have labels in each epoch as follows:

$$\begin{cases} R = [q_1, q_2, \ldots, q_v], \\ q_i = \arg\max_j q_{ij}, \quad j = 1, 2, \ldots, k, \end{cases} \tag{14}$$

where $q_1, q_2, \ldots, q_v$ represent the assigned labels of those labelled data and their values range from 1 to k. Finally, the temporal labels $q_1, q_2, \ldots, q_v$ are rectified to the label whose number is maximum.

Figure 2 is an example of the label adaptive strategy. For the subset of labelled data with category o, we assume that after cluster assignment, most of the samples are assigned to

category j and a few samples are assigned to other categories such as s and u. Here, o, j, s, and u, respectively, represent different categories. Through the voting mechanism, we believe that the category j with the largest number of samples is the correct result of this subset in cluster assignment. Then, we can rectify the samples that are clustered incorrectly in this round of calculation, that is, make them move closer to category j. The adaptive label algorithm is introduced as follows:

Step 1: Semi-DEC gathers the label distribution of each cluster $\{c_i|i = 1, \ldots, k\}$ based on the output of the semisupervised embedding network $R = [q_1, q_2, \ldots, q_v]$ in each epoch. At the same time, the label with the maximum number is dynamically treated as the correct label.

Step 2: Semi-DEC rectifies those wrong labels according to the statistics of the label distribution.

Step 3: Semi-DEC computes the loss of those samples that are wrongly labelled according to equation (9) to rectify the parameters of network.

Step 4: Semi-DEC fine-tunes the parameters of the deep code network and the semisupervised clustering network to find the final assignment strategy.

With the help of the proposed label adaptive strategy, the labelled data that were wrongly divided in the clustering process are corrected via the voting mechanism, which can effectively solve the label shift problem in a strong-supervision manner by forcing the data that have the same label to be in the same cluster. In other words, this label adaptive strategy preserves the data structure in clustering assignment and cluster-specific feature learning. The overall steps of the back-propagation algorithm of Semi-DEC are shown in Algorithm 1.

5. Experiments

In this section, extensive experiments are conducted on several representative datasets to evaluate the performance of Semi-DEC. The datasets used in our experiment are first introduced. Then, several state-of-the-art clustering algorithms and evaluation metrics are presented. Finally, the implementation and experimental results are illustrated in detail. The detailed information of the datasets is shown in Table 1.

5.1. Datasets

5.1.1. MNIST. The MNIST dataset is composed of 70000 handwritten digits of $28 * 28$ pixel size. In the experiment, each image is reshaped to a 784-dimensional vector.

5.1.2. USPS. The USPS dataset is composed of 9298 handwritten digits of $16 * 16$ pixel size. The images are divided into 10 categories, with a training set size of 7291 and a test set size of 2007.

5.1.3. REUTERS-10K. In the original Reuters data set, there are around 810000 English news stories labelled with a category. Four root categories are as follows: corporate/industrial, government/social, markets, and economics as labels are used, and all documents with multiple labels are further excluded. We computed TF-IDF features on the 2000 most frequent words to represent all documents. A subset of 10000 samples is randomly sampled, referred as REUTERS-10K.

5.2. Compared Methods. To verify the effectiveness of the proposed method, several state-of-the-art algorithms are used as the compared methods. The following is a summary of these algorithms.

5.2.1. K-Means. K-means is a traditional unsupervised clustering algorithm [2]. It guides the division of data sets into K classes based on the principle of minimizing the sum of the distances from the data points to the centroids.

5.2.2. DEC. The deep embedding clustering (DEC) is a deep unsupervised clustering algorithm [7]. It uses an automatic encoder to transform feature of the original data and then performs the clustering process in the feature space.

5.2.3. DCN. The deep clustering network (DCN) is a deep unsupervised clustering algorithm [24]. It combines autoencoder with the K-means and proposes an algorithm that jointly optimizes reconstruction loss and K-means loss.

5.2.4. IDEC. The improved deep embedding clustering (IDEC) is also a deep unsupervised clustering algorithm [25]. It is an improvement to DEC by adding the local structure preservation.

5.2.5. SMKL. The self-weighted multiple kernel learning (SMKL) is a traditional semisupervised clustering algorithm [13]. It constructs the best kernel and assigns an optimal weight for each kernel automatically.

5.2.6. SDEC. The semisupervised deep embedded clustering (SDEC) is a deep semisupervised clustering algorithm [40]. It incorporates pairwise constraints in the process of the feature learning.

5.3. Evaluation Metric. The clustering accuracy (ACC) and normalized mutual information (NMI) are used to evaluate the performance of the proposed method and other compared algorithms, which are widely used in clustering tasks. The values of both ACC and NMI range from 0 to 1. The larger values of both metrics indicate the better clustering results.

ACC is defined as follows:

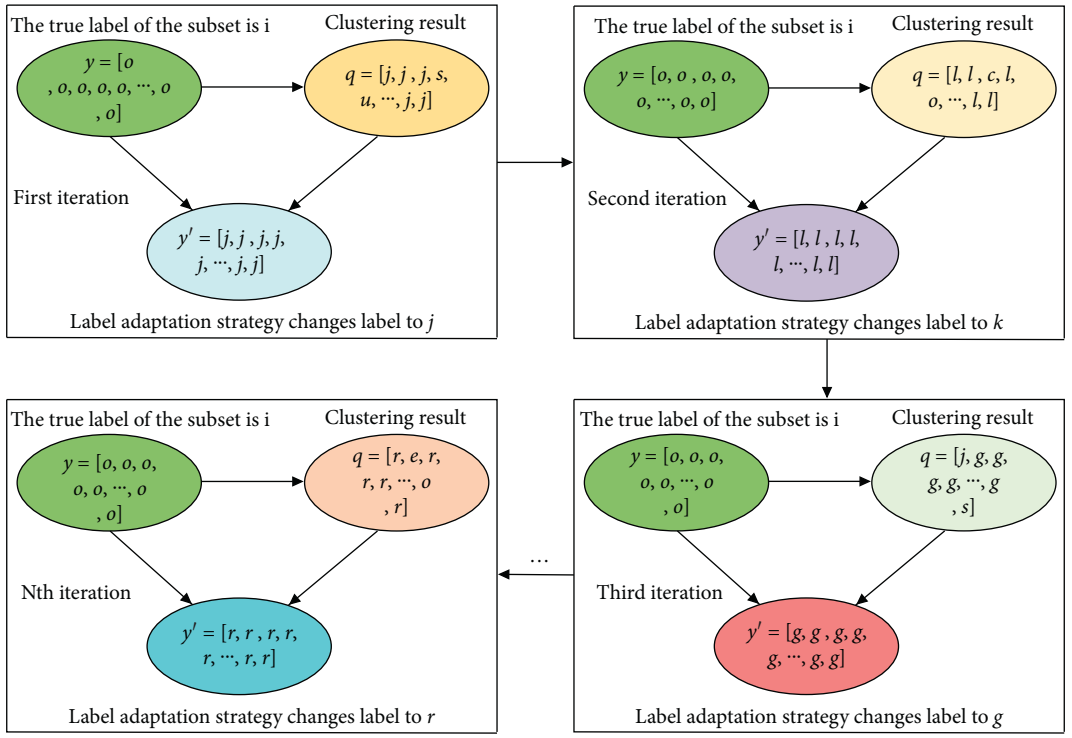

Figure 2: An example of the label adaptive strategy.

Input: the training dataset $\{x_i \in X\}_{i=1}^{n}$, the number of clusters k, the iteration maximum maxiter, and the training threshold.
Output: the cluster assignment Q, the cluster centroids $\{\mu_j\}_{j=1}^{k}$, and the nonlinear mapping f_θ.
Begin
Pretraining computing:
To construct the deep code network.
To initialize network parameters based on the normal distribution.
To train each layer of the deep code network based on the denoising autoencoder strategy.
To connect each pretrained layer and fine-tune network parameters in an end-to-end manner.
To use pretrained deep code network to map raw data into the latent space for obtaining feature z_i.
To use K-means to initialize centroids $\{\mu_j\}_{j=1}^{k}$ based on feature z_i.
Clustering computing with adaptive labels:
To use equations (7) and (8) to compute cluster assignment Q and target assignment P.
To compute $(\sum_{i=1}^{n} q_{\text{old}_i} \neq q_i) < \text{tol}\,\%$.
To use equation (10) for constructing the label list.
To dynamically rectify labels based on the adaptive label algorithm.
To compute the loss based on equation (11).
To update network parameters and centroids.
End

Algorithm 1: Deep semiclustering with adaptive labels.

$$\text{ACC} = \frac{1}{N} \max_{k} \sum_{i=1}^{n} 1\{l_i = k(c_i)\}, \tag{15}$$

where N is the number of samples, l_i is the true label, c_i is the cluster assignment label produced by the algorithm, and k ranges over all possible one-to-one mappings between clusters and labels.

NMI is defined as follows:

$$\text{NMI}(A, B) = \frac{\text{MI}(A, B)}{\sqrt{H(A)H(B)}}, \tag{16}$$

where A is the true cluster set and B is the predicted cluster set. MI (A, B) is the mutual information between A and B. H (A) and H (B) denote the entropies of A and B.

5.4. Parameters Setting. The encoder layer structure of deep code network is set to d-500-500-2000-10 for all data sets, where d is the dimension of the input data. All layers are fully connected, and all internal layers (except the input layer, embedding layer and output layer) are activated by the ReLU nonlinear function. During the pretraining and fine-tuning of the autoencoder network, we use the same parameter settings as in DEC to ensure that the improvement of the experimental results is the contribution of the method proposed in this paper.

For each dataset, the monitor information list A is dynamically generated based on the presence or absence of label information in the dataset. The length of the list is consistent with the size of the data batch taken each time, and its corresponding element value is 1 if the data point has a real label, or 0 if there is no label. The learning rate of SGD is 0.01. The convergence threshold to l % is set to 0.1%. After experimental testing, the trade-off parameter λ of label loss is set to 0.2 (this is determined by a grid search in {0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0}). For all algorithms, we set the cluster number k as the number of ground truth categories. We independently run each algorithm 10 times and report average results.

5.5. Experiment Results. This section demonstrates the results of the compared methods on the three representative datasets. In detail, Tables 2 and 3 report the results in terms of ACC and NMI, respectively. The percentage of labelled data is 30%. In the two tables, the best-performance results are highlighted in bold. It can be seen the proposed method is superior to the state-of-the art methods.

Specifically, compared with the traditional K-means and SMKL methods, the proposed method can learn features of more representational capabilities by the deep code network. Also, the K-means is an unsupervised method, which cannot utilize label information in the clustering process, further leading to the degradation of the performance. Although DEC, DCN, and IDEC also take advantage of deep features of data, they ignore the information hidden in the small amount of label data, resulting that those deep methods produced lower performance than the proposed method. SDEC uses pairwise constraints to guide the process of clustering, which belongs to a weak utilization of supervisory information. Through the label adaptive strategy, we can directly use the label loss, which is a strong use of label information. This is also the key to our proposed approach.

To further illustrate the superiority of the proposed method, we also visualize the clustering results in the training process in Figure 3. We randomly select 1000 samples in each dataset and map the latent representations z into the 2D space. From the change trend of the clustering results, it can be seen that the samples in different clusters become easier to distinguish as the number of trainings increases, and the samples in the same cluster also become closer. This indicates that the learned feature space becomes more suitable for clustering tasks, and it is also a proof that the label adaptive strategy can effectively guide the learning of the feature space and cluster assignment.

Table 1: Datasets statistics.

Datasets	Samples	Dimension	Classes
MNIST	70000	784	10
USPS	9298	256	10
REUTERS-10K	10000	2000	4

Table 2: Clustering results measured by ACC.

Methods	MNIST	USPS	REUTERS-10K
K-means	0.5298	0.6567	0.5162
DEC	0.843	0.7408	0.7369
DCN	0.811	0.73	0.7505
IDEC	0.8806	0.7605	0.7564
SMKL	0.783	0.6819	0.7203
SDEC	0.8611	0.7639	0.6937
Semi-DEC	**0.9648**	**0.8609**	**0.9176**

Also, to evaluate the influence of the prior knowledge on the performance of Semi-DEC, the radio of labelled training samples is increased from 1% to 50%. Each experiment is carried out 10 times, and the average results are shown in Table 4. And Table 5 shows the classification accuracy results produced by the same network architecture with Semi-DEC.

As shown in Tables 4 and 5 and Figure 4, there are two observations. First, the ACC and NMI results become larger in all three datasets as the number of labelled samples increases. Especially, the ACC and NMI can reach 97.5% and 95.2%, respectively, on the MNIST dataset with 50% labelled training images. Second, the clustering ACC of Semi-DEC on datasets with 50% labelled data is approximately equal to the classification ACC on the three datasets. Those observations indicate the outperformance of Semi-DEC.

In order to further test the method in this paper, we conducted experiments in many aspects, including the impact of different proportions of labelled data on performance, the change process of loss function and accuracy, and the effect of trade-off parameter λ on clustering performance and running time analysis.

Specifically, about the impact of different proportions of labelled data on performance, Figure 4 shows the trend of the accuracy of the clustering results on the MNIST, USPS, and REUSTER-10K datasets. The dotted line represents the classification accuracy results obtained through multiple experiments under the same network architecture with Semi-DEC. It can be more intuitively shown that with the gradual increase of the proportion of labelled data, the effect of Semi-DEC can be close to the classification effect in the MNIST and REUSTER-10K datasets. Although the clustering effect on the USPS dataset still has a certain gap with the classification effect, it is not far away.

The change process of the loss function and accuracy with the increase of training times is recorded in Figure 5. It can be seen that after reaching a certain number of iterations,

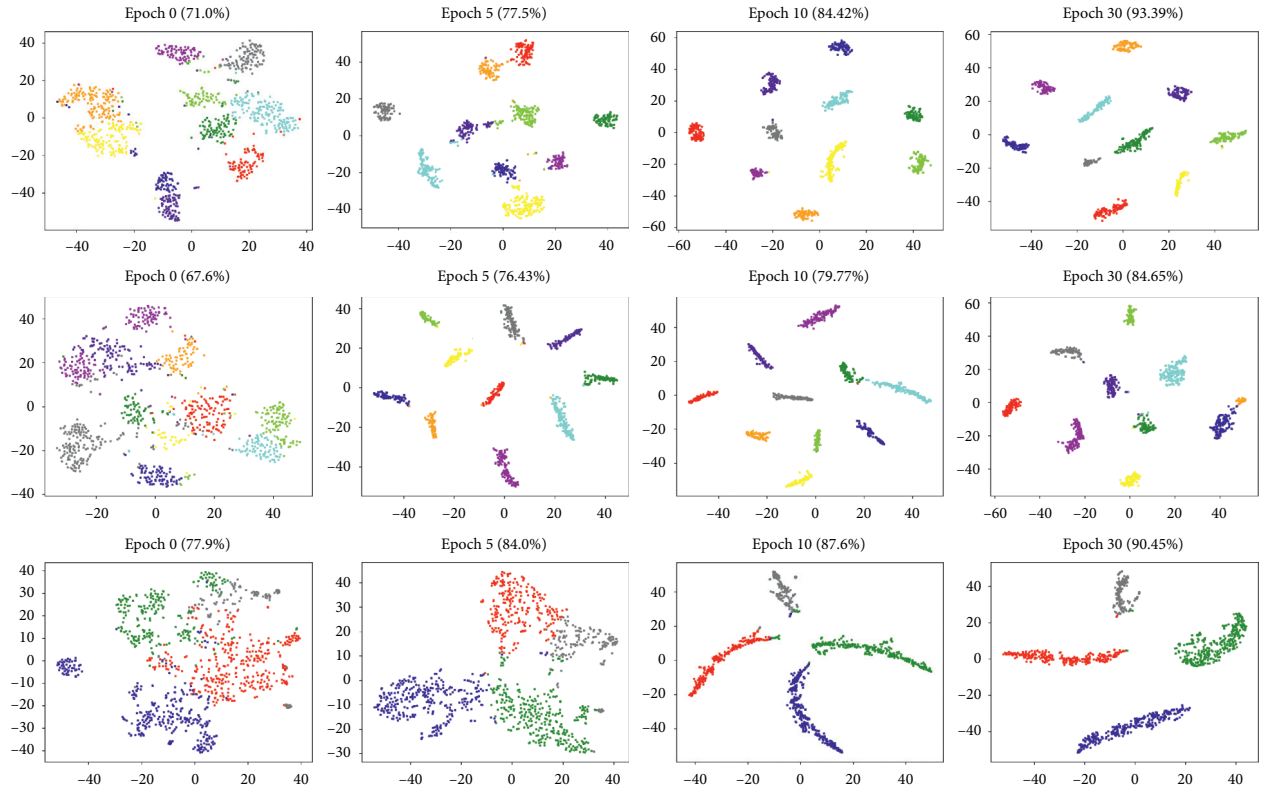

Figure 3: The visualization of clustering results during training on subset of MNIST, USPS, and REUTERS-10K from top to bottom. Different colours mark different clusters. The clustering accuracy of the corresponding epoch is given in parentheses. It can be seen that the data of the same class become more compact while the data of different classes are further away from each other as the number of epochs increases. This also shows that the learned feature embedding space is more and more suitable for clustering tasks.

Table 3: Clustering results measured by NMI.

Methods	MNIST	USPS	REUTERS-10K
K-means	0.4974	0.62	0.4932
DEC	0.8372	0.7529	0.4976
DCN	0.757	0.719	0.4106
IDEC	0.8672	0.7846	0.4981
SMKL	0.6842	0.7105	0.4076
SDEC	0.8289	0.7768	0.4762
Semi-DEC	**0.9457**	**0.8654**	**0.7642**

Table 4: Clustering results on datasets of various ratios of labelled data.

Datasets	1%		2%		5%		10%		20%		30%		40%		50%	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
MNIST	0.809	0.774	0.815	0.783	0.843	0.828	0.886	0.881	0.920	0.916	0.965	0.946	0.965	0.949	0.975	0.952
USPS	0.748	0.755	0.758	0.776	0.776	0.784	0.787	0.807	0.805	0.847	0.861	0.884	0.884	0.881	0.885	0.878
REUTERS-10K	0.751	0.506	0.758	0.519	0.769	0.554	0.795	0.586	0.863	0.68	0.918	0.764	0.954	0.829	0.956	0.831

the loss value and accuracy will tend to be stable, which is also a proof of the robustness of the method in this paper.

To see how the trade-off parameter λ of label loss affects the performance of the method in this paper, we conduct experiment on three datasets by sampling in range [0.01, 5.0]. Figure 6 gives the results. As shown in this figure, our method performs stably in a wide range of λ. The main reason is that the semisupervised loss dominates in this case. When λ is 0.2, the performance is asymptotically optimal.

About the running time, Figure 7 records the running time comparison between our method and DEC. Since the method in this paper is a further study on the basis of DEC, it only compares the running time with DEC. It can be seen

Table 5: Classification accuracy on the three datasets.

Datasets	MNIST	USPS	REUSTERS-10K
Average ACC	0.972	0.931	0.949

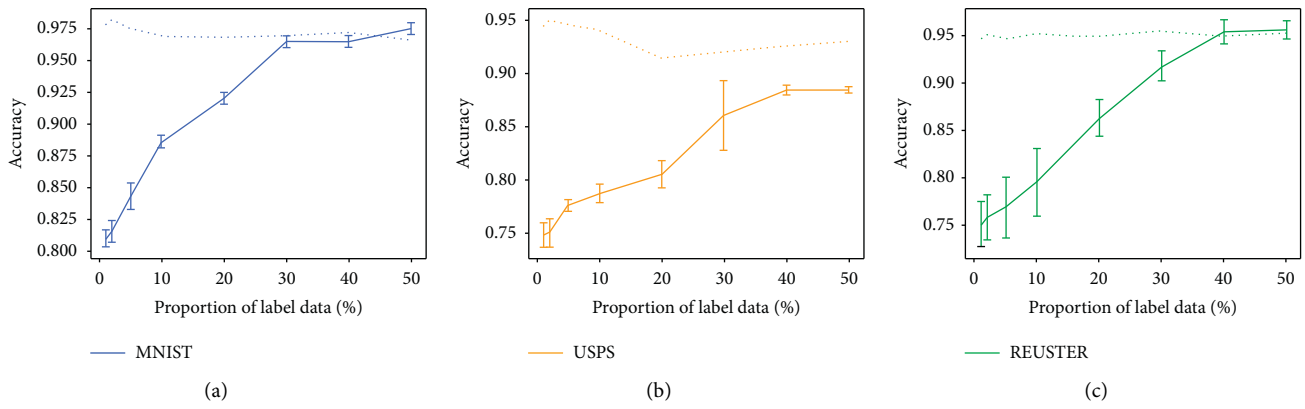

Figure 4: Accuracy of labelled data at different proportions on (a) MNIST, (b) USPS, and (c) REUTERS-10K.

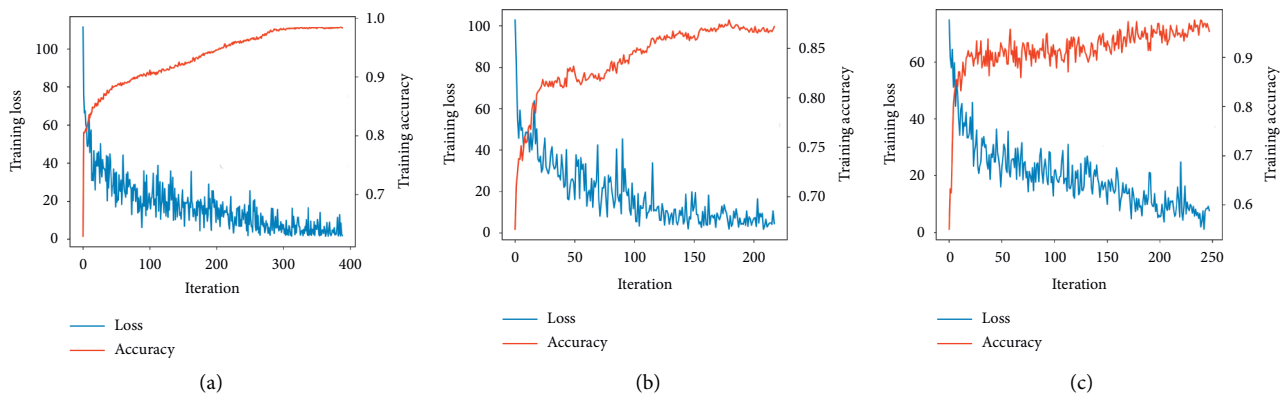

Figure 5: Trend of accuracy and loss with the number of iterations on (a) MNIST, (b) USPS, and (c) REUTERS-10K.

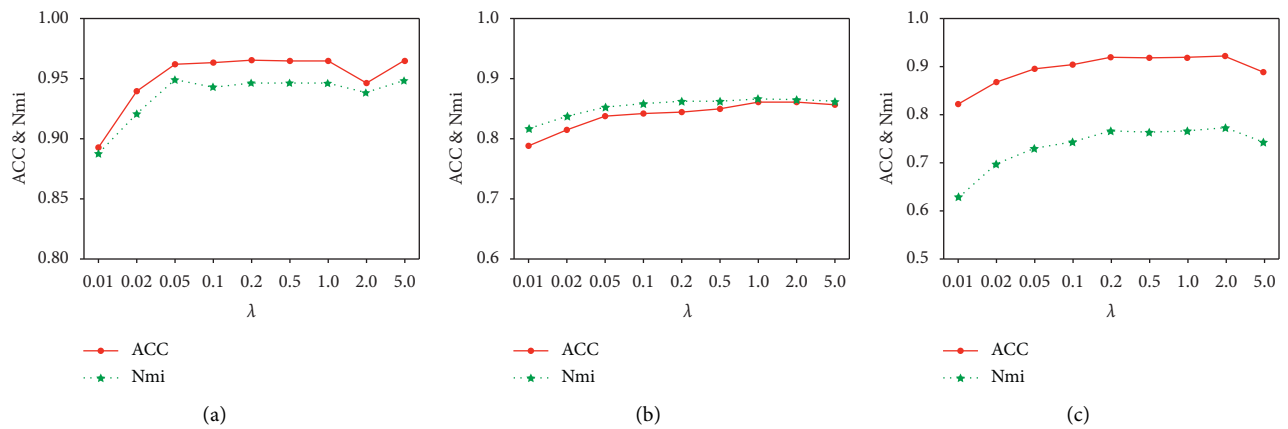

Figure 6: The effect of trade-off parameter λ on clustering performance on (a) MNIST, (b) USPS, and (c) REUTERS-10K.

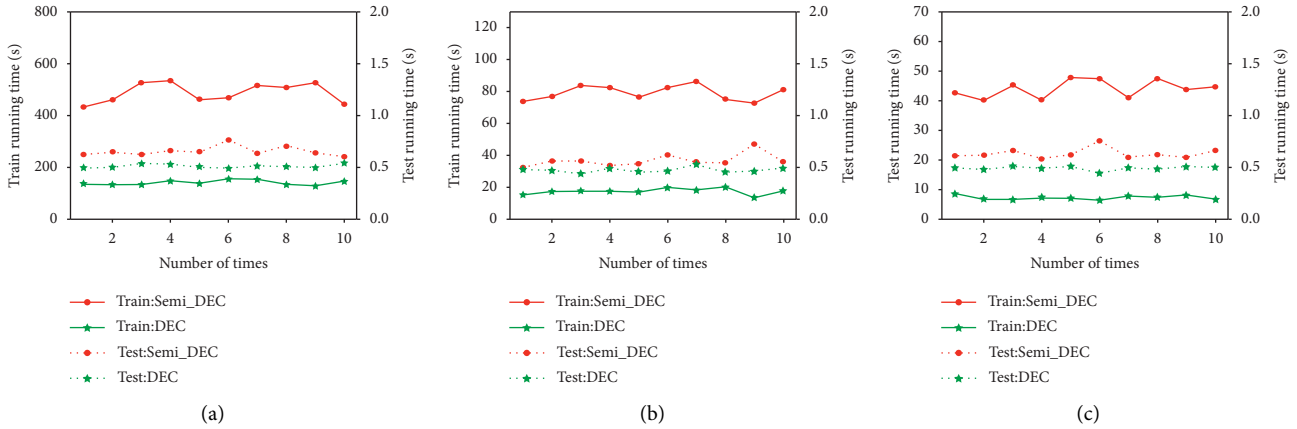

FIGURE 7: Running time statistics. The solid line represents the training process and dotted line represents the testing process. The circle represents the method in this paper, and the asterisk represents the DEC method. (a) MNIST, (b) USPS, and (c) REUTERS-10K.

that the method in this paper consumes more time in the training process than DEC. This is because the label adaptive strategy is added and the label loss needs to be calculated. But we think the limited time for training is worth it because we have got a big improvement in performance.

6. Conclusions

In this paper, a novel semisupervised deep embedded clustering method with adaptive labels is proposed to jointly learn cluster representation and assignment of data with the help of a priori knowledge. A deep semi-supervised clustering network is proposed, as well as a label adaptive strategy that can directly guide the clustering process by using the existing label information. Also, a joint optimization of the KL divergence loss and label loss in semisupervised deep clustering framework is designed to learn more powerful deep representation and more accurate cluster centres. Experimental results on MNIST, USPS, and REUSTER-10K show the method proposed in this paper has achieved significant performance improvement in both ACC and NMI, proving the effectiveness of the method. In the future, more efficient ways to use label information in the deep embedded clustering will be explored.

Data Availability

We perform experiment on two image datasets and one text dataset. The datasets used are commonly used public datasets, which are linked as follows: MNIST: http://yann.lecun.com/exdb/mnist/. USPS: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html. Reuters: http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61672123, Grant 61602083, and Grant 62002044, the Doctoral Scientific Research Foundation of Liaoning Province (20170520425), the Fundamental Research Funds for the Central Universities under Grant DUT20LAB136, Grant DUT20TD107, and Grant DUT15RC(3)100, and the China Scholarship Council.

References

[1] X. Li, H. Yin, K. Zhou, and X. Zhou, "Semi-supervised clustering with deep metric learning and graph embedding," *World Wide Web*, vol. 23, no. 2, pp. 781–798, 2020.

[2] Z. Lv, H. Song, P. Basanta-Val, A. Steed, and M. Jo, "Next-generation big data analytics: state of the art, challenges, and future research topics," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1891–1899, 2017.

[3] W. Wang, Y. Wu, C. Tang, and M. Hor, "Adaptive density-based spatial clustering of applications with noise (DBSCAN) according to data," in *Proceedings of the 2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 445–451, Guangzhou, China, 2015.

[4] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001.

[5] V. Bureva, E. Sotirova, S. Popov, D. Mavrov, and V. Traneva, "Generalized net of cluster analysis process using STING: a statistical information grid approach to spatial data mining," in *Proceedings of the 12th International Conference Flexible Query Answering Systems (FQAS)*, London, UK, 2017.

[6] B. Lv, W. Hou, G. Liu et al., "A deep cfs model for text clustering," in *Proceedings of the 2018 International Conference on Internet of Things*, pp. 132–137, Halifax, Canada, 2018.

[7] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, vol. 48, pp. 478–487, New York, NY, USA, 2016.

[8] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image

classification," *IEEE Transactions on Geoscience and Remote Sensing*, p. 1, 2020.

[9] L. Zhao, T. Zhao, T. Sun, Z. Liu, and Z. Chen, "Multi-view robust feature learning for data clustering," *IEEE Signal Processing Letters*, vol. 27, pp. 1750–1754, 2020.

[10] W. Fan, C. Wang, and J. Lai, "SDenPeak: semi-supervised nonlinear clustering based on density and distance," in *Proceedings of the 2016 International Conference on Big Data Computing Service and Applications*, pp. 269–275, Oxford, UK, 2016.

[11] H. Li, J. Zhang, G. Shi, and J. Liu, "Graph-based discriminative nonnegative matrix factorization with label information," *Neurocomputing*, vol. 266, pp. 91–100, 2017.

[12] X. Li, Y. Wu, M. Ester et al., "Semi-supervised clustering in attributed heterogeneous information networks," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 1621–1629, Perth, Australia, 2017.

[13] Z. Kang, X. Lu, J. Yi, and Z. Xu, "Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2312–2318, Stockholm, Sweden, 2018.

[14] D. Hong, N. Yokoya, G.-S. Xia, J. Chanussot, and X. X. Zhu, "X-ModalNet: a semi-supervised deep cross-modal network for classification of remote sensing data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 12–23, 2020.

[15] P. Li, Z. Chen, J. Gao et al., "A deep fusion Gaussian mixture model for multiview land data clustering," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8880430, 9 pages, 2020.

[16] P. Li, Z. Chen, L. T. Yang, L. Zhao, and Q. Zhang, "A privacy-preserving high-order neuro-fuzzy c-means algorithm with cloud computing," *Neurocomputing*, vol. 256, pp. 82–89, 2017.

[17] A. Saha and S. Das, "Clustering of fuzzy data and simultaneous feature selection: a model selection approach," *Fuzzy Sets and Systems*, vol. 340, pp. 1–37, 2018.

[18] T. Yuan, W. Deng, J. Hu, Z. An, and Y. Tang, "Unsupervised adaptive hashing based on feature clustering," *Neurocomputing*, vol. 323, pp. 373–382, 2019.

[19] S. Xiang, F. Nie, and C. Zhang, "Learning a Mahalanobis distance metric for data clustering and classification," *Pattern Recognition*, vol. 41, no. 12, pp. 3600–3612, 2008.

[20] N. Heidari, Z. Moslehi, A. Mirzaei, and M. Safayani, "Bayesian distance metric learning for discriminative fuzzy c-means clustering," *Neurocomputing*, vol. 319, pp. 21–33, 2018.

[21] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 1969–1976, Phoenix, AZ, USA, 2016.

[22] X. Wang, F. Nie, and H. Huang, "Structured doubly stochastic matrix for graph based clustering: structured doubly stochastic matrix," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1245–1254, San Francisco, CA, USA, 2016.

[23] D. Xie, Q. Gao, Q. Wang, and S. Xiao, "Multi-view spectral clustering via integrating global and local graphs," *IEEE Access*, vol. 7, pp. 31197–31206, 2019.

[24] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards K-means-friendly spaces: simultaneous deep learning and clustering," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 3861–3870, Sydney, Australia, 2017.

[25] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1753–1759, Melbourne, Australia, 2017.

[26] L. Zhao, T. Yang, J. Zhang, Z. Chen, Y. Yang, and Z. J. Wang, "Co-learning non-negative correlated and uncorrelated features for multi-view data," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2020.

[27] S. Anand, S. Mittal, O. Tuzel, and P. Meer, "Semi-supervised kernel mean shift clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1201–1215, 2014.

[28] C.-L. Liu, W.-H. Hsaio, C.-H. Lee, and F.-S. Gou, "Semi-supervised linear discriminant clustering," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 989–1000, 2014.

[29] D. Wang, X. Gao, and X. Wang, "Semi-supervised nonnegative matrix factorization via constraint propagation," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 233–244, 2016.

[30] V.-V. Vu, "An efficient semi-supervised graph based clustering," *Intelligent Data Analysis*, vol. 22, no. 2, pp. 297–307, 2018.

[31] A. Barhillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a Mahalanobis metric from equivalence constraints," *Journal of Machine Learning Research*, vol. 6, pp. 937–965, 2005.

[32] S. C. H. Hoi, W. Liu, M. R. Lyu, and W. Ma, "Learning distance metrics with contextual constraints for image retrieval," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2072–2078, New York, NY, USA, 2006.

[33] G. Niu, B. Dai, M. Yamada, and M. Sugiyama, "Information-theoretic semi-supervised metric learning via entropy regularization," *Neural Computation*, vol. 26, no. 8, pp. 1717–1762, 2014.

[34] L. Wu, S. C. H. Hoi, R. Jin, J. Zhu, and N. Yu, "Learning Bregman distance functions for semi-supervised clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 478–491, 2012.

[35] Z. Yu, L. Li, J. Liu, J. Zhang, and G. Han, "Adaptive noise immune cluster ensemble using affinity propagation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 12, pp. 3176–3189, 2015.

[36] J. Yi, L. Zhang, T. Yang, W. Liu, and J. Wang, "An efficient semi-supervised clustering algorithm with sequential constraints," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1405–1414, Sydney, Australia, 2015.

[37] S. Wei, Z. Li, and C. Zhang, "A semi-supervised clustering ensemble approach integrated constraint-based and metric-based," in *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service (ICIMCS)*, Zhangjiajie, China, 2015.

[38] D. Hong, L. Gao, N. Yokoya et al., "More diverse means better: multimodal deep learning meets remote-sensing imagery classification," *IEEE Transactions on Geoscience and Remote Sensing*, p. 1, 2020.

[39] G. Chen, "Deep transductive semi-supervised maximum margin clustering," 2015, https://arxiv.org/abs/1501.06237.

[40] Y. Ren, K. Hu, X. Dai, L. Pan, S. C. H. Hoi, and Z. Xu, "Semi-supervised deep embedded clustering," *Neurocomputing*, vol. 325, pp. 121–130, 2019.

[41] J. Gao, P. Li, Z. Chen, and J. Zhang, "A survey on deep learning for multimodal data fusion," *Neural Computation*, vol. 32, no. 5, pp. 829–864, 2020.