LRSC: Learning Representations for Subspace Clustering

Changsheng Li[1]*, Chen Yang[2], Bo Liu[3], Ye Yuan[1], Guoren Wang[1]
[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[2]School of Computer Science and Engineering, University of Electronic Science and Technology of China
[3]JD Digits
{lcs,yuan-ye}@bit.edu.cn, chenyangah@gmail.com, {kfliubo,wanggrbit}@126.com

Abstract

Deep learning based subspace clustering methods have attracted increasing attention in recent years, where a basic theme is to non-linearly map data into a latent space, and then uncover subspace structures based upon the data self-expressiveness property. However, almost all existing deep subspace clustering methods only rely on target domain data, and always resort to shallow neural networks for modeling data, leaving huge room to design more effective representation learning mechanisms tailored for subspace clustering. In this paper, we propose a novel subspace clustering framework through learning precise sample representations. In contrast to previous approaches, the proposed method aims to leverage external data through constructing lots of relevant tasks to guide the training of the encoder, motivated by the idea of meta-learning. Considering limited networks layers of current deep subspace clustering models, we intend to distill knowledge from a deeper network trained on the external data, and transfer it into the shallower model. To reach the above two goals, we propose a new loss function to realize them in a unified framework. Moreover, we propose to construct a new auxiliary task for self-supervised training of the model, such that the representation ability of the model can be further improved. Extensive experiments are performed on four publicly available datasets, and experimental results clearly demonstrate the efficacy of our method, compared to state-of-the-art methods.

Introduction

Clustering is a popular tool for unsupervised data analytics (Hinton et al. 1999). In many real-world applications, data from each cluster can be approximately modelled by a proper low-dimensional subspace. For example, in movie recommendation systems, users having similar watching interests approximately span a low-dimensional subspace (Zhang et al. 2012). In bioinformatics domain, the scRNA-seq expression of the instances from one cell type forms a subspace structure (Zheng et al. 2019). A typical example in vision problems is that face images are known to lie in a linear subspace of dimension up to nine (Zhang et al. 2019a). To deal with such data, subspace clustering is proposed to cluster data into groups, where each group contains samples from the same subspace. The topic of subspace clustering has been extensively studied and applied in many research domains (Moise and Sander 2008; Peng et al. 2016; You, Robinson, and Vidal 2016; Liu et al. 2016; Peng et al. 2017; Liu et al. 2017; Ji et al. 2017; Zhang et al. 2018; Peng et al. 2018; Zhou et al. 2019; Zhang et al. 2019b; Li et al. 2020b).

To date, a number of subspace clustering methodologies, such as iterative methods (Zhang, Szlam, and Lerman 2009; Ho et al. 2003), statistical methods (Gruber and Weiss 2004; Rao et al. 2009) and spectral analysis based methods (Yan and Pollefeys 2006; Goh and Vidal 2007; Peng, Yi, and Tang 2015), have been proposed. The spectral analysis based methods have received more attention in recent years. The algorithm belonging to this category generally learns an affinity graph, and then applies spectral analysis to obtain the clustering result. Many of the spectral analysis based algorithms are built upon the self-representative property of data which has been successfully applied to many areas (Elhamifar and Vidal 2013; Peng, Zhang, and Yi 2013; Li et al. 2018, 2020a). For subspace clustering, the typical models include low-rank representation (LRR) (Liu, Lin, and Yu 2010), sparse subspace clustering (SSC) (Elhamifar and Vidal 2013), smooth representation clustering (Hu et al. 2014), block diagonal representation (Lu et al. 2018), stochastic sparse subspace clustering (Chen, Li, and You 2020), etc.

In many practical scenarios, the data do not necessarily comply with the assumption that they are drawn from multiple linear subspaces. Thereby this limits the applications of traditional subspace clustering methods. Several deep learning based solutions are thus proposed to alleviate this problem. A Basic idea among deep subspace clustering models is to non-linearly map original input into a latent space in which a self-representative layer is incorporated to learn the affinity graph (Ji et al. 2017). Based on this basic strategy, some representative works are successively proposed in recent years to further boost the performance of subspace clustering, including dual self-supervised CNN (Zhang et al. 2019a), latent distribution preserving (Zhou et al. 2019), deep adversarial subspace clustering (Zhou, Hou, and Feng 2018), multi-scale fusion (Dang et al. 2020).

Although existing deep subspace clustering approaches have achieved remarkable results, there is still huge room to design more effective representation learning mechanisms due to the following reasons: First, all methods mentioned

*Changsheng Li is the corresponding author.

above attempt to model subspaces only depending on the target dataset, but ignore a lot of external data available on the internet. This is a quite challenging problem if we only consider the target domain data, owing to the unsupervised nature of subspace clustering. In contrast, it should be beneficial to subspace clustering if we can leverage external useful information during learning; Second, current deep models basically take advantage of very shallow neural network architectures for uncovering subspace structures. Because of the introduction of the self-representative layer, it is usually hard for these methods to utilize very deep neural networks to model data, thus limiting their representation abilities.

In light of these, this paper proposes a novel subspace clustering framework by learning precise sample representations, to overcome the above deficiencies. To make use of external relevant data, the proposed method intends to construct lots of auxiliary relevant tasks on external data with the purpose of improving the ability of the encoder, motivated by the emerging meta-learning research (Vilalta and Drissi 2002). In view of the limited layers of current approaches, we attempt to distill valuable knowledge from a deeper neural network well trained on the external data, and propose a new loss function to transfer the knowledge into the shallower model. Finally, a new auxiliary task is constructed based upon the self-representative layer, which provides self-supervision information to further enhance the representation ability of the model.

Our major contributions can be summarized as follows:

- We propose a principled framework for subspace clustering under a new problem setting, i.e., external relevant data are available. To the best of our knowledge, this is the first work to explore how to leverage external data to facilitate model learning for subspace clustering.
- We devise a new strategy to jointly transfer useful knowledge from both task-level and model-level aspects, and propose a new loss function to simultaneously realize them in a joint framework.
- We design a new self-supervised learning task to guide learning of the model, such that a better sample representation can be obtained, and the performance of subspace clustering can be thus further improved.
- Extensive experiments are conducted on four real-world datasets to evaluate the performance of the proposed method, and the results demonstrate its effectiveness.

Related Work

In this section, we will briefly review some works related to our method, including deep subspace clustering and meta-learning.

Deep Subspace Clustering

An earlier work, named PARTY, is proposed in (Peng et al. 2016), which realizes the idea of self-expression in the neural network and incorporates a prior sparsity constraint into the latent space for sparseness reconstruction. Similarly, the work in (Ji et al. 2017) incorporates the self-expression layer into an auto-encoder, and presents a two-stage strategy to effectively learn the parameters of the network in an end-to-end framework. Motivated by this work, more advanced deep learning based approaches have been proposed in recent years. For instance, deep adversarial subspace clustering (Zhou, Hou, and Feng 2018) introduces adversarial learning to simultaneously guide the learning of sample representation and subspace clustering. The authors in (Zhang et al. 2019b) utilize the complementary property of the classifier-induced affinities and the subspace-based affinities to train a deep subspace clustering model in a collaborative scheme. Moreover, a self-supervised convolutional subspace clustering network proposed in (Zhang et al. 2019a) integrates the stacked convolution based feature extraction module, the self-expression based affinity learning model, as well as the spectral clustering based data segmentation into a joint framework. To make the distribution of latent representation be consistent with original data distribution, a distribution consistency loss is presented in (Zhou et al. 2019) by minimizing the KL divergence between the two distributions. Although the above deep subspace clustering methods generally achieve better performance than traditional methods, almost all of them only rely on the target data to learn subspace structures, leaving huge room to explore more effective models by leveraging data from relevant domains.

Meta-Learning

Despite of the concept of meta-learning (a.k.a. learning to learn) has a long standing history (Schmidhuber, Zhao, and Wiering 1997), it has resurged in popularity in recent years. Meta-learning intends to learn knowledge from a lot of previous related tasks, and quickly adapt to new tasks with a few training examples. Because of its potential, meta-learning has been widely studied for recommendation systems (Lee et al. 2019), few-shot learning (Snell, Swersky, and Zemel 2017), domain adaption (Balaji, Sankaranarayanan, and Chellappa 2018). The most related of these studies to ours is the model-agnostic meta-learning (MAML) approach (Finn, Abbeel, and Levine 2017). MAML aims to find a meta-learner that learns to effectively initialize a base-learner for a new learning task, where the meta-learner is optimized by gradient descent using the validation loss of the base-learner. Motivated by MAML, a recent line of work (Finn, Xu, and Levine 2018; Yao et al. 2019; Jamal and Qi 2019), has been proposed. Different from these methods, we attempt to leverage meta-learning for subspace clustering in this paper.

Proposed Method

In this section, we will elaborate the details of the proposed model. As shown in Figure 1, our method is mainly made up of three modules: A meta-learn module is used to gain rich experiences from a task-level point of view, i.e., constructing relevant tasks on external data to help the target model learn effectively; A distillation module aims to transfer knowledge from a model-level point of view, i.e., training a deeper network on external data to help the shallower target model learn, such that the problem of limited representation ability of deep subspace clustering can be circumvented; A self-supervised module intends to construct

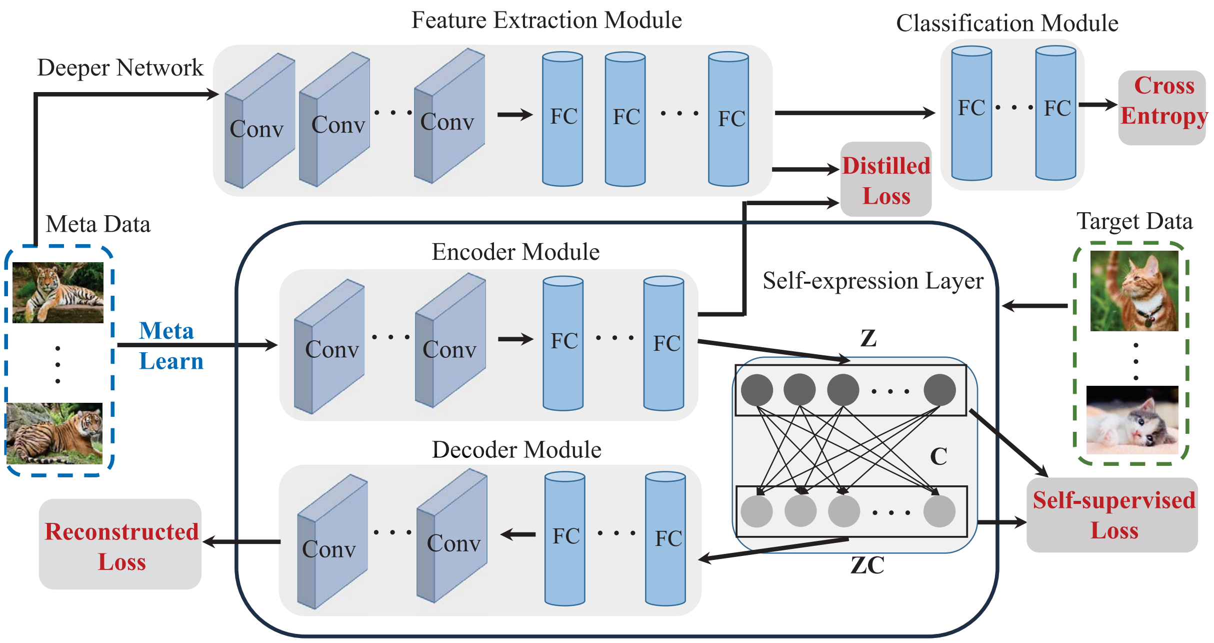

Figure 1: Illustration of the overall architecture. Our method mainly consists of three modules: A meta-learn module to acquire useful information from lots of auxiliary relevant tasks based on the external data, making the target model effectively learn on the target data; A distillation module to transfer knowledge from a deeper network to the shallower target network, so as to improve the representation ability of the model; A self-supervised module used to construct a auxiliary task based on the self-expression layer for self-supervised learning on the target data.

an auxiliary task through the self-expression layer, such that the representation ability of the target model can be further improved. Before introducing our modules in detail, we first give some preliminaries and our new problem setting.

Preliminaries and Problem Statement

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ be a data matrix from the target domain, whose data points are drawn from a union of multiple low-dimensional subspaces. d is the dimension of samples and n is the number of the target data. In the meantime, given another external data matrix $\mathcal{M} = \{\mathbf{x}_i^m, y_i^m\}_{i=1}^{\hat{n}}$, called meta data, from a source domain relevant to the target domain. $\mathbf{x}_i^m \in \mathbb{R}^d$ is the i-th sample representation in $\mathcal{M}$, and $y_i^m \in \mathbf{R}$ is its label. $\hat{n}$ is the number of meta data. Our goal is to build a principled and generic framework that can leverage the external data $\mathcal{M}$ to help the model better uncover subspace structures of the target data $\mathbf{X}$. To the best of our knowledge, this is the first work to explore how to leverage external data for subspace clustering.

In order to learn subspace structures of data, current deep methods usually map inputs into a latent space, and then learn an affinity graph through introducing a self-expression layer. A representative work is the DSC-Nets (Ji et al. 2017) which consists of an encoder module, a decoder module, as well as a self-expression layer. The encoder and decoder modules are used to learn a nonlinear latent space, and the self-expression module aims to learn pairwise affinities between all data points based on this latent space. The loss function of DSC-Nets can be formulated as:

$$\min \mathcal{L}_b = \frac{1}{2}||\mathbf{X} - \widehat{\mathbf{X}}||_F^2 + \frac{\lambda_1}{2}||\mathbf{Z} - \mathbf{ZC}||_F^2 + \lambda_2||\mathbf{C}||_l^2 \quad (1)$$
$$s.t. \ \mathrm{diag}(\mathbf{C}) = \mathbf{0},$$

where $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are two tradeoff parameters. $||\cdot||_F$ is the Frobenius norm of a matrix. $||\mathbf{C}||_l^2$ denotes a certain matrix norm on $\mathbf{C}$ to discover the subspace structure, where DSC-Nets uses the l_1-norm and Frobenius norm, respectively. $\mathbf{Z}$ and $\widehat{\mathbf{X}}$ are the latent representation and the reconstruction of $\mathbf{X}$, respectively. They can be obtained by:

$$\begin{cases} \mathbf{Z} = \Phi_e(\mathbf{X}, \theta_e) \\ \widehat{\mathbf{X}} = \Phi_d(\mathbf{ZC}, \theta_d) \end{cases} \quad (2)$$

where Φ_e and Φ_d denote the architecture of the encoder and decoder, respectively. θ_e and θ_d are the learned parameters of the encoder and decoder, respectively.

In (1), the first term aims to minimize the reconstruction error. The second term is a self-representation loss measured in a self-expression layer which is made up of one fully connected layer without bias and activation functions. The last one is a regularization term used to discover the subspace structure. The constraint ensures the diagonal elements of $\mathbf{C}$ equal to zeros, avoiding degenerated solutions.

Leveraging External Data

As aforementioned, the objective function in (1) only relies on the target data $\mathbf{X}$ for subspace clustering, while ignores external data. To learn subspace structures more effectively, we devise a new mechanism to leverage external data from both task-level and model-level aspects simultaneously.

Task-level: Motivated by MAML (Finn, Abbeel, and Levine 2017), we adopt a similar strategy in our meta-learn module to gain experience from a lot of auxiliary relevant tasks: Let $\mathcal{T}$ denote a subspace clustering task sampled from a distribution $p(\mathcal{T})$ based on meta data $\mathcal{M}$, where $\mathcal{T}$ is called episode. Each episode is divided into two non-overlapping splits: a training split $\mathcal{M}^{(tr)}$ for updating the base-learner $\theta_{\mathcal{T}}$, as well as a test split $\mathcal{M}^{(te)}$ for optimizing the meta-learner θ. The forms of the loss functions in the above two updating processes are the same with (1), and are defined as:

$$\mathcal{L}_t = \frac{1}{2}\sum_i \|\mathbf{x}_i^m - \tilde{\mathbf{x}}_i^m\|_2^2 + \frac{\lambda_3}{2}\|\tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}\tilde{\mathbf{C}}\|_F^2 + \lambda_4\|\tilde{\mathbf{C}}\|_l^2, \quad (3)$$

where $\tilde{\mathbf{x}}_i^m$ is the reconstruction of $\mathbf{x}_i^m$. $\lambda_3 \geq 0$ and $\lambda_4 \geq 0$ are two tradeoff parameters. We omit the constraint condition for writing conveniently.

Model-level: In (3), it still only allows shallow neural networks to model data, due to large memory requirements for learning the affinity graph. To circumvent this problem, we attempt to train a deeper neural network by a supervised task, and then distill knowledge from this deeper model to guide the learning of the shallower model, such that the representation ability of the latter can be improved.

Assuming Φ_T is a deeper network which is trained on the meta data $\mathcal{M}$ by a classification loss (e.g., the cross-entropy loss), as shown in Figure 1. Let $\mathbf{T}_i^m$ denote a latent feature representation of $\mathbf{x}_i^m$, obtained by the feature extraction module in the deeper network Φ_T. In order to transfer knowledge from the deeper model Φ_T into the shallower network, we propose a new distilled loss to minimize:

$$\mathcal{L}_m = \frac{1}{2}\sum_{i \neq j} \mathbf{W}_{ij}^m ||\tilde{\mathbf{Z}}_i - \tilde{\mathbf{Z}}_j||_2^2, \quad (4)$$

where $\tilde{\mathbf{Z}}_i$ is the i-th column in $\tilde{\mathbf{Z}}$, denoting the latent representation of the sample $\mathbf{x}_i^m$. $\mathbf{W}_{ij}^m$ is a weight to incur a heavy penalty to a large distance between $\tilde{\mathbf{Z}}_i$ and $\tilde{\mathbf{Z}}_j$ if they are close based on $\mathbf{T}^m$. In other words, if the distance of two feature vectors in $\mathbf{T}^m$ is small (or large), then the distance of the corresponding two feature vectors in $\tilde{\mathbf{Z}}$ is expected to also be small (or large). The weight $\mathbf{W}_{ij}^m$ is defined as:

$$\mathbf{W}_{ij}^m = e^{-\frac{||\mathbf{T}_i^m - \mathbf{T}_j^m||_2^2}{t}}, \quad (5)$$

where $\mathbf{T}_i^m$ is the i-th column in $\mathbf{T}^m$. $t \in \mathbb{R}$ is a hyperparameter.

In order to simultaneously transfer knowledge from both task-level and model-level aspects, we can plug (4) into (3), and obtain a new loss function as:

$$\begin{aligned}\min\ \mathcal{L}_e =& \frac{1}{2}\sum_i ||\mathbf{x}_i^m - \tilde{\mathbf{x}}_i^m||_2^2 + \frac{\lambda_3}{2}||\tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}\tilde{\mathbf{C}}||_F^2 \\ &+ \lambda_4||\tilde{\mathbf{C}}||_l^2 + \frac{\eta}{2}\sum_{i \neq j}\mathbf{W}_{ij}^m||\tilde{\mathbf{Z}}_i - \tilde{\mathbf{Z}}_j||_2^2, \quad (6)\end{aligned}$$

where η is a tradeoff parameter.

Algorithm 1: Leveraging External Data

```
Input: $P(\mathcal{T})$ denotes a distribution over meta tasks;
  tradeoff parameters
Initialization: randomly Initialize $\theta = \{\theta_e, \theta_d\}$ , $\tilde{\mathbf{C}}$
while not done do
    Sample batch of tasks $\mathcal{T}_i \sim P(\mathcal{T})$
    for all $\mathcal{T}_i$ do
        (S1) Extract features of $\mathcal{M}_i^{(tr)}$ using the
          feature extraction module of the deeper
          network $\Phi_T$, denoted as $\mathbf{T}_i^{(tr)}$;
        (S2) Compute $\mathbf{W}$ using $\mathbf{T}_i^{(tr)}$ by (5);
        (S3) Compute $L_{\mathcal{T}_i}(\theta; \mathcal{M}_i^{(tr)})$ by (6);
        (S4) Update the base-learner $\theta_i$ by
          minimizing $L_{\mathcal{T}_i}(\theta; \mathcal{M}_i^{(tr)})$;
    end
    for all $\mathcal{T}_i$ do
        (S1) Extract features of $\mathcal{M}_i^{(te)}$ using the
          feature extraction module of the deeper
          network $\Phi_T$, denoted as $\mathbf{T}_i^{(te)}$;
        (S2) Compute $\mathbf{W}$ using $\mathbf{T}_i^{(te)}$ by (5);
        (S3) Compute $L_{\mathcal{T}_i}(\theta_i; \mathcal{M}_i^{(te)})$ by (6);
    end
    Update the meta-learner $\theta$ by minimizing
      $\sum_{\mathcal{T}_i \sim P(\mathcal{T})} L_{\mathcal{T}_i}(\theta_i; \mathcal{M}_i^{(te)})$;
end
Output: Meta-learner $\theta$.
```

By minimizing the above loss function, the knowledge from the external data can be simultaneously distilled from two views: task-level and mode-level. The key steps of leveraging external data are summarized in Algorithm 1.

Subspace Clustering on Target Data

When training the meta-learner θ well based on the external data $\mathcal{M}$, we can quickly adapt it to a new subspace clustering task. To this end, we utilize the meta-learner θ to provide a good initialization to the target model. In the meantime, we also utilize the deeper network to further guide the representation learning of the target model. The loss function on the target data is defined as:

$$\begin{aligned}\min\ \mathcal{L}_n =& \frac{1}{2}||\mathbf{X} - \tilde{\mathbf{X}}||_2^2 + \frac{\lambda_1}{2}||\mathbf{Z} - \mathbf{Z}\mathbf{C}||_F^2 \\ &+ \lambda_2||\mathbf{C}||_l^2 + \frac{\mu}{2}\sum_{i \neq j}\mathbf{W}_{ij}||\mathbf{Z}_i - \mathbf{Z}_j||_2^2, \quad (7)\end{aligned}$$

where $\mu \geq 0$ is a tradeoff parameter. $\mathbf{W}_{ij}$ is defined as in (5), except that the features extracted by the deeper neural network are from the target data $\mathbf{X}$.

In (7), by taking advantage of the meta-learner θ to initialize the target model, and utilizing the deeper network to assist in the learning of the target model, the knowledge from the external data can be better transferred into the target task. Moreover, in order to further improve the representation ability, we construct a new auxiliary task to learn a better representation in a self-supervised manner, motivated by the popular self-supervised learning (Wu et al. 2018).

Given an input of the self-expression layer, $\mathbf{Z}_i$, and a set of the outputs of the self-expression layer, $\{\mathbf{G}_j = \mathbf{Z}\mathbf{C}_j\}_{j=1}^n$. Here we construct the following self-supervised task: we take $\mathbf{Z}_i$ and $\mathbf{G}_i$ as a positive pair, i.e., $\mathbf{Z}_i$ and $\mathbf{G}_i$ are deemed to be matched. In the meantime, we take $\mathbf{Z}_i$ and all other points as $\{\mathbf{G}_j\}_{j=1,j\neq i}^n$ negative pairs. Then, we consider to optimize the following self-supervised loss function as:

$$\min \mathcal{L}_c = -\log \frac{\exp(\overrightarrow{\mathbf{Z}}_i^T \overrightarrow{\mathbf{G}}_i/\sigma)}{\sum_{j=1}^n \exp(\overrightarrow{\mathbf{Z}}_i^T \overrightarrow{\mathbf{G}}_j/\sigma)} \quad (8)$$

where σ is a temperature parameter that controls the concentration level of the distribution (Wu et al. 2018). $\overrightarrow{\mathbf{Z}}_i$ and $\overrightarrow{\mathbf{G}}_i$ are the normalization of $\mathbf{Z}_i$ and $\mathbf{G}_i$ respectively, where $||\overrightarrow{\mathbf{Z}}_i|| = 1$ and $||\overrightarrow{\mathbf{G}}_i|| = 1$ are realized by a L2-normalization layer. The sum in the denominator of (8) is over one positive and $n-1$ negative samples. Intuitively, this loss is the log loss of a n-way softmax-based classifier that classify $\mathbf{Z}_i$ as $\mathbf{G}_i$.

Based on (7) and (8), the final loss function on the target data is defined as:

$$\min \mathcal{L}_f = \frac{1}{2}\|\mathbf{X} - \tilde{\mathbf{X}}\|_2^2 + \frac{\lambda_1}{2}\|\mathbf{Z} - \mathbf{Z}\mathbf{C}\|_F^2 + \lambda_2\|\mathbf{C}\|_l^2 + \frac{\mu}{2}\sum_{i\neq j}\mathbf{W}_{ij}\|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2 - \gamma \log \frac{\exp(\overrightarrow{\mathbf{Z}}_i^T \overrightarrow{\mathbf{G}}_i/\sigma)}{\sum\limits_{j=1}^n \exp(\overrightarrow{\mathbf{Z}}_i^T \overrightarrow{\mathbf{G}}_j/\sigma)}, \quad (9)$$

The key steps to perform subspace clustering on the target data are listed in Algorithm 2. Through training, the self-expression matrix $\mathbf{C}$ can be obtained, and then the affinity matrix can be computed by $\frac{1}{2}(\|\mathbf{C}\| + \|\mathbf{C}^T\|)$, as in most self-representation methods (Ji et al. 2017). Finally, the spectral clustering algorithm (Ng, Jordan, and Weiss 2002) is performed on the affinity matrix for subspace clustering.

Training

To train our method, we propose a two-stage training strategy: First we pre-train the meta-learner based on Algorithm 1, to provide a good initialization to the target model; Second, we fine-tune the target model by Algorithm 2.

Remark. *To extract features using the deeper neural network on the meta data and target data, we divide $\mathcal{M}$ into two non-overlapping subsets: $\mathcal{M}_1$ and $\mathcal{M}_2$. We first train the deeper network using $\mathcal{M}_1$, and then extract features of $\mathcal{M}_2$ by the trained model. After that, we re-train the model by $\mathcal{M}_2$, and use it to extract features for $\mathcal{M}_1$. Finally, we merge the two feature sets as a knowledge base for transferring knowledge. Meanwhile, the deeper network trained on either $\mathcal{M}_1$ or $\mathcal{M}_2$ can be used as a feature extractor to extract features of target data $\mathbf{X}$ for guiding the learning of the target model.*

EXPERIMENTS

In this section, we study the effectiveness of the proposed approach on four publicly available datasets including two text datasets and two image datasets.

Algorithm 2: Subspace Clustering on Target Tasks

```
Input: $\mathbf{X}$ denotes the target data matrix; tradeoff
  parameters
Initialization: The parameters of the target model is
  initialized by those of meta-learner.
Extract features of $\mathbf{X}$ using the deeper network $\Phi_T$,
  denotes as $\mathbf{T}$;
// Pre-train auto-encoder without self-expression
  layer
for $k = 1, 2, \cdots, K_1$ do
    (S1) Sample a mini batch $D_k$ from $\mathbf{X}$;
    (S2) Select a subset $\mathbf{T}_k$ from $\mathbf{T}$ based on $D_k$;
    (S3) Compute $\mathcal{L}_f$ by Eq. (9) with $\lambda_2 = \gamma = 0$;
    (S4) Update auto-encoder via the BP algorithm.
end
// Fine-tune auto-encoder and self-expression layer
for $k = 1, 2, \cdots, K_2$ do
    (S1) Compute the loss $\mathcal{L}_f$ by Eq. (9);
    (S2) Update the whole network via the BP
      algorithm.
end
Output: $\mathbf{C}$.
```

Dataset

We evaluate the performance of our proposed approaches on four publicly available datasets:

- **Fashion-MNIST dataset**: It contains 10 categories of fashion clothing. Similar attributes in different classes and diversity in the same class make this dataset more difficult for subspace clustering. We randomly select 1,000 images from each class in our experiment.
- **notMNIST**: It consists of 10 classes with letters A-J taken from different fonts. Similar to Fashion-MNIST, We randomly selected 1,000 images from each class. Thus, there are 10,000 images in our experiment.
- **BBC dataset**: This dataset consists of 2,225 documents from BBC news corresponding to stories in five topical areas. We extract the TF-IDF features to represent samples, and utilize TruncatedSVD (Hansen 1987) to reduce the dimensions to 256.
- **20 Newsgroups dataset**: This dataset is a collection of 18,846 messages from 20 different news groups. We extract the TF-IDF features and use TruncatedSVD to reduce the dimensions to 512.

We use the miniImageNet (Vinyals et al. 2016) dataset as the meta data for two image datasets, and use the 20-newsgroup dataset as the meta data for the BBC dataset. For the 20 Newsgroups dataset, we randomly select 10 categories as the target data, and use the rest for meta data.

Experimental Setting

Compared Methods: To show the competitiveness of the proposed method, we compare with the following methods:

- Clustering methods: ClusterGAN (Mukherjee et al. 2019) based on the generative adversarial network (GAN).

Dataset	Metric	SSC	EnSC	SSC-OMP	LRR	ClusterGAN	DSCN -L1	DSCN -L2	DASC	DPSC	LRSC -L1	LRSC -L2
Fashion-MNIST	ACC	0.528	0.519	0.423	0.519	0.581	0.608	0.585	0.617	0.624	**0.652**	0.651
	NMI	0.524	0.559	0.508	0.625	0.597	0.640	0.626	0.647	0.645	**0.672**	**0.672**
	ARI	0.386	0.392	0.302	0.399	0.446	0.470	0.447	0.483	0.484	**0.529**	0.527
not-MNIST	ACC	0.240	0.267	0.277	0.282	0.510	0.517	0.517	0.514	0.566	0.587	**0.608**
	NMI	0.224	0.226	0.223	0.335	0.345	0.501	0.501	0.506	0.508	0.517	**0.523**
	ARI	0.023	0.026	0.076	0.074	0.246	0.413	0.412	0.408	0.415	0.439	**0.467**
BBC	ACC	0.410	0.479	0.307	0.524	0.510	0.855	0.879	0.893	0.864	0.928	**0.929**
	NMI	0.201	0.387	0.108	0.451	0.201	0.722	0.727	0.751	0.726	0.802	**0.804**
	ARI	0.144	0.314	0.064	0.209	0.172	0.694	0.725	0.759	0.724	0.834	**0.836**
20-News	ACC	0.235	0.229	0.210	0.186	0.244	0.348	0.352	0.356	0.336	0.369	**0.406**
	NMI	0.138	0.127	0.093	0.139	0.116	0.275	0.283	0.280	0.250	0.287	**0.315**
	ARI	0.064	0.067	0.064	0.024	0.064	0.112	0.122	0.122	0.105	0.116	**0.161**

Table 1: Clustering results on the four datasets in terms of ACC, NMI, and ARI.

- Traditional subspace clustering methods: SSC (Elhamifar and Vidal 2009), LRR (Liu, Lin, and Yu 2010), ENSC (You et al. 2016), SSC-OMP (You, Robinson, and Vidal 2016).
- Deep subspace clustering methods: DSCN-L1 (Ji et al. 2017), DSCN-L2 (Ji et al. 2017), DASC(Zhou, Hou, and Feng 2018), DPSC(Zhou et al. 2019).
- Our method: LRSC-L1: Our method using the l_1 norm to regularize the self-expression matrix. LRSC-L2: Our method using the l_2 norm as a regularization.

Evaluation Metrics: For all quantitative evaluations, we use three widely used evaluation metrics for subspace clustering: accuracy (ACC), the normalized mutual information (NMI), and the adjusted rand index (ARI).

Implementation Details: For two image datasets, we employ a CNN with 4 convolutional layers as the encoder, and a symmetric structure for the decoder. For all convolutional layers, we set the kernel size to 3×3 with stride 2 and the number of channels to 10-20-30-40 for the four convolutional layers in the encoder, respectively. Moreover, we use a deep convolutional neural network with 17 convolutional layers plus one fully-connected layer as the larger network. All images in the two datasets are resized to 32×32 for training. For two text datasets, we take advantage of a shallow neural network with one fully-connected layer as the encoder, and one fully-connected layer for the decoder. In addition, we utilize a deep neural networks with four fully-connected layers as the deper network. In all layers, the rectified linear unit (ReLU) (Krizhevsky, Sutskever, and Hinton 2012) is used as the activations. For hyper-parameter setting, we report the detailed setting in the supplementary material.

Experimental Result

General Performance: We show the performance of our proposed approaches and baselines on the four datasets in Table 1. Our methods consistently outperform other models on the four datasets. Moreover, deep subspace clustering methods consistently outperform the deep clustering method ClusterGAN, which indicates that data are sampled from a union of multiple subspaces in many real-world applications, and it is necessary to develop advanced subspace clustering algorithms to better exploit subspace structures for clustering problems. Traditional linear subspace clustering algorithms obtain poor performances on the four datasets. This is because the data do not necessarily conform to linear subspace models in many practices. In addition, as DSCN-L1 and DSCN-L2 are our two base models, LRSC-L1 achieves better performance than DSCN-L1, and LRSC-L2 outperforms DSCN-L2 on the four datasets. This demonstrates that learning more precise representations can improve the performance of the model.

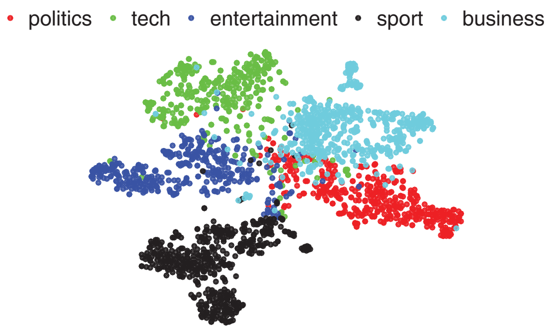

Figure 2: The visualization of the latent space of our LRSC-L2 through dimensionality reduction by t-SNE (Der Maaten and Hinton 2008) on the BBC dataset.

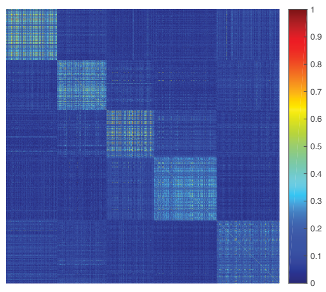

Figure 3: Affinity matrix derived by LRSC-L2 on the BBC dataset. The subspace structure of the data can be clearly discovered.

Effect of Different Sizes of Data: We report the results

(a) Fashion-MNIST

No. Points	1000			5000			10000		
Metric	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SSC	0.474	0.484	0.327	0.522	0.503	0.371	0.528	0.524	0.386
EnSC	0.566	0.510	0.366	0.526	0.540	0.387	0.519	0.559	0.392
SSC-OMP	0.343	0.301	0.148	0.434	0.358	0.223	0.423	0.508	0.302
LRR	0.561	0.619	0.417	0.551	0.625	0.415	0.519	0.625	0.399
ClusterGAN	0.558	0.504	0.354	0.582	0.602	0.448	0.581	0.597	0.446
DSCN-L1	0.536	0.594	0.406	0.563	0.558	0.372	0.608	0.640	0.470
DSCN-L2	0.536	0.594	0.406	0.563	0.558	0.372	0.585	0.626	0.447
DASC	0.548	0.573	0.400	0.578	0.598	0.431	0.617	0.647	0.483
DPSC	0.560	0.606	0.426	0.599	0.584	0.425	0.624	0.645	0.484
LRSC-L1	0.588	**0.645**	**0.458**	0.631	0.674	0.508	**0.652**	**0.672**	**0.529**
LRSC-L2	**0.607**	0.627	0.439	**0.636**	**0.682**	**0.516**	0.651	**0.672**	0.527

(b) notMNIST

No. Points	1000			5000			10000		
Metric	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SSC	0.429	0.332	0.184	0.412	0.350	0.180	0.240	0.224	0.023
EnSC	0.481	0.372	0.246	0.431	0.369	0.201	0.267	0.226	0.026
SSC-OMP	0.424	0.351	0.226	0.385	0.368	0.204	0.277	0.223	0.075
LRR	0.454	0.451	0.158	0.306	0.315	0.097	0.282	0.335	0.074
ClusterGAN	0.475	0.386	0.230	0.461	0.343	0.225	0.510	0.345	0.246
DSCN-L1	0.488	0.440	0.318	0.521	0.488	0.407	0.517	0.501	0.413
DSCN-L2	0.489	0.439	0.316	0.520	0.487	0.406	0.517	0.501	0.412
DASC	0.483	0.441	0.312	0.527	0.500	0.421	0.514	0.506	0.408
DPSC	0.471	0.430	0.303	0.523	0.477	0.403	0.566	0.508	0.415
LRSC-L1	0.574	0.509	0.427	0.612	0.542	0.465	0.587	0.517	0.439
LRSC-L2	**0.578**	**0.512**	**0.430**	**0.615**	**0.544**	**0.468**	**0.608**	**0.523**	**0.467**

Table 2: Clustering results on the Fashion-MNIST and notMNIST datasets with different numbers of data points.

(a) notMNIST

Method	Task -level	Model -level	Self-supervision	ACC	NMI	ARI
LRSC-L2				0.520	0.487	0.406
	✓			0.552	0.510	0.393
	✓	✓		0.597	0.522	0.440
	✓	✓	✓	**0.615**	**0.544**	**0.468**

(b) BBC

Method	Task -level	Model -level	Self-supervision	ACC	NMI	ARI
LRSC-L2				0.879	0.727	0.725
	✓			0.918	0.784	0.812
	✓	✓		0.927	0.792	0.830
	✓	✓	✓	**0.929**	**0.804**	**0.836**

Table 3: Ablation study on LRSC on the notMNISt and BBC datasets.

of all methods with different sizes of training data, in order to study the effect of varying the dataset sizes. Because of space limitations, we report the results on the Fashion-MNIST and notMNIST datasets. We randomly select 1,000, 5,000, and 10,000 samples from each dataset as the training data, respectively. The results are listed in Table 2(a) and 2(b). We can see that our methods consistently outperform other baselines with different sizes of data.

Ablation Study: To gain further understanding of the proposed method, we evaluate the effect of our components via an ablation study on the notMNISt and BBC datasets. We take LRSC-L2 as an example, owing to its superior performance. The experimental setting is as follows: We set $\eta = \mu = \gamma = 0$ in (6), (7) and (9), which means we only transfer knowledge from the task-level aspect. When setting $\gamma = 0$, the self-supervised learning module is not considered. In addition, DSC-Nets is our baseline. The experimental results are reported in Table 3, which demonstrates the effectiveness of each component in our algorithm.

Visualization To intuitively show that data are distributed in a union of multiple low-dimensional subspaces, and demonstrate our method can learn an effective representation for subspace clustering. We perform our LRSC-L2 on the BBC dataset. The results are shown in Figure 2 and 3. From Figure 2, we can clearly observe that LRSC-L2 can learn a good sample representation. Moreover, LRSC-L2 can discover the subspace structures of the data from Figure 3.

Conclusion

In this paper, we proposed a novel representation learning based subspace clustering method by leveraging external data. To this end, we devised a strategy to transfer knowledge from both task-level and model-level aspects. Moreover, we constructed a new auxiliary self-supervised task to further improve the representation ability of the encoder. Experimental results on four publicly available datasets demonstrated the effectiveness of our method.

Acknowledgments

This work was supported by the NSFC under Grants 61806044, U2001211, 61932004, and 61732003.

References

Balaji, Y.; Sankaranarayanan, S.; and Chellappa, R. 2018. Metareg: Towards domain generalization using meta-regularization. In *NIPS*, 998–1008.

Chen, Y.; Li, C.-G.; and You, C. 2020. Stochastic Sparse Subspace Clustering. In *CVPR*, 4155–4164.

Dang, Z.; Deng, C.; Yang, X.; and Huang, H. 2020. Multi-Scale Fusion Subspace Clustering Using Similarity Constraint. In *CVPR*, 6658–6667.

Der Maaten, L. V.; and Hinton, G. E. 2008. Visualizing Data using t-SNE. *JMLR* 9: 2579–2605.

Elhamifar, E.; and Vidal, R. 2009. Sparse subspace clustering. In *CVPR*, 2790–2797.

Elhamifar, E.; and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *TPAMI* 35(11): 2765–2781.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 1126–1135.

Finn, C.; Xu, K.; and Levine, S. 2018. Probabilistic model-agnostic meta-learning. In *NIPS*, 9516–9527.

Goh, A.; and Vidal, R. 2007. Segmenting motions of different types by unsupervised manifold clustering. In *CVPR*, 1–6. IEEE.

Gruber, A.; and Weiss, Y. 2004. Multibody factorization with uncertainty and missing data using the EM algorithm. In *CVPR*.

Hansen, P. C. 1987. The Truncated SVD as a Method for Regularization. *BIT* 27(4): 534–553.

Hinton, G. E.; Sejnowski, T. J.; Poggio, T. A.; et al. 1999. *Unsupervised learning: foundations of neural computation*. MIT press.

Ho, J.; Yang, M.-H.; Lim, J.; Lee, K.-C.; and Kriegman, D. 2003. Clustering appearances of objects under varying illumination conditions. In *CVPR*.

Hu, H.; Lin, Z.; Feng, J.; and Zhou, J. 2014. Smooth representation clustering. In *CVPR*, 3834–3841.

Jamal, M. A.; and Qi, G.-J. 2019. Task agnostic meta-learning for few-shot learning. In *CVPR*, 11719–11727.

Ji, P.; Zhang, T.; Li, H.; Salzmann, M.; and Reid, I. 2017. Deep subspace clustering networks. In *NIPS*, 24–33.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Lee, H.; Im, J.; Jang, S.; Cho, H.; and Chung, S. 2019. MeLU: Meta-Learned User Preference Estimator for Cold-Start Recommendation. In *KDD*, 1073–1082.

Li, C.; Ma, H.; Kang, Z.; Yuan, Y.; Zhang, X.-Y.; and Wang, G. 2020a. On Deep Unsupervised Active Learning. In *IJCAI*, 2626–2632.

Li, C.; Wang, X.; Dong, W.; Yan, J.; Liu, Q.; and Zha, H. 2018. Joint active learning with feature selection via cur matrix decomposition. *IEEE transactions on pattern analysis and machine intelligence* 41(6): 1382–1396.

Li, C.; Yang, C.; Liang, L.; Yuan, Y.; and Wang, G. 2020b. On Robust Grouping Active Learning. *IEEE Transactions on Emerging Topics in Computational Intelligence* .

Liu, B.; Yuan, X.-T.; Yu, Y.; Liu, Q.; and Metaxas, D. N. 2016. Decentralized robust subspace clustering. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 3539–3545.

Liu, B.; Yuan, X.-T.; Yu, Y.; Liu, Q.; and Metaxas, D. N. 2017. Parallel sparse subspace clustering via joint sample and parameter blockwise partition. *ACM Transactions on Embedded Computing Systems (TECS)* 16(3): 1–17.

Liu, G.; Lin, Z.; and Yu, Y. 2010. Robust subspace segmentation by low-rank representation. In *ICML*, 663–670.

Lu, C.; Feng, J.; Lin, Z.; Mei, T.; and Yan, S. 2018. Subspace clustering by block diagonal representation. *TPAMI* 41(2): 487–501.

Moise, G.; and Sander, J. 2008. Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In *KDD*, 533–541.

Mukherjee, S.; Asnani, H.; Lin, E.; and Kannan, S. 2019. Clustergan: Latent space clustering in generative adversarial networks. In *AAAI*, volume 33, 4610–4617.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *NIPS*, 849–856.

Peng, X.; Feng, J.; Lu, J.; Yau, W.-Y.; and Yi, Z. 2017. Cascade subspace clustering. In *Thirty-First AAAI conference on artificial intelligence*.

Peng, X.; Feng, J.; Xiao, S.; Yau, W.-Y.; Zhou, J. T.; and Yang, S. 2018. Structured autoencoders for subspace clustering. *IEEE Transactions on Image Processing* 27(10): 5076–5086.

Peng, X.; Xiao, S.; Feng, J.; Yau, W.-Y.; and Yi, Z. 2016. Deep Subspace Clustering with Sparsity Prior. In *IJCAI*, 1925–1931.

Peng, X.; Yi, Z.; and Tang, H. 2015. Robust subspace clustering via thresholding ridge regression. In *AAAI*, volume 25, 3827–3833.

Peng, X.; Zhang, L.; and Yi, Z. 2013. Scalable sparse subspace clustering. In *CVPR*, 430–437.

Rao, S.; Tron, R.; Vidal, R.; and Ma, Y. 2009. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *TPAMI* 32(10): 1832–1845.

Schmidhuber, J.; Zhao, J.; and Wiering, M. 1997. Shifting inductive bias with success-story algorithm, adaptive Levin search, and incremental self-improvement. *Machine Learning* 28(1): 105–130.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *NIPS*, 4077–4087.

Vilalta, R.; and Drissi, Y. 2002. A perspective view and survey of meta-learning. *Artificial intelligence review* 18(2): 77–95.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *NIPS*, 3630–3638.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 3733–3742.

Yan, J.; and Pollefeys, M. 2006. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 94–106. Springer.

Yao, H.; Wei, Y.; Huang, J.; and Li, Z. 2019. Hierarchically Structured Meta-learning. In *ICML*, 7045–7054.

You, C.; Li, C.-G.; Robinson, D. P.; and Vidal, R. 2016. Oracle based active set algorithm for scalable elastic net subspace clustering. In *CVPR*, 3928–3937.

You, C.; Robinson, D.; and Vidal, R. 2016. Scalable sparse subspace clustering by orthogonal matching pursuit. In *CVPR*, 3918–3927.

Zhang, A.; Fawaz, N.; Ioannidis, S.; and Montanari, A. 2012. Guess who rated this movie: identifying users through subspace clustering. In *UAI*, 944–953.

Zhang, C.; Fu, H.; Hu, Q.; Cao, X.; Xie, Y.; Tao, D.; and Xu, D. 2018. Generalized latent multi-view subspace clustering. *TPAMI* 42(1): 86–99.

Zhang, J.; Li, C.-G.; You, C.; Qi, X.; Zhang, H.; Guo, J.; and Lin, Z. 2019a. Self-Supervised Convolutional Subspace Clustering Network. In *CVPR*, 5473–5482.

Zhang, T.; Ji, P.; Harandi, M.; Huang, W.; and Li, H. 2019b. Neural Collaborative Subspace Clustering. In *ICML*, 7384–7393.

Zhang, T.; Szlam, A.; and Lerman, G. 2009. Median k-flats for hybrid linear modeling with many outliers. In *ICCV Workshops*, 234–241. IEEE.

Zheng, R.; Li, M.; Liang, Z.; Wu, F.-X.; Pan, Y.; and Wang, J. 2019. SinNLRR: a robust subspace clustering method for cell type detection by non-negative and low-rank representation. *Bioinformatics* 35(19): 3642–3650.

Zhou, L.; Xiao, B.; Liu, X.; Zhou, J.; Hancock, E. R.; et al. 2019. Latent distribution preserving deep subspace clustering. In *IJCAI*.

Zhou, P.; Hou, Y.; and Feng, J. 2018. Deep adversarial subspace clustering. In *CVPR*, 1596–1604.