International Journal of Innovative
Computing, Information and Control
Volume **19**, Number **4**, August **2023**

ICIC International ©2023 ISSN 1349-4198
pp. **1057–1072**

SELF-WEIGHTED DEEP SUBSPACE CLUSTERING WITH FUZZY LABELS

Zhaoqiang Bao and Lihong Wang*

School of Computer and Control Engineering
Yantai University
No. 30, Qingquan Road, Laishan District, Yantai 264005, P. R. China
b_zhaoqiang@163.com; *Corresponding author: wanglh@ytu.edu.cn

Received November 2022; revised February 2023

Abstract. *In weakly-supervised learning, the background knowledge of applications could be given in the form of fuzzy labels, where experts are not sure about the exact class annotations for all data objects (e.g., images), but they can provide candidate labels for a subset of the objects. Subspace clustering aims to segment data objects into different clusters, each of which is related to a subspace. In this study, we propose a novel self-weighted deep subspace clustering algorithm, named DSCF, to learn the subspaces of datasets provided with a small number of fuzzy labels. In the proposed loss function, we consider the local relationship of data objects to reconstruct the input data and design the constraints on the self-weighted similarity of objects to embed the fuzzy labels. The self-expression matrix for final spectral clustering is learned by minimizing the loss function to jointly optimize the reconstruction error and the comprehensive constraints. The experiment results show that the proposed DSCF outperforms the compared algorithms on five gray-scale image datasets. Further experiments on the cardinality of fuzzy labels are conducted and the effectiveness of fuzzy label simulation is discussed.*

Keywords: Subspace clustering, Self-weighted, Fuzzy labels, Deep learning, Auto encoder

1. **Introduction.** In data-driven machine learning, data quality may substantially influence the performance of a data model. Exact and sufficient labeled data for learning models is critical. However, exact annotations of training data are challenging in real applications due to high costs; thus, incomplete, inaccurate or inexact labeled data have to be utilized to weakly supervise a model training [1, 2]. For example, in the underwater plankton image classification, it is difficult to give clean class boundaries due to a limited information content in the images; thus, different experts have different opinions and produce ambiguous or inexact labels [3]. In Figure 1, the experts are not sure about the annotation of the test sample, and both Collodaria and Phaeodarea could be considered as possible labels; then {Collodaria, Phaeodarea} is called a fuzzy label of the test sample in this study. Similarly, the handwritten digit images are confused sometimes, one could doubt about the images in Figure 2, since maybe the first two digits are "3" or "8", and the last two ones are "7" or "9", then $\{3, 8\}$ is a fuzzy label of the first two digits, and $\{7, 9\}$ is a fuzzy label of the last two ones.

In fact, the background knowledge of applications can be given in the form of fuzzy labels, where experts are not sure about the exact annotation of each sample, but they can provide candidate labels for a subset of the samples. Specifically, we assume that some data samples have fuzzy labels to represent the prior knowledge. The mathematical formulation of a fuzzy label would be an unknown soft probability distribution for its

DOI: 10.24507/ijicic.19.04.1057

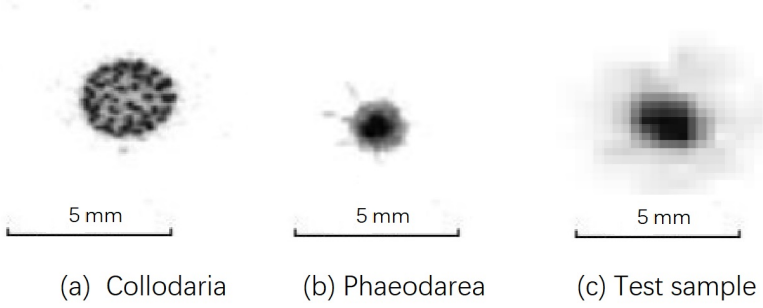


FIGURE 1. A test sample and its possible labels

FIGURE 2. Confused handwritten digits

candidate classes [3], so we approximate the weight of each label by averaging over the multiple annotations. In Figure 1, the weights of the sample are 0.5 for both Collodaria and Phaeodarea.

Fuzzy labels are closely related with coarse class labels, which are high level information of fine-grained labels. For example, "dog" is a coarse label of "husky" [4], and the coarse class "large carnivore" includes fine-grained classes bear, leopard, lion, tiger and wolf [5]. In this study, the fuzzy label of an image is a set of several potential fine-grained labels of the image to simulate the ambiguous expert annotations, which is neither a coarse class label nor a high-level abstraction of the potential labels. Additionally, fuzzy labels are different from the labels in multiple instance learning, where training instances are arranged in bags. Since in multiple instance learning, supervision is provided only for entire bags, and the individual labels of the instances contained in the bags are not provided, i.e., each bag has a unique label, which is positive if the bag contains at least one positive instance or negative if not [6]. In summary, the fuzzy label is a new way of inexact knowledge representation which is different from multi-grained learning and multiple instance learning, and how to combine it with learning process deserves careful study.

As a classical data analysis method, clustering has been widely used in computer, biology, economics and other fields. Traditional clustering methods, such as k-means, hierarchical clustering [7] and density clustering [8], cannot perform well on high-dimensional data. For high-dimensional data, clusters usually exist in subspaces of the entire data space [9]. Subspace clustering or subspace segmentation aims to segment data samples into different clusters, where each cluster is related to a subspace [10]. Therefore, the subspace clustering is proposed to find different subspaces and the clusters in subspaces for high-dimensional data.

Over the past decade, we have seen the successful applications of deep learning to many problems, such as near-human level image classification [11], automatic drive [12] and state diagnosis of machine [13]. In deep learning, the network transforms the input data through successive layers into representations that are increasingly purified and informative for a specific task. The combination of deep learning with subspace clustering attracts much attention. For example, autoencoder [14], a deep learning method, can be

used to non-linearly map the input data into a low-dimensional latent space for subspace clustering [15]. Efforts on the semi-supervised clustering show that the clustering performance on a dataset will be improved if the background knowledge of the dataset is efficiently used to lead the clustering process [16]. To the best of our knowledge, there is very little research on how to use fuzzy labels to improve the clustering performance, even if fuzzy labels are easier to be obtained than exact labels in reality.

In this study, we propose a novel deep subspace clustering algorithm to learn the subspaces for datasets provided with a small number of fuzzy labels. The self-expressive coefficient matrix C for final spectral clustering is learned by minimizing the loss function designed for jointly optimizing the reconstruction error and constraints on the matrix C. In the loss function, we consider both the local relationship of data points and the comprehensive constraint on the self-weighted similarity of instances with fuzzy labels.

The rest of this paper is organized as follows. In Section 2, we review the related work of deep subspace clustering algorithms. In Section 3, we propose our algorithm for deep subspace clustering with fuzzy labels. The experimental results are introduced in Section 4, and finally Section 5 summarizes the paper.

2. **Related Work.**

2.1. **Deep subspace clustering.**

In subspace clustering, self-expressiveness and subspace-preserving are two common assumptions used in current linear subspace clustering algorithms. Specifically, the self-expressiveness property assumes that each data point can be expressed as a linear combination of all the other points in the union of subspaces, and a representation of a self-expression matrix is called subspace-preserving if each data point in a subspace can be linearly expressed by other points in the same subspace [17]. Low-rank representation (LRR) [18] and sparse subspace clustering (SSC) [9] are two classical linear subspace clustering algorithms. These methods, however, do not perform satisfactorily in practical applications because they can only explore linear subspaces. Deep subspace clustering (DSC) [15], proposed by Ji et al., uses a convolutional autoencoder to perform nonlinear transformation of data points into the latent space. A fully connected self-expressive layer is added between the encoder and decoder to learn the nonlinear data mapping to the latent space in an unsupervised manner. Existing deep subspace clustering methods perform more robustly than traditional subspace clustering algorithms, benefiting from the powerful feature extraction capability of neural networks.

In the absence of data point labels, self-supervision and pseudo-supervision are two possible approaches to improving clustering performance. Self-supervised learning is an unsupervised approach that usually requires a pre-defined task whose objective function can be computed without supervised information, and self-supervised learning can obtain high-level semantic information needed for subsequent tasks such as classification. Zhang et al. proposed the self-supervised convolutional subspace clustering network S^2ConvSCN [19], which directly uses spectral clustering results as labels to supervise the learning of subspace clustering and deep networks in the absence of labeled data. Pseudo-supervised learning can be semi-supervised [20] or unsupervised [21]. Lee used both a small amount of labeled data and a large amount of unlabeled data to train the network and used the label prediction results of unlabeled points as real labels to find low-density separation boundaries between the classes, thus improving the generalization performance [20]. Lv et al. proposed pseudo-supervised deep subspace clustering algorithm PSSC [21], in which there are no labeled data and thus it is unsupervised learning. The PSSC network consists of a locality preserving module, a self-expression module, and a pseudo-supervision module. PSSC introduces pseudo-graph supervision and pseudo-label supervision to guide

the network training by constructing pseudo-graphs and pseudo-labels. In the network, a fully-connected layer with a softmax function is added after the encoder, which uses the the encoder output as the latent representation to construct pseudo-labels for the samples, and only samples with highly confident pseudo-labels contribute to network training. Finally, the self-expression matrix is used for spectral clustering on the dataset. In these efforts, the subspace clustering uses unlabeled data to train the network and regards the predicted labels of unlabeled points as real labels. Without a prior knowledge, the clustering process will find a data structure misrepresented by the data set if it is twisted due to noise or inefficient sample. A small amount of labeled data help to correct the process if provided. However, the labeled data need to be accurately annotated by human experts. In the cases that the experts are not sure about the annotation of a data sample due to the difficulty of recognition, we can reduce the annotation effort by replacing the exact annotation with possible ones, i.e., the fuzzy label. How to use fuzzy labels to lead the clustering process is worthy of careful study.

2.2. **Self-weighted clustering.** Self-weighted techniques balance the importance of two or more factors in clustering. For example, in multiview clustering, it is essential to assign a reasonable weight to each view according to the view importance [22, 23]. Nie et al. proposed a method to achieve clustering multiview data while learning the view weights by self-weighted clustering with multiple graphs [22]. Zhang et al. proposed self-weighted semi-supervised spectral clustering, which assigned different weights to the pairwise constraints (i.e., must links and cannot links). By transforming the given pairwise constraints into the intrinsic graph similarity and the penalty graph similarity respectively, the proposed algorithm achieves the optimal weight automatically to balance the graph optimization problems between the intrinsic graph and the penalty graph [24]. Additionally, self-weighted features can assign weights for features in clustering to adjust the importance of different features in practical applications, which is usually assumed to be equal [25, 26, 27]. Following this idea, we assume that the input data features are of different importance to the clustering, and the self-weighted feature term is introduced to the loss function to constrain the distances between similar points, and then feature weights are achieved in the optimization iterations of the proposed loss function.

3. **The Proposed DSCF.** In this section, we propose a self-weighted deep subspace clustering algorithm with fuzzy labels (DSCF). Firstly, a new pairwise similarity is proposed to embed the fuzzy labels, and then the loss function is designed to jointly minimize the reconstruction error of the network and the constraints on the self-expression matrix.

Table 1 shows the notations used in this paper.

Table 1. Notations used in this paper

Notation	Description
$X = \{X_1, X_2, \ldots, X_n\}$	Data set including n instances (points) with d dimensions.
K	Cluster number.
$Z_{m\times n}$	Low dimensional expression of X after encoder layers, $m \ll d$.
$C_{n\times n}$	Self-expressive coefficient matrix of Z, such that $Z = ZC$.
$D_{n\times n}$	Degree matrix of C, a diagonal matrix.
$S_{n\times n}$	Proposed pairwise similarity for n instances.
$P_{n\times K}$	Pseudo-label probability predicted by the softmax layer.
$Y_{n\times K}$	Fuzzy labels of data points.
θ	Self-weighted column vector for each dimension of input data.
L	Proposed loss function.

3.1. **A new pairwise similarity embedding fuzzy labels.** To make full use of the fuzzy labels, we define a new similarity between two points X_i and X_j as follows:

$$S_{ij} = C_{ij}^2 \left(Y_{i,:} Y_{j,:}^T \right), \tag{1}$$

where C_{ij} is an element of the coefficient matrix C, and $Y_{i,:}$ is the membership vector expressing the fuzzy label of point X_i.

For example, let X_i have a fuzzy label $\{1, 2, 3\}$, then $Y_{i,:} = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, \ldots, 0\right)$. The fuzzy label $\{1, 2, 3\}$ indicates the ground-truth label of X_i is one of the three class labels, but there is no information on the possibility distribution of its candidate class, and no label is more convincing than the others, so we approximate the weight of each label by averaging over the multiple annotations [3], and then the three labels share the same confidence.

The coefficient matrix C is crucial to the clustering results, since it represents the subspace structure of the dataset. In the view of similarity graph, $|C_{ij}|$ is the edge weight between X_i and X_j, and $C_{ij} = 0$ means that X_i and X_j are not in the same subspace. Obviously, S_{ij} is large if X_i and X_j are closely connected, i.e., C_{ij}^2 is large, and X_i and X_j have similar fuzzy labels. In the proposed algorithm, we will use this similarity to constrain the coefficient matrix C.

3.2. **The loss function.**

3.2.1. *The reconstruction error.* Let the data set $X = \{X_1, X_2, \ldots, X_n\}$ and the self-expressive coefficient matrix be $C = [C_{ij}]$. Let X_j be a neighbor point of X_i, and the similarity between X_i and X_j is expressed by the self-expressive coefficient C_{ij}. The decoded value $\hat{X}_j$ of the point X_j can be used to reconstruct the data point X_i, and the local relationship of data points is maintained by constraining the reconstruction error [21]. Inspired by the work of Zhou et al. [28], the reconstruction error L_1 is defined by adjusting each data point to its degree as follows:

$$\begin{aligned}
L_1 &= \sum_{i,j=1}^{n} C_{ij} \left\| \frac{X_i}{\sqrt{d_i}} - \frac{\hat{X}_j}{\sqrt{d_j}} \right\|^2 \\
&= \sum_{i,j=1}^{n} C_{ij} \left(\frac{X_i}{\sqrt{d_i}} - \frac{\hat{X}_j}{\sqrt{d_j}} \right)^T \left(\frac{X_i}{\sqrt{d_i}} - \frac{\hat{X}_j}{\sqrt{d_j}} \right) \\
&= \sum_{i=1}^{n} X_i^T X_i + \sum_{j=1}^{n} \hat{X}_j^T \hat{X}_j - 2 \sum_{i,j=1}^{n} \frac{X_i^T}{\sqrt{d_i}} C_{ij} \frac{\hat{X}_j}{\sqrt{d_j}} \\
&= Tr\left(X^T X\right) + Tr\left(\hat{X}^T \hat{X}\right) - 2Tr\left(X D^{-\frac{1}{2}} C D^{-\frac{1}{2}} \hat{X}^T\right) \\
&= Tr\left(\left(X - \hat{X}\right)\left(X - \hat{X}\right)^T\right) + 2Tr\left(X \hat{X}^T\right) - 2Tr\left(X D^{-\frac{1}{2}} C D^{-\frac{1}{2}} \hat{X}^T\right) \\
&= \left\| X - \hat{X} \right\|_F^2 + 2Tr\left(X L_{sym} \hat{X}^T\right),
\end{aligned} \tag{2}$$

where the degree matrix D is a diagonal matrix with elements on the diagonal $d_i = \sum_{j=1}^{n} C_{ij}$ and $L_{sym} = I - D^{-\frac{1}{2}} C D^{-\frac{1}{2}}$ is the normalized Laplacian matrix.

3.2.2. *Self-expressiveness term.* AE (autoencoder) is a feature extraction network structure commonly used in deep subspace clustering, which consists of a mutually symmetric encoder and decoder. The encoder extracts higher-level semantic feature representations $Z \in \mathbb{R}^{m \times n}$ from data $X \in \mathbb{R}^{n \times d}$, where $m \ll d$, and then the decoder restores it to its original form $\hat{X}$. The self-expressive layer is a full-connected layer between the encoder

and the decoder, which represents a sample point in a subspace by a linear combination of other sample points in the subspaces. The self-expressiveness property can be expressed as $Z = ZC$, which can be optimized by the following equation [21].

$$L_2 = \|C\|_p + \frac{1}{2}\|Z - ZC\|_F^2 \quad \text{s.t. } diag(C) = 0. \tag{3}$$

The constraint $diag(C) = 0$ is used to eliminate a trivial solution $C = I$, and force a sample point to be represented by other points.

3.2.3. *The consistency between fuzzy labels and pseudo-labels.* In addition to the fuzzy labels, pseudo-labels are utilized to supervise the network's training and further improve the clustering performance. As shown in the DSCF network structure in Figure 3, a softmax classification layer is introduced behind the encoder, which transforms Z into $P(Z)$, where $P(z_i) \in \mathbb{R}^K$ denotes the predicted label distribution of z_i, then

$$\sum_{k=1}^{K} P(z_i)_k = 1, \quad i \in 1, \ldots, n,\ P(z_i)_k \geq 0. \tag{4}$$

For convenience, $P(z_i)_k$ is denoted as $P_i^{(k)}$, which denotes the probability that point z_i belongs to cluster k. Following [29], we define L_3 to constrain the consistency between the fuzzy labels and pesudo-labels.

$$L_3 = -\sum_{i=1}^{n}\sum_{k=1}^{K} \left[h\left(Y_i \odot P_i\right)\right]^{(k)} \log\left(P_i^{(k)}\right), \tag{5}$$

where $\odot$ is the Hadamard product, and $h(.)$ is a function that maps a vector to its corresponding one-hot vector by assigning 1 to the maximum and 0 to the other elements, and the superscript (k) indicates the k-th element of the vector. The minimization of L_3 tries to make the fuzzy label be consistent with the pseudo-label predicted by the softmax layer for each point z_i, and meanwhile, maximize the probability that point z_i belongs to its predicted cluster k.

3.2.4. *Self-weighted clustering with the proposed pairwise similarity.* In the previous studies, most of the features of the samples were assumed equally important by default. However, not all features are equally important for the sample. For example, on a face, the features of the five senses (eyebrows, eyes, nose, ears, and mouth) are more important than the other features and should be assigned higher weights. However, manually assigning weights to all features is a challenging problem, so we define adaptive weights θ to assign weights to the features of each sample. Nie et al. assigned weights for features automatically in clustering to adjust the importance of different features in practical application [25, 27]. We evaluate the similarity of two data points with the proposed S_{ij}, and minimize the loss function L_4 by self-weighting the features to make sure that the closer similarity of two points, the smaller distance between them.

$$L_4 = \sum_{i,j=1}^{n} \|\theta X_i - \theta X_j\|_2^2 S_{ij} \quad \text{s.t. } \theta^T \mathbf{1} = 1. \tag{6}$$

Finally, we combine loss functions L_1, L_2, L_3 and L_4 in a unified framework, i.e., the final loss function includes four parts, corresponding to the reconstruction error, self-expressiveness property, consistency between the fuzzy labels and pseudo-labels and the new pairwise similarity respectively. Therefore, the final objective function of DSCF is as follows:

$$L = \left\|X - \hat{X}\right\|_F^2 + 2Tr\left(XL_{sym}\hat{X}^T\right) + \gamma_1\|Z - ZC\|_F^2$$
$$-\gamma_2 \sum_{i=1}^{n}\sum_{k=1}^{K}\left[h\left(Y_i \odot P_i\right)\right]^{(k)} \log\left(P_i^{(k)}\right) + \gamma_3 \sum_{i,j=1}^{n}\|\theta X_i - \theta X_j\|_2^2 S_{ij}$$
$$\text{s.t. } \theta^T\mathbf{1} = 1, \quad diag(C) = 0, \tag{7}$$

where γ_1, γ_2 and γ_3 are tradeoff factors. In the objective function, we remove the first term of L_2 to avoid too many constraints on the matrix C.

For convenience, we move the constraint on θ to the function and rewrite the loss function as follows:

$$L = \left\|X - \hat{X}\right\|_F^2 + 2Tr\left(XL_{sym}\hat{X}^T\right) + \gamma_1\|Z - ZC\|_F^2$$
$$-\gamma_2 \sum_{i=1}^{n}\sum_{k=1}^{K}\left[h\left(Y_i \odot P_i\right)\right]^{(k)} \log\left(P_i^{(k)}\right) + \gamma_3\left(\sum_{i,j=1}^{n}\|\theta X_i - \theta X_j\|_2^2 S_{ij} + \left|\theta^T\mathbf{1} - 1\right|\right)$$
$$\text{s.t. } diag(C) = 0. \tag{8}$$

Similar to the work of Ji et al. [15] and Lv et al. [21], we implement the network using neural network frameworks and train it by the back-propagation algorithm for each dataset to be clustered. Once the network architecture is optimized, we obtain the lower-dimensional representation Z and the coefficient matrix C.

Following the traditional subspace clustering algorithms, we construct the affinity matrix $W = \frac{|C|+|C^T|}{2}$ using the obtained coefficient matrix C, and then input W to the spectral clustering algorithm to finish the final clustering.

3.3. **The framework of DSCF.** The framework of the proposed DSCF is shown in Figure 3. Given a dataset X to be clustered, a small number of sample data points

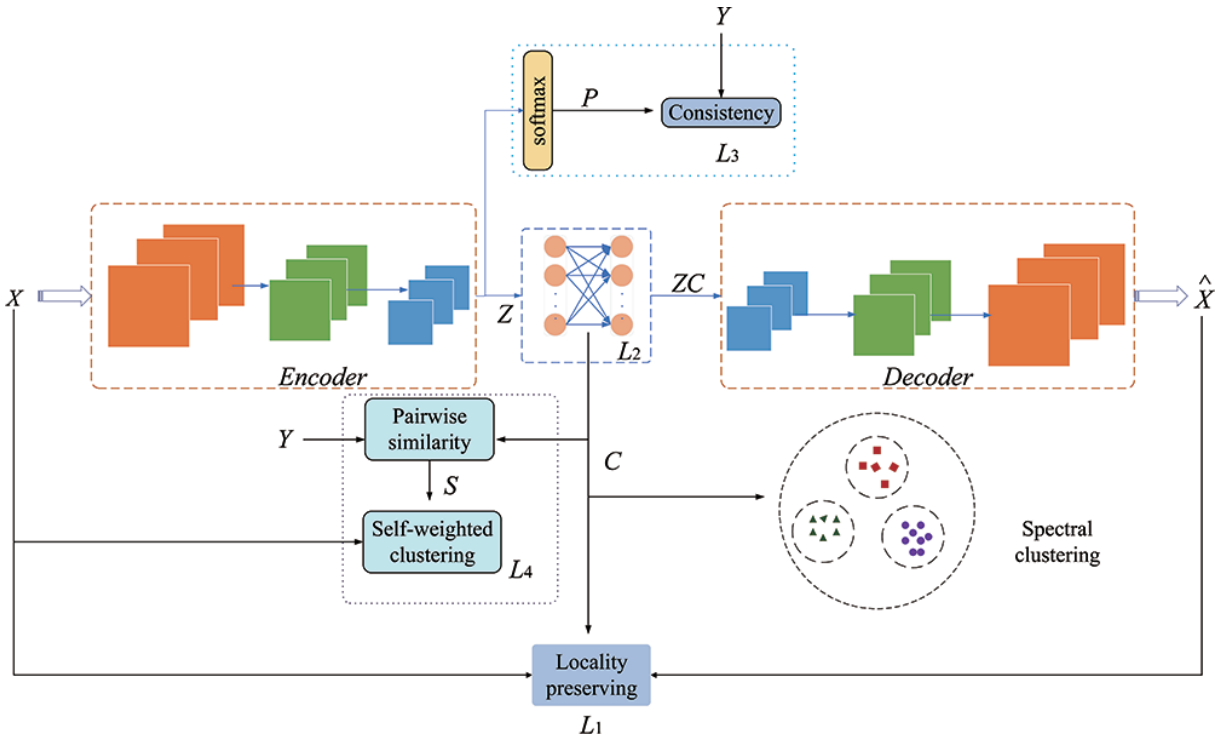


FIGURE 3. The framework of DSCF

are randomly selected from the dataset, and fuzzy labels Y are randomly assigned to selected points to simulate the inexact expert annotations. Then, the whole dataset is input to the encoder layers to abstract high level information, i.e., the lower-dimensional representation Z, and the fuzzy labels are used to constrain the output of the softmax layer, and jointly constrain the coefficient matrix C by the loss function L_4. Finally, the reconstructed input $\hat{X}$ is fed back to constrain the reconstruction error and the locality relationship. Figure 3 shows four parts of DSCF, where L_3 and L_4 are specially designed in this study to embed the fuzzy labels into the clustering process. L_3 tries to constrain the fuzzy label to be consistent with the pseudo-label predicted by the softmax layer, and L_4 self-weights the features of input data to make sure that the distance between two similar points is small.

The details of the DSCF algorithm are shown in Algorithm 1. Firstly, the network parameters of the AE are randomly initialized, and then all data points are fed to the AE without the self-expressive layer and the softmax layer to pre-train the auto encoder (lines 1-2). Secondly, the self-expressive layer parameters and the self-weighted parameters θ are initialized (lines 3-4), and then the whole DSCF network is trained in iterations and all parameters are optimized by Adam optimizer [30] to minimize the objective function L (lines 5-8). Finally, the affinity matrix $W = \frac{1}{2}\left(|C| + |C|^T\right)$ is calculated based on the obtained coefficient matrix C and fed to the spectral clustering to obtain the clustering result G (lines 9-10). In DSCF, the fuzzy labels Y simulates the inexact knowledge of the dataset, and combines with the self-expressive coefficients to evaluate the similarity of two points; meanwhile, the parameter θ weights the features of the input data and further combines with the similarity of points to calculate the loss function L_4. At the end of each iteration, the loss function L is calculated and optimized for an optimal solution of the clustering.

Algorithm 1. DSCF

Input: Dataset X, fuzzy labels Y, the tradeoff factors γ_1, γ_2, γ_3, the number of clusters K
Output: The clustering result G

1: Randomly initialize the AE network parameters
2: Pre-train the AE network
3: Randomly initialize the self-expressive layer parameters
4: Initialize the self-weighted parameters θ
5: **while** Not reach the maximum number of training **do**
6: Calculate the loss function L in Equation (8) using fuzzy labels Y and θ
7: Optimize the function L and update the network parameters with Adam optimizer
8: **end while**
9: Calculate the affinity matrix $W = \frac{1}{2}\left(|C| + |C|^T\right)$ using the coefficient matrix C
10: Feed the affinity matrix W to spectral clustering to obtain the clustering result G
11: **return** G

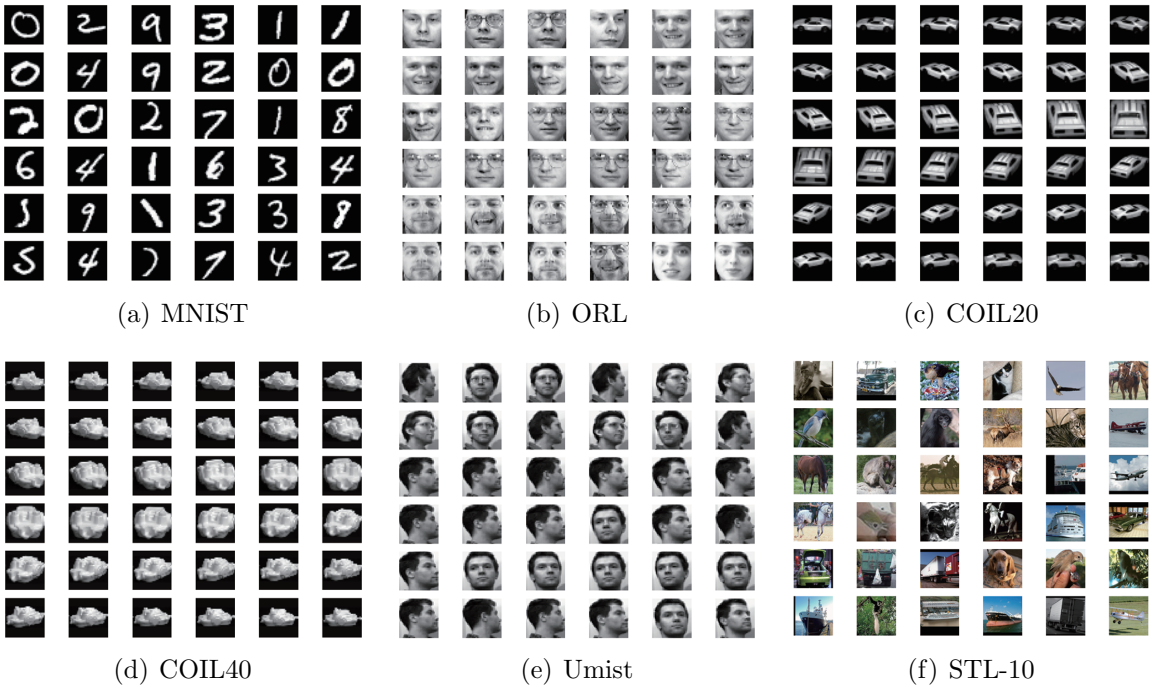
4. Experiments.

4.1. Datasets and evaluation metrics.

In order to test the proposed DSCF, we carry out experiments on 6 benchmark datasets for subspace clustering. The detailed informationed of each data set is shown in Table 2. Five of them are commonly used gray-scale image datasets, including MNIST for handwritten digits, ORL and Umist for face images, COIL20 and COIL40 for object datasets, and one color image dataset STL-10 with 10 classes. Some sample images are shown in Figure 4.

TABLE 2. Information of the 6 datasets

Datasets	Samples	Classes	Dimensions
MNIST	1000	10	28×28
ORL	400	40	32×32
COIL20	1440	20	32×32
COIL40	2880	40	32×32
Umist	480	20	32×32
STL-10	5000	10	64×64



(a) MNIST (b) ORL (c) COIL20

(d) COIL40 (e) Umist (f) STL-10

FIGURE 4. Some images of 6 benchmark datasets

1) MNIST: This dataset contains handwritten digits in 10 classes from 0 to 9, each class contains 6000 training images and 1000 test images of size 28×28. We use the first 100 ones of each class, totally 1000 images.
2) ORL: The dataset consists of face images of 40 individuals with 10 images of each person, and the images were taken with variations in light conditions and facial expressions. All images are of 32×32 pixels.
3) Umist: This dataset contains 480 face images, which are face photos taken by 20 people in different poses, with 24 gray-face images for each person. All images are down-sampled to 32×32 pixels.
4) COIL20 and COIL40: COIL20 contains 1440 gray-scale images of 20 different shaped objects, while COIL40 consists of 2880 gray scale images of 40 different shaped objects. All images are of 32×32 pixels.
5) STL-10: This dataset contains color images of 10 classes, including airplane, bird, car, cat, dog, deer, horse, monkey, ship, and truck. Each class consists of 500 images of 96×96 pixels. Histogram of oriented gradient (HOG) [31] is adopted for feature extraction of these color images and then each image is represented as 64×64 pixels.

Two evaluation metrics are used to compare the experiments results: clustering accuracy (ACC) [32], and normalized mutual information (NMI) [33]. Clustering accuracy of

clustering results is defined as

$$ACC = 1 - \frac{\sum_{i=1}^{n}\{B_i \neq map(Q_i)\}}{n}, \tag{9}$$

where B_i denotes the ground truth label of the i-th sample, and $map(Q_i)$ represents the label mapped by the i-th sample clustering result Q_i.

NMI is used to measure the mutual information entropy between the labels classified by clustering results and the ground truth, and defined as follows:

$$NMI(B, Q) = \frac{2 \times M(B, Q)}{E(B) + E(Q)} \tag{10}$$

where B is the true classes of sample points, and Q is the clustering labels. $M(.,.)$ is the mutual information, and $E(.)$ is the entropy.

Both ACC and NMI range in $[0, 1]$, and the larger the values, the better the clustering results.

4.2. **Experimental setup.** The proposed DSCF is compared with some common subspace clustering algorithms, including sparse subspace clustering (SSC) [9], low-rank representation (LRR) [18], deep embedding clustering (DEC) [34], DSC with ℓ_1 norm (DSC-L1), DSC with ℓ_2 norm (DSC-L2) [15], and pseudo-supervised deep subspace clustering (PSSC) [21]. The ablation experiments are performed by removing L_3 or L_4 in Equation (8) to test their effects on the performance respectively.

First, the AE is pre-trained without the self-expressive layer and the softmax classification layer, and the network architecture information is shown in Table 3. Then the entire network is fine-tuned by adding the self-expressive layer and the softmax classification layer. The learning rate is set to 1.0×10^{-3} for the pre-training phase and 1.0×10^{-4} for the fine-tuning phase [21]. The learning rate is important to the efficiency of iterations, since the larger rate in pre-training phase helps the network to converge quickly from scratch, while smaller rate in fine-tuning phase gradually searches and converges to the (sub)optimal solution. The adaptive momentum-based gradient descent method Adam is used to minimize the loss function.

TABLE 3. Network settings for datasets

Dataset	Encoder-1/Decoder-3	Encoder-2/Decoder-2	Encoder-3/Decoder-1	C
MNIST	$5 \times 5 \times 20$	$3 \times 3 \times 10$	$3 \times 3 \times 5$	1000×1000
ORL	$5 \times 5 \times 5$	$3 \times 3 \times 3$	$3 \times 3 \times 5$	400×400
COIL20	$3 \times 3 \times 15$	–	–	1440×1440
COIL40	$3 \times 3 \times 20$	–	–	2880×2880
Umist	$5 \times 5 \times 15$	$3 \times 3 \times 10$	$3 \times 3 \times 5$	480×480
STL-10	$3 \times 3 \times 15$	–	–	5000×5000

4.3. **Experimental results and study of ablation.** Tables 4 and 5 record the ACC and NMI of DSCF with the other 6 subspace clustering algorithms on the 6 benchmark datasets, respectively. The ratio 5% in DSCF(5%) denotes that 5% data points of the entire dataset are randomly sampled, and 3 random labels are assigned to each of these points. DSCF(10%) is defined similarly. The cardinality and effectiveness of fuzzy labels are discussed in the following subsection.

For each pair of dataset and algorithm, the best values of ACC and NMI are shown in Tables 4 and 5, respectively. The best values for each dataset are shown in bold, and the second-best ones are shown in italics. We observe that, in the view of ACC and NMI,

TABLE 4. ACC of the seven algorithms on 6 datasets

Methods\Datasets	MNIST	ORL	COIL20	COIL40	Umist	STL-10
SSC	0.4530	0.7425	0.8631	0.7191	0.6904	0.7508
LRR	0.5386	0.8100	0.8118	0.6493	0.6979	0.7752
DEC	0.6120	0.5175	0.7215	0.4872	0.5521	0.8604
DSC-L1	0.7280	0.8550	0.9314	0.8003	0.7242	*0.9340*
DSC-L2	0.7500	0.8600	0.9368	0.8075	0.7312	**0.9342**
PSSC	*0.8430*	0.8675	0.9722	0.8358	0.7917	0.9178
DSCF_l	0.7770	0.8500	0.9736	0.8354	*0.8208*	0.9194
DSCF_u	0.7880	0.8825	0.9743	0.8392	*0.8208*	0.9196
DSCF(5%)	0.8330	**0.8900**	*0.9757*	*0.8444*	*0.8208*	0.9202
DSCF(10%)	**0.8440**	*0.8850*	**0.9764**	**0.8448**	**0.8333**	0.9204

TABLE 5. NMI of the seven algorithms on 6 datasets

Methods\Datasets	MNIST	ORL	COIL20	COIL40	Umist	STL-10
SSC	0.4709	0.8459	0.8892	0.8212	0.7489	0.7126
LRR	0.5632	0.8603	0.8747	0.7828	0.7630	0.7687
DEC	0.5743	0.7449	0.8007	0.7417	0.7125	0.8280
DSC-L1	0.7217	0.9023	0.9353	0.8852	0.7556	*0.8757*
DSC-L2	0.7319	0.9034	0.9408	0.8941	0.7662	**0.8760**
PSSC	0.7676	0.9349	0.9779	0.9258	0.8670	0.8535
DSCF_l	0.7332	0.9190	0.9774	0.9310	*0.8784*	0.8596
DSCF_u	0.7456	0.9298	0.9780	0.9348	*0.8784*	0.8599
DSCF(5%)	*0.7691*	*0.9353*	*0.9786*	*0.9403*	*0.8784*	0.8606
DSCF(10%)	**0.7707**	**0.9386**	**0.9791**	**0.9412**	**0.8835**	0.8606

DSCF with 5% fuzzy labeled data performs significantly better than the other compared methods on 5 datasets. Additionally, due to more sample points, DSCF(10%) has slightly better or equal performance on 5 and 6 datasets in terms of ACC and NMI respectively when compared with DSCF(5%).

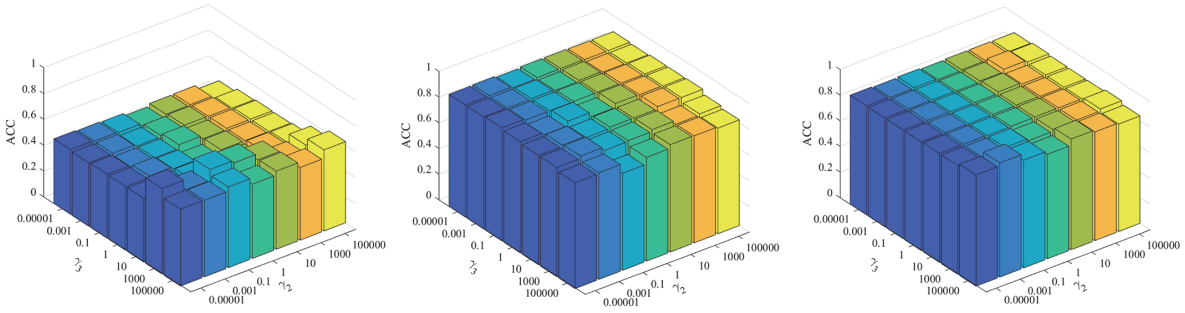
Furthermore, we carry out an ablation study to verify the two critical parts of the proposed DSCF (i.e., the consistency term L_3 and the self-weighted term L_4 in Equation (8)). For simplicity, the objective functions of DSCF with only L_3 and L_4 are denoted as DSCF_l and DSCF_u, respectively. As shown in Tables 4 and 5, the performance of DSCF_l and DSCF_u are worse than DSCF, which confirms the necessity of both L_3 and L_4.

4.4. **Parameter analysis.** The objective function L of DSCF has 3 hyperparameters, named γ_1, γ_2 and γ_3. To test the effects of parameters on DSCF, we use grid search on $\left\{10^{-5}, 10^{-3}, 10^{-1}, 1, 10, 10^3, 10^5\right\}$ to find the optimal parameters for each data set, and the optimal hyperparameters for all datasets are listed in Table 6. In the meantime, we fix γ_1 to show the effects of γ_2 and γ_3 on ACC and NMI for different datasets, and the results on the datasets ORL and COIL20 are shown in Figures 5-8. We observe that when γ_1 is fixed, the performance of DSCF is less sensitive to γ_2 and γ_3, which shows that DSCF achieves stable performance on the test cases. Other datasets support the same conclusion.

4.5. **Simulation of fuzzy labels and its effectiveness.** In the partial loss function L_4, the proposed pairwise similarity S_{ij} is used to tune automatically the weight vector θ for dimensions in the input data space. From the definition of S_{ij}, it is easy to find that

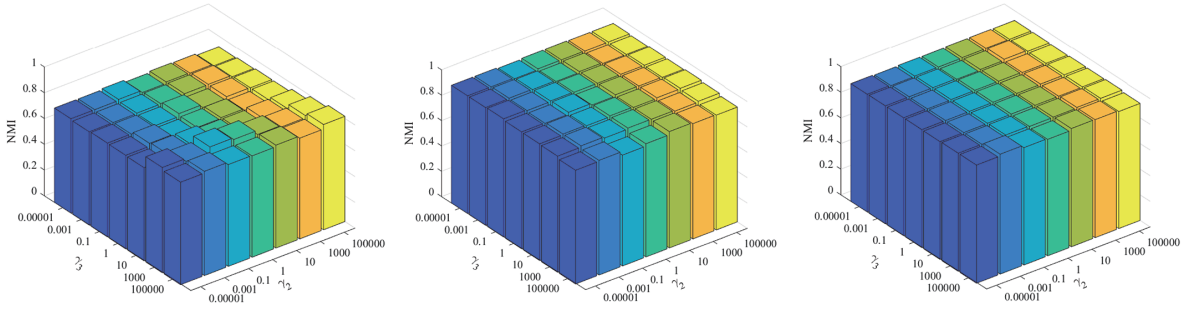
TABLE 6. Hyperparameters for DSCF on 6 datasets

Hyperparameters	γ_1	γ_2	γ_3
MNIST	1	10	10
ORL	10^5	10^5	10^{-3}
COIL20	10	10^3	10^5
COIL40	1	10^5	10^{-3}
Umist	10	1	10^{-5}
STL-10	10^3	10^{-1}	10^{-5}



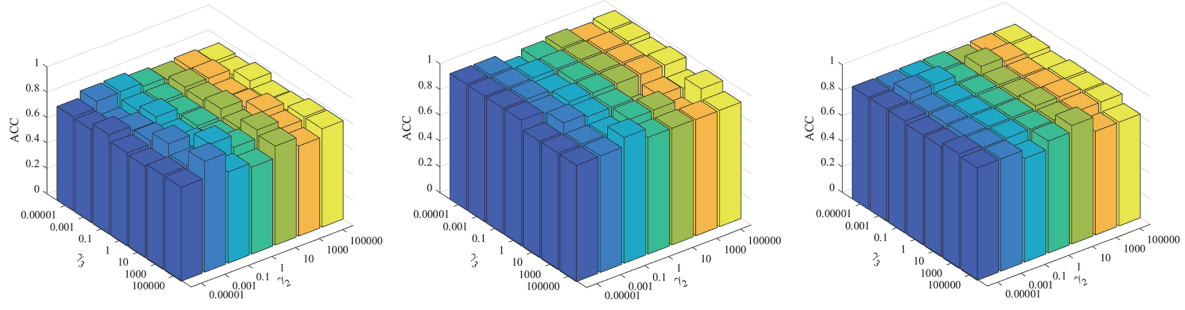
(a) $\gamma_1 = 10$ (b) $\gamma_1 = 1000$ (c) $\gamma_1 = 100000$

FIGURE 5. The effects of parameters on ACC of ORL dataset



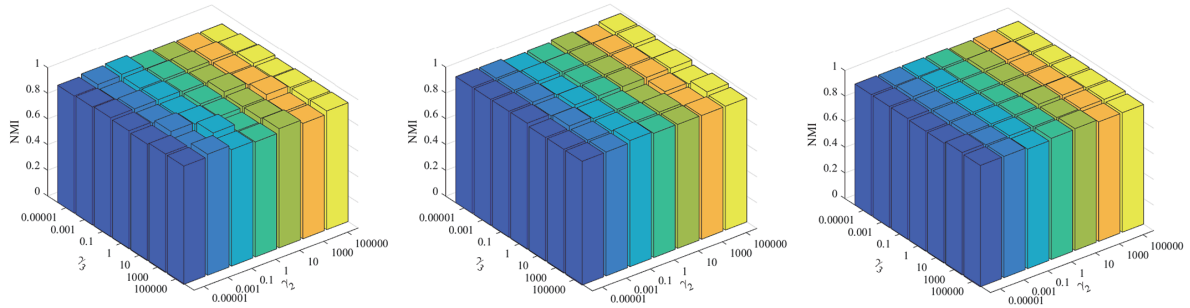
(a) $\gamma_1 = 10$ (b) $\gamma_1 = 1000$ (c) $\gamma_1 = 100000$

FIGURE 6. The effects of parameters on NMI of ORL dataset



(a) $\gamma_1 = 1$ (b) $\gamma_1 = 10$ (c) $\gamma_1 = 1000$

FIGURE 7. The effects of parameters on ACC of COIL20 dataset



(a) $\gamma_1 = 1$ (b) $\gamma_1 = 10$ (c) $\gamma_1 = 1000$

FIGURE 8. The effects of parameters on NMI of COIL20 dataset

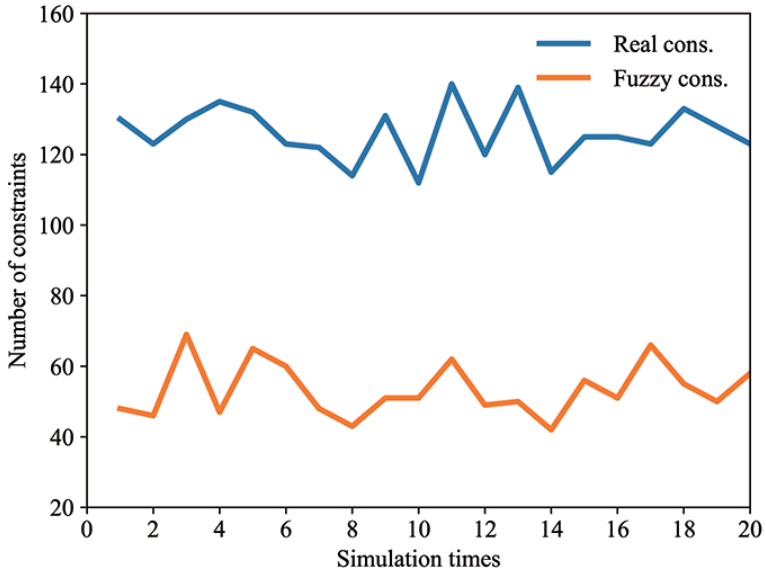


FIGURE 9. The number of pairwise constraints

a closer similarity is achieved if X_i and X_j have a larger C_{ij}^2 and a larger overlap of fuzzy labels simultaneously. The simulation of fuzzy labels should preserve the overlap of fuzzy labels to a certain extent for the instances from the same real class. In this section, we show the effectiveness of our simulation by taking the dataset COIL20 as an example.

We record the number of pairwise constraints in each simulation, and compare it to the real constraints. We randomly select a small number of sample points, e.g., 5% of the entire dataset, and assign 3 labels randomly for each of them. For example, we get 4 sample points from the instances with real class label 0, and denote them as A, B, C and D. We assign random fuzzy labels $\{17, 2, 11\}$, $\{17, 11, 15\}$, $\{2, 3, 4\}$ and $\{9, 19, 18\}$ for them. As far as the overlap is concerned, the point A has common labels 17 and 11 with the point B, and has the common label 2 with the point C. Thus, in this simulation, we retain 2 pairwise constraints, i.e., (A, B) and (A, C), for these 4 sample points, which have 6 real constraints, i.e., (A, B), (A, C), (A, D), (B, C), (B, D) and (C, D), since they have the same real class labels. Collecting pairwise constraints for all classes, i.e., from 0 to 19 in the COIL20 dataset, we obtain the number of real constraints and that of fuzzy constraints in the current simulation. We run the simulation program 20 times and illustrate the number of constraints in Figure 9. The simulation shows that on the average,

randomly assigned fuzzy labels for the selected sample points can preserve approximate 42% of all real constraints among the samples.

Furthermore, we test the performance of DSCF on different cardinalities of fuzzy labels, as shown in Figure 10. In Figure 10, the case that the cardinality of fuzzy labels equals 1 corresponds to that each sample point is armed with its real class label. From the curves of performance, we observe that the fuzzy labels with small cardinality have approximately the same performance as the real ones. With the increase of the cardinality, the performance decreases on 4 datasets significantly. The more labels, the more opportunity that the instance will be clustered with instances from other classes. As the datasets Umist and STL-10 are concerned, the best values of the parameter γ_3 for them are all 10^{-5} in Table 6, which weaken the values of L_4 in the loss function L; thus, the influence of cardinality of fuzzy labels is not significant in the curves.

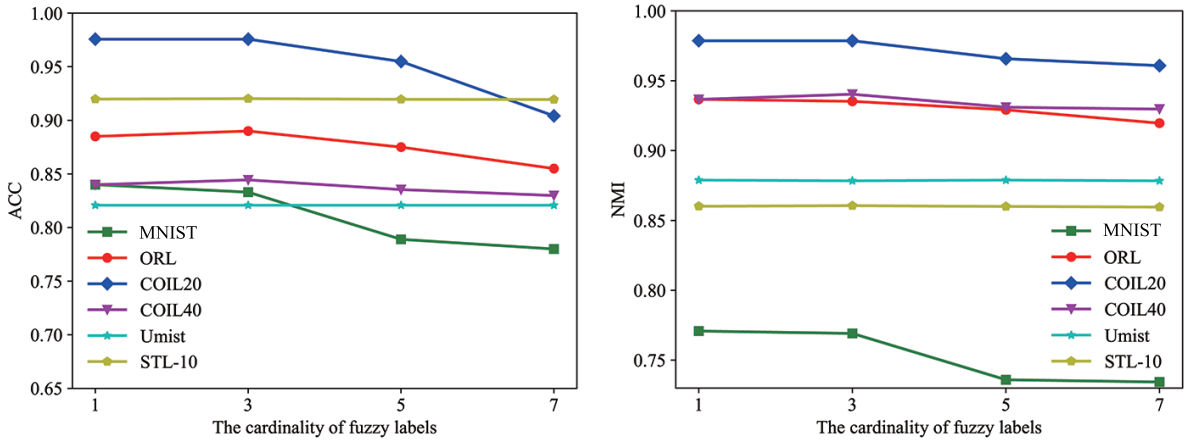


FIGURE 10. Influence of the cardinality of fuzzy labels

4.6. **Applications.** Fuzzy labels are alternatives to ground-truth class labels in real applications when the experts are not sure about the annotations of the train samples. A small number of fuzzy labels will help to improve the clustering accuracy of datasets based on the proposed DSCF algorithm. In this study, we have shown some potential applications of DSCF, such as object clustering (COIL20 and COIL100), face clustering (ORL and Umist) and handwritten digit clustering (MNIST). Moreover, in the field of person reidentification (Re-ID), the exact annotation of each image in the data sets, e.g., CUHK03 and Market-1501, is critical for the identification task. To alleviate the annotation burden for supervised learning, Wang et al. replaced the exact annotations with inexact ones by grouping the images into bags in terms of time and assigned a bag-level label for each bag, which includes all the annotations of individuals in this bag [29]. This greatly reduces the annotation effort, and the images in each bag share the same label. Since each individual has no exact annotation but a fuzzy label, the image bags can also be considered as inexact data, and DSCF can be applied to Re-ID by clustering on the image bags with fuzzy labels.

5. **Conclusion.** In this paper, we propose a self-weighted deep subspace clustering algorithm to deal with the embedding of knowledge provided in the form of fuzzy labels. By minimizing the proposed loss function, we embed the fuzzy labels to constrain the pseudo-labels generated by the softmax layer and the pairwise similarity between data objects. Experiments on six image datasets show that a small number of fuzzy labels help to improve the performance of subspace clustering. However, the number of fuzzy labels

is limited due to the heavy work of annotations in the real applications. This stimulates us to investigate an expanding method of the fuzzy labels, such as data augmentation, to improve the efficiency of fuzzy labels in the future work.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant 62072391 and Grant 72072154. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the representation.

REFERENCES

[1] Z. H. Zhou, A brief introduction to weakly supervised learning, *National Science Review*, vol.5, no.1, pp.44-53, 2018.

[2] D. G. Hao, L. Zhang, J. Sumkin et al., Inaccurate labels in weakly-supervised deep learning: Automatic identification and correction and their impact on classification performance, *IEEE Journal of Biomedical and Health Informatics*, vol.24, no.9, pp.2701-2710, 2020.

[3] L. Schmarje, J. Brünger, M. Santarossa et al., Fuzzy overclustering: Semi-supervised classification of fuzzy labels with overclustering and inverse cross-entropy, *Sensors*, vol.21, no.19, 2021.

[4] Y. H. Xu, Q. Qian, H. Li et al., Weakly supervised representation learning with coarse labels, *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp.10593-10601, 2021.

[5] H. Touvron, A. Sablayrolles, M. Douze et al., Grafit: Learning fine-grained image representations with coarse labels, *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp.874-884, 2021.

[6] M. Carbonneau, V. Carbonneau, E. Granger et al., Multiple instance learning: A survey of problem characteristics and applications, *Pattern Recognition*, vol.77, pp.329-353, 2018.

[7] S. C. Johnson, Hierarchical clustering schemes, *Psychometrika*, vol.32, no.3, pp.241-254, 1967.

[8] H. P. Kriegel, P. Kröger, J. Sander and A. Zimek, Density-based clustering, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol.1, no.3, pp.231-240, 2011.

[9] E. Elhamifar and R. Vidal, Sparse subspace clustering, *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, pp.2790-2797, 2009.

[10] H. D. Zhao, Z. M. Ding and Y. Fu, Ensemble subspace segmentation under blockwise constraints, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.28, no.7, pp.1526-1539, 2018.

[11] T. He, Z. Zhang and H. Zhang, Bag of tricks for image classification with convolutional neural networks, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp.558-567, 2019.

[12] C. F. Camerer and X. M. Li, Neural autopilot and context-sensitivity of habits, *Current Opinion in Behavioral Sciences*, vol.41, pp.185-190, 2021.

[13] W. Wei, X. Zhang and L. Yang, Full-cycle state evaluation of S700K switch machine based on residual network and fuzzy clustering, *International Journal of Innovative Computing, Information and Control*, vol.18, no.4, pp.1203-1216, 2022.

[14] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, vol.313, no.5786, pp.504-507, 2006.

[15] P. Ji, T. Zhang and H. D. Li et al., Deep subspace clustering networks, *Proc. of the 31st Conference on Neural Information Processing Systems*, pp.23-32, 2017.

[16] L. Bai, J. Liang and F. Cao, Semi-supervised clustering with constraints of different types from multiple information sources, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.43, no.9, pp.3247-3258, 2021.

[17] R. Vidal, Y. Ma and S. Sastry, *Generalized Principal Component Analysis*, Springer, 2016.

[18] G. C. Liu, Z. C. Lin, S. C. Yan et al., Robust recovery of subspace structures by low-rank representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.35, no.1, pp.171-184, 2013.

[19] J. J. Zhang, C. G. Li, C. You et al., Self-supervised convolutional subspace clustering network, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp.5473-5482, 2019.

[20] D. H. Lee, Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, *Workshop on Challenges in Representation Learning*, vol.3, no.2, 2013.

[21] J. C. Lv, K. Zhao, X. Lu et al., Pseudo-supervised deep subspace clustering, *IEEE Transactions on Image Processing*, vol.30, pp.5252-5263, 2021.
[22] F. P. Nie, J. Li and X. L. Li, Self-weighted multiview clustering with multiple graphs, *Proc. of the 26th International Joint Conference on Artificial Intelligence*, pp.2564-2570, 2017.
[23] Z. J. Wu, J. Li, J. H. Xu et al., Subspace-based self-weighted multiview fusion for instance retrieval, *Information Sciences*, vol.592, pp.261-276, 2022.
[24] R. Zhang, F. P. Nie and X. L. Li, Self-weighted spectral clustering with parameter-free constraint, *Neurocomputing*, vol.241, pp.164-170, 2017.
[25] F. P. Nie, D. Y. Wu and R. Wang, Self-weighted clustering with adaptive neighbors, *IEEE Transactions on Neural Networks and Learning Systems*, vol.31, no.9, pp.3428-3441, 2020.
[26] T. Wu, Y. C. Zhou and R. Zhang, Self-weighted discriminative feature selection via adaptive redundancy minimization, *Neurocomputing*, vol.275, pp.2824-2830, 2018.
[27] W. Chang, F. P. Nie and Z. Wang, Self-weighted learning framework for adaptive locality discriminant analysis, *Pattern Recognition*, vol.129, 108778, DOI: 10.1016/j.patcog.2022.108778, 2022.
[28] D. Y. Zhou, O. Bousquet, T. Lal et al., Learning with local and global consistency, *Advances in Neural Information Processing Systems*, vol.16, pp.321-328, 2003.
[29] G. R. Wang, G. C. Wang et al., Weakly supervised person Re-ID: Differentiable graphical learning and a new benchmark, *IEEE Transactions on Neural Networks and Learning Systems*, vol.32, no.5, pp.2142-2156, 2021.
[30] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *Proc. of the 3rd International Conference on Learning Representations*, 2015.
[31] N. D and B. Triggs, Histograms of oriented gradients for human detection, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.1, pp.886-893, 2005.
[32] C. Y. Lu, J. S. Feng, Z. C. Lin et al., Clustering ensemble selection considering quality and diversity, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.41, no.2, pp.487-501, 2019.
[33] N. X. Vinh, J. Epps and J. Bailey, Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance, *Journal of Machine Learning Research*, vol.11, pp.2837-2854, 2010.
[34] J. Y. Xie, R. Girshick and A. Farhadi, Unsupervised deep embedding for clustering analysis, *Proc. of the 33rd International Conference on Machine Learning*, vol.48, pp.478-487, 2016.

Author Biography

Zhaoqiang Bao received the B.S. degree from Yantai University, China, in 2021. He is currently pursuing the M.S. degree in computer science with Yantai University, under the supervision of Prof. Lihong Wang. His main research interests include data mining and subspace clustering.

Lihong Wang received the B.S. degree from Tsinghua University, China, in 1990, the M.S. degree from the University of Science and Technology of China (USTC), in 1993, and the Ph.D. degree from Shanghai University, China, in 2004. She is currently a Professor with the School of Computer and Control Engineering, Yantai University. Her research interests include data mining and machine learning.