

Linearity-Aware Subspace Clustering

Yesong Xu¹, Shuo Chen^{1*}, Jun Li¹, Jianjun Qian^{1*}

¹PCA Lab[†], Nanjing University of Science and Technology
yesong_xu@163.com, {shuochen, junli, csjqian}@njust.edu.cn

Abstract

Obtaining a good similarity matrix is extremely important in subspace clustering. Current state-of-the-art methods learn the similarity matrix through self-expressive strategy. However, these methods directly adopt original samples as a set of basis to represent itself linearly. It is difficult to accurately describe the linear relation between samples in the real-world applications, and thus is hard to find an ideal similarity matrix. To better represent the linear relation of samples, we present a subspace clustering model, Linearity-Aware Subspace Clustering (LASC), which can consciously learn the similarity matrix by employing a linearity-aware metric. This is a new subspace clustering method that combines metric learning and subspace clustering into a joint learning framework. In our model, we first utilize the self-expressive strategy to obtain an initial subspace structure and discover a low-dimensional representation of the original data. Subsequently, we use the proposed metric to learn an intrinsic similarity matrix with linearity-aware on the obtained subspace. Based on such a learned similarity matrix, the inter-cluster distance becomes larger than the intra-cluster distances, and thus successfully obtaining a good subspace cluster result. In addition, to enrich the similarity matrix with more consistent knowledge, we adopt a collaborative learning strategy for self-expressive subspace learning and linearity-aware subspace learning. Moreover, we provide detailed mathematical analysis to show that the metric can properly characterize the linear correlation between samples.

Introduction

Subspace clustering has emerged as a powerful technique for a variety of computer vision applications, including image processing (Li et al. 2020a; Zhou et al. 2020; Lu 2021), motion segmentation (Liu et al. 2013), face clustering (Zhang et al. 2020b,a; Peng et al. 2021), and gene expression analysis (McWilliams and Montana 2014), *etc.* Generally, subspace clustering models are based on the assumption that the

whole samples are drawn from a union of multi-subspace, and its purpose is to find these subspaces and arrange each sample into its corresponding subspace.

In the past several years, several subspace clustering algorithms have been developed (Nie et al. 2020; Kang et al. 2020b; Li et al. 2021). Among them, the spectral-based method has achieved promising performance in the real-world (Zhang et al. 2021; Lv et al. 2021). Especially, it is well known that the Sparse Subspace Clustering (SSC) (Elhamifar and Vidal 2013; Li, Kong, and Fu 2017) and Low-Rank Representation (LRR) (Liu et al. 2013; Li et al. 2016) are the state-of-the-art approaches and have the theoretical guarantee, which belongs to the spectral-based method. In addition, the spectral-based method is consists of two steps. Firstly, learning a similarity matrix to represent the similarity between samples by utilizing the original data. Then, the spectral clustering algorithm is employed to segment the learned similarity matrix and obtain the final clustering result. Moreover, it is worth noting that the success of subspace clustering generally relies on the learned similarity matrix.

For the spectral-based method, SSC exploits the ℓ_1 -norm to find a sparse representation from the original samples. LRR adopts the nuclear norm to regularize the similarity matrix for capturing the correlation structure of the original data. Least Squares Regression (LSR) employs the Frobenius norm to learn the similarity matrix. Its purpose is to encourage the grouping effect, which can group highly correlated samples together (Lu et al. 2012). Block Diagonal Representation (BDR) utilizes the k -block diagonal regularizer to directly pursue a similarity matrix with a block-diagonal structure (Lu et al. 2018).

In summary, all the above mentioned methods utilize the self-expressive property of linear subspaces to generate the similarity matrix, that is, let the original n samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ as a set of basis, the similarity matrix $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n]$ can be learned by exploiting the basis $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ to represent itself linearly. Mathematically, the self-expressive is expressed as $\mathbf{x}_i \approx \mathbf{X}\mathbf{c}_i$ for i -th sample (You et al. 2020; Kang et al. 2020a; Peng et al. 2021). Moreover, different methods use various regularizations on \mathbf{C} , their purpose is to make each sample to be represented only by samples in its own subspace. In many practical application scenarios, considering that real-world sam-

*Corresponding author.

[†]PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

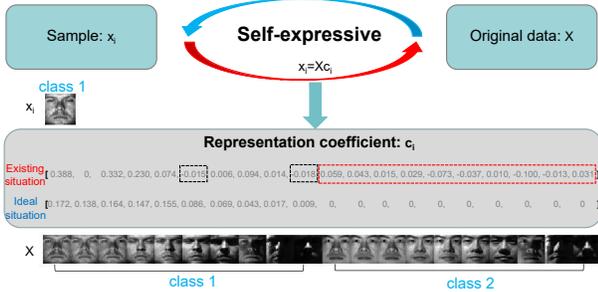


Figure 1: Example on a subset with 2 classes and 20 samples of the Extended Yale B dataset. For a given sample x_i and an original data X , x_i can be approximately expressed as a linear representation of all original data X by self-expressive-based subspace clustering methods. The representation coefficient c_i is obtained by SSC in the existing situation. We can observe that the obtained coefficient is difficult to achieve the ideal situation in reality (see red dashed box). Moreover, the obtained representation does not accurately describe the similarity between samples (see black dashed box).

ples may contain noise corruption, it is difficult to satisfy such a purpose in practice (Peng et al. 2018; Xu et al. 2020). In addition, self-expressive-based subspace clustering methods have largely ignored the precise description of the similarity between samples, which will directly affect the quality of the learned similarity matrix (see Fig. 1).

Towards addressing the above-mentioned problem, metric-based subspace clustering methods have been proposed to obtain a “good” similarity matrix (Yang et al. 2020; Wang et al. 2019; Liang et al. 2019). However, the self-expressive-based strategy explores the degree of linear correlation between samples to acquire the similarity matrix. The existing metric-based subspace clustering methods have not established an essential connection between self-expressive and metric for subspace clustering, which cannot well uncover the subspace structure of data (Zhang et al. 2018).

To avoid the loss of the connection between self-expressive and metric, and to prevent the establishment of the similarity matrix directly on the original data, in this paper, we propose a new subspace clustering method (*i.e.*, LASC) composed of a novel distance definition and a similarity matrix learning. In particular, the proposed metric can effectively measure the degree of linear correlation between samples, and we also provide theoretical guarantees of the property. To sufficiently discover subspace structure for complex real-world data, the self-expressive is adopted to discover an initial subspace (*i.e.*, representation coefficient). Linear correlation distance metric is then integrated into the framework to guide and enhance the learning of similarity matrix with linearity-aware on the learned representation coefficient. Especially, to enhance the learned similarity matrix, self-expressive subspace learning and linearity-aware subspace learning are combined for collaborative learning. The overall flowchart of our model is shown in Fig. 2. The main contributions of this paper are summarized as follows.

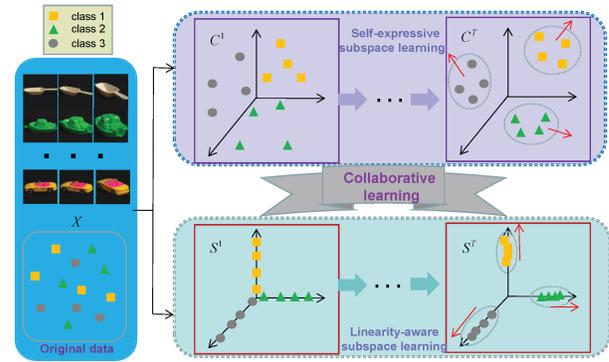


Figure 2: Overview of our LASC approach. Our model first adopts the original data to learn an initial subspace, which can find subspace structure for complex real-world data. Second, we utilize the learned initial subspace to guide the linearity-aware subspace learning. Third, we employ the learned linearity-aware subspace to guide the self-expressive subspace learning, and it enhances the structure for learned subspace. Finally, self-expressive subspace learning and linearity-aware subspace learning are used for collaborative learning, and it captures an ideal similarity matrix with more consistent information.

- A novel linearity-aware metric is proposed to measure the degree of linear correlation between samples, and we provide the theoretical analysis.
- We utilize a self-expressive strategy to learn an initial subspace structure, and further adopt the proposed linearity-aware metric to guide and enhance the learned subspace structure.
- We utilize a collaborative learning strategy to pursue the similarity matrix, which can enrich the learned similarity matrix with consistent information, and is conducive to subsequent clustering tasks.

Linearity-Aware Subspace Clustering

We start this section by defining a linearity-aware metric, and then giving the details of our objective function. Moreover, we present an optimization algorithm for the objective function, and the complexity analysis is described.

Linearity-Aware Metric

As we all know, metrics have been widely used to accurately obtain the similarity between samples for various data distribution, which can effectively adapt to complex real-world data (Chen et al. 2018, 2021a). To accurately describe the linear relationship between samples, we attempt to employ metrics to guide the learning of the similarity matrix. Motivated by the fact that the Pearson correlation coefficient can predict the linear correlation between variables (Benesty et al. 2009). To effectively gain the degree of linear correlation between samples, we first introduce the linearity-aware metric by the following definition and then analyze its advantages.

Definition 1. (Linearity-Aware Metric) For a given data matrix $X \in \mathbb{R}^{m \times n}$, where m is the dimension of the data,

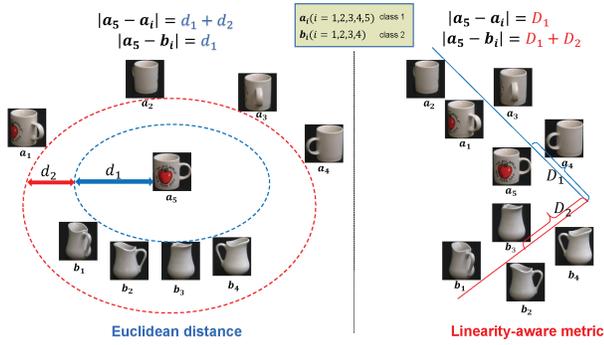


Figure 3: The comparison of Euclidean distance and our proposed linearity-aware metric. For two classes of samples (*i.e.*, a_i and b_i), the distance between a_5 and the same class samples (*i.e.*, a_1, a_2, a_3 and a_4) is $d_1 + d_2$ for Euclidean distance. However, the distance between a_5 and the different class samples (*i.e.*, b_1, b_2, b_3 and b_4) is d_1 for Euclidean distance. Thus, Euclidean distance is fail to measure the distance between samples. In contrast, linearity-aware metric is defined in Eq. (1), which is an effective kernel function. It can effectively measure the linearity-aware distance between samples.

and n is the number of the data points. For two arbitrary sample $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$, and $\mathbf{x}_i - \bar{\mathbf{x}}_i, \mathbf{x}_j - \bar{\mathbf{x}}_j \neq \mathbf{0}$, where $\bar{\mathbf{x}}_i = \frac{1}{m} \sum_{k=1}^m x_i^k$ is the average value of i -th sample. The linearity-aware metric between them is defined as:

$$\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{(\mathbf{x}_i - \bar{\mathbf{x}}_i)^\top (\mathbf{x}_j - \bar{\mathbf{x}}_j)}{\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|_2 \|\mathbf{x}_j - \bar{\mathbf{x}}_j\|_2}. \quad (1)$$

In the following, we are ready to theoretically analyze the properties of the defined metric in Eq. (1), which can characterize the degree of linear correlation between two samples. We first give the following Theorem 1 regarding the boundedness of Eq. (1).

Theorem 1. (Boundedness of the Metric) For a given data matrix \mathbf{X} , it holds for all $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$ simultaneously that:

$$0 \leq \mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) \leq 2. \quad (2)$$

The complete proof of the above theorem is presented in the supplementary materials. To evaluate the degree of linear correlation between samples, we state the linear relationship for Definition 1 in the following theorem.

Theorem 2. (Linear Dependence of the Metric) For any two samples \mathbf{x}_i and \mathbf{x}_j , when they satisfy:

$$\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) \in \{0, 2\}, \quad (3)$$

if and only if there is a linear relationship between \mathbf{x}_i and \mathbf{x}_j , that is, existing $u (u \neq 0)$ and v such that $\mathbf{x}_j = u\mathbf{x}_i + v$ (or $\mathbf{x}_i = u\mathbf{x}_j + v$). Among them, when $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = 0$, we have $u > 0$. When $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = 2$, we have $u < 0$.

The detailed proof is provided in the supplementary materials.

Remark 1: Theorem 2 shows that the metric we defined can describe the strength of the linear relationship between two samples. There are several explanations for this theorem:

- When $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = 0$, it is said that x_i and x_j are completely positive correlated. When $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = 2$, x_i and x_j are completely negative correlated.
- When $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) = 1$, that is, x_i and x_j are not correlated.
- When $0 < \mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) < 1$, it is said that x_i and x_j have a ‘‘certain degree’’ of positive linear relationship, if $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$ tends to 0, the degree of positive linear correlation is higher. If $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$ tends to 1, the degree of positive linear correlation is lower.
- When $1 < \mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) < 2$, it is said that x_i and x_j have a ‘‘certain degree’’ of negative linear relationship, if $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$ tends to 2, the degree of negative linear correlation is higher. If $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$ tends to 1, the degree of negative linear correlation is lower.

According to the above Theorem 2, the results illustrate that the linearity-aware metric $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j)$ can precisely characterize the linear relationship between \mathbf{x}_i and \mathbf{x}_j . Compared to the other metrics (*e.g.*, ℓ_1 -norm, nuclear norm and Frobenius norm), our proposed metric is essentially connected to self-expressive, *i.e.*, they are both to characterize the linear correlation between samples. Therefore, we introduce the linearity-aware metric into the subspace learning framework to enhance and guide the subspace structure.

Remark 2: The similarity matrix is constructed by employing the self-expressive, which is a linear assumption strategy, it makes the learned similarity matrix may not accurately describe the degree of linear correlation between samples. However, the linearity-aware metric can obtain the degree of the linear correlation between samples well, it can be illustrated in Fig. 3. Moreover, we naturally gain a ‘‘good’’ subspace structure by using our proposed metric.

Problem Formulation

It is well known that the purpose of subspace clustering is to discover underlying subspaces of the original data (You et al. 2020; Kang et al. 2021; Fan 2021). Remarkably, the self-expressive is generally conducted to represent itself to gain representation coefficient (Kang et al. 2020b; Xiao et al. 2021; Wen et al. 2021). Its usual model can be formulated as follows:

$$\min_{\mathbf{C}} \underbrace{\mathcal{L}_1(\mathbf{X}; \mathbf{C})}_{\text{Initial Subspace Learning}} = \min_{\mathbf{C}} \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 + \lambda_1 \Phi(\mathbf{C}), \quad (4)$$

where λ_1 is the tunable parameter. $\Phi(\mathbf{C})$ is a certain regularization. For simplicity, we here choose the Frobenius norm as a regularization (*i.e.*, $\Phi(\mathbf{C}) = \|\mathbf{C}\|_F^2$), which can preserve the grouping effect and avoid the trivial solution.

The self-expressive-based subspace clustering method has shown its superior performance in machine learning and computer vision (Lu et al. 2018; Li et al. 2020a; Chen et al. 2021c), but there are two non-negligible drawbacks. One is that it may fail to discover subspace structure sufficiently when the original data is directly utilized to acquire the similarity matrix (Liu and Yan 2011; Xu et al. 2021). The other is that the self-expressive strategy is hard to describe the linear relationship between samples for real-world data accurately, it makes the learned similarity matrix may be inaccurate.

To address the above-mentioned problems, we present a novel LASC model in this paper. First, by exploiting the Eq. (4) to learn a initial low-dimensional subspace $\mathbf{C} \in \mathbf{R}^{n \times n}$. Specifically, original data usually contains noisy or redundant features (Li et al. 2020b; Zhang et al. 2020a; Pan et al. 2021), we use the representation coefficient \mathbf{C} to represent the original data \mathbf{X} , thus alleviating the influence of redundancy or noise (Zhou et al. 2019; Peng et al. 2019, 2020). Then, motivated by the advance of adaptive neighbors learning (Nie, Wang, and Huang 2014; Wu et al. 2020), the i -th sample \mathbf{c}_i can be connected to other samples with probability s_{ij} , where s_{ij} is an element of similarity matrix \mathbf{S} . If the samples are close to each other, they have a higher similarity score, and if they are far apart, they have a smaller similarity score. To further boost the linear correlation of the self-expressive strategy, the linearity-aware metric is utilized to measure the representation coefficient, which can enhance the learned subspace structure and performs well in practice. Therefore, we have the following model:

$$\begin{aligned} & \min_{\mathbf{S}} \quad \underbrace{\mathcal{L}_2(\mathbf{C}; \mathbf{S})}_{\text{Linearity-Aware Subspace Learning}} \\ & = \min_{\mathbf{S}} \sum_{i=1}^n \sum_{j=1}^n \mathcal{D}^2(\mathbf{c}_i, \mathbf{c}_j) s_{ij} + \lambda_2 \Psi(\mathbf{S}), \quad (5) \\ & \text{s.t. } \mathbf{s}_i^\top \mathbf{1} = 1, 0 \leq s_{ij} \leq 1, \end{aligned}$$

where λ_2 is the parameter. For simplicity, the regularization term $\Psi(\mathbf{S}) = \|\mathbf{S}\|_F^2$ is adopted to constrain the similarity matrix \mathbf{S} .

However, the nature data may contain different factors (e.g., occlusion, illumination and expression) (Yang et al. 2016). The single-step learning representation coefficient \mathbf{C} and similarity matrix \mathbf{S} is not enough to eliminate the influence of those undesirable factors and obtain an ideal similarity representation. Considering these reasons, we build a collaborative learning framework to seek a ideal similarity matrix with more consistent knowledge to promote the subsequent clustering. Then, we utilize similarity matrix \mathbf{S} as the input of Eq. (4) to learn representation coefficient \mathbf{C} . Thus, we have:

$$\min_{\mathbf{C}} \quad \underbrace{\mathcal{L}_1(\mathbf{S}; \mathbf{C})}_{\text{Self-Expressive Subspace Learning}} = \min_{\mathbf{C}} \|\mathbf{S} - \mathbf{S}\mathbf{C}\|_F^2 + \lambda_1 \Phi(\mathbf{C}). \quad (6)$$

Finally, we present a greedy process to alternate multiple times learn the similarity matrix, inspired by the recursive thoughts, the single-layer mode can only discover the ‘‘shallow’’ features, which cannot discover deep hidden features and hierarchical information. We adopt a collaborative learning strategy to obtain the final ideal similarity matrix by Eq. (5) and Eq. (6), which can make a good input more conducive to obtaining a good result, and a stable solution can be acquired by alternating multiple times. Thus, we propose Linearity-Aware Subspace Clustering (LASC) model:

$$\begin{aligned} & \min_{\mathbf{C}, \mathbf{S}} \quad \underbrace{\mathcal{L}_2(\mathbf{C}; \mathbf{S})}_{\text{Linearity-Aware Subspace Learning}} + \underbrace{\mathcal{L}_1(\mathbf{S}; \mathbf{C})}_{\text{Self-Expressive Subspace Learning}}, \\ & \text{s.t. } \mathbf{s}_i^\top \mathbf{1} = 1, 0 \leq s_{ij} \leq 1, \quad (7) \end{aligned}$$

Algorithm 1: Subspace Segmentation via LASC

Input: The data matrix \mathbf{X} , parameters λ_1, λ_2 , the number of subspaces k and maximal iteration number T .

Initialize: $\mathbf{C}_0 = \mathbf{S}_0 = \mathbf{0}$ and $\varepsilon_1 = 10^{-4}, \varepsilon_2 = 10^{-5}$.

Obtain the initial subspace: Learning the initial subspace \mathbf{C} of \mathbf{X} by adopting Eq. (9).

While not converge ($t = 0, 1, \dots, T$) **do**

1). Obtain the linearity-aware subspace: Learning the linearity-aware subspace \mathbf{S}^{t+1} of \mathbf{C}^t by adopting Eq. (16).

2). Obtain the self-expressive subspace: Learning the subspace \mathbf{C}^{t+1} of \mathbf{S}^{t+1} by adopting Eq. (20).

3). Check convergence: If

$$\begin{aligned} & \max\{\sum_{i=1}^n \sum_{j=1}^n \mathcal{D}^2(\mathbf{c}_i^{t+1}, \mathbf{c}_j^{t+1}) s_{ij}^{t+1} / \|\mathbf{X}\|_F, \\ & \|\mathbf{S}^{t+1} - \mathbf{S}^{t+1} \mathbf{C}^{t+1}\|_F^2 / \|\mathbf{X}\|_F^2\} < \varepsilon_1 \\ & \text{and } \max\{\|\mathbf{S}^{t+1} - \mathbf{S}^t\|_F, \|\mathbf{C}^{t+1} - \mathbf{C}^t\|_F\} < \varepsilon_2, \\ & \text{then break.} \end{aligned}$$

End while

Obtain the affinity matrix by $|\mathbf{S}^*| + |(\mathbf{S}^*)^\top|$.

Apply the spectral clustering algorithm (Ng, Jordan, and Weiss 2002) to segment the data into k subspaces.

Output: The final clustering result.

Optimization

Once given the proper values of parameters λ_1 and λ_2 , we first compute the representation coefficient \mathbf{C} by utilizing Eq. (4). Then, we can adopt iteratively strategies to optimize the problems (5) and (6).

Step 1: The variable \mathbf{C} of problem (4) can be solved by the following problem:

$$\mathbf{C}^0 = \arg \min_{\mathbf{C}} \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 + \lambda_1 \|\mathbf{C}\|_F^2, \quad (8)$$

the above problem is the well-known ridge regression (Yang et al. 2016; Chen et al. 2021b). We take the derivative of the Eq. (8) *w.r.t.* \mathbf{Z} , and setting the derivative to zero. It has a closed-form solution:

$$\mathbf{C}^0 = (\mathbf{X}^\top \mathbf{X} + \lambda_1 \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{X}. \quad (9)$$

Step 2: The variable \mathbf{S} of the problem (5) can be optimized by solving the following optimization problem:

$$\begin{aligned} \mathbf{S}^{t+1} &= \arg \min_{\mathbf{S}} \sum_{i=1}^n \sum_{j=1}^n \mathcal{D}^2(\mathbf{c}_i^t, \mathbf{c}_j^t) s_{ij} + \lambda_2 \|\mathbf{S}\|_F^2, \quad (10) \\ \text{s.t. } \mathbf{s}_i^\top \mathbf{1} &= 1, 0 \leq s_{ij} \leq 1. \end{aligned}$$

By defining $d_{ij}^t = \mathcal{D}^2(\mathbf{c}_i^t, \mathbf{c}_j^t)$ for the convenience of notation, the above problem can be further rewritten as following independent problem:

$$\begin{aligned} \mathbf{s}_i^{t+1} &= \arg \min_{\mathbf{s}_i} \|\mathbf{s}_i + \frac{1}{2\lambda_2} \mathbf{d}_i^t\|_2^2, \quad (11) \\ \text{s.t. } \mathbf{s}_i^\top \mathbf{1} &= 1, 0 \leq s_{ij} \leq 1. \end{aligned}$$

To solve the Eq. (11), the Lagrangian function of Eq. (11) is shown below:

$$f(\mathbf{s}_i, \mu, \nu) = \frac{1}{2} \|\mathbf{s}_i + \frac{1}{2\lambda_2} \mathbf{d}_i^t\|_2^2 - \mu(\mathbf{s}_i^\top \mathbf{1} - 1) - \nu^\top \mathbf{s}_i, \quad (12)$$

where μ and ν are Lagrangian multipliers. When the Lagrange multipliers are μ^* and ν^* , suppose the optimal solu-

Datasets	# of Classes	Samples per Class	# of Features
MNIST	10	1000	784
USPS	10	500	256
COIL100	100	72	1024
CIFAR10	10	800	2048
ExYaleB	38	64	1024

Table 1: Description of datasets

tion of the Eq. (12) is \mathbf{s}_i^* . Based on the Karush-Kuhn-Tucker (KKT) conditions, for $\forall j$, we can obtain:

$$\begin{cases} s_{ij}^* + \frac{1}{2\lambda_2} d_{ij} - \mu^* - \nu_j^* = 0, \\ s_{ij}^* \geq 0, \\ \nu_j^* \geq 0, \\ s_{ij}^* \nu_j^* = 0. \end{cases} \quad (13)$$

Due to $\mathbf{1}^\top \mathbf{s}_i^* = 1$ and $\mathbf{s}_i^* + \frac{1}{2\lambda_2} \mathbf{d}_i - \mu^* \mathbf{1} - \nu^* = \mathbf{0}$, we obtain $\mu^* = \frac{1 + \frac{\mathbf{1}^\top \mathbf{d}_i - \mathbf{1}^\top \nu}{2\lambda_2}}{n}$. Therefore, we gain the optimal solution:

$$s_i^* = \left(-\frac{1}{2\lambda_2} \mathbf{d}_i + \frac{1}{n} \mathbf{1} + \frac{\mathbf{1}^\top \mathbf{d}_i}{2n\lambda_2} \mathbf{1} - \frac{\mathbf{1}^\top \nu}{n} \mathbf{1} \right) + \nu. \quad (14)$$

For the convenience of notation, we denote $\hat{\mathbf{d}}_i = -\frac{1}{2\lambda_2} \mathbf{d}_i + \frac{1}{n} \mathbf{1} + \frac{\mathbf{1}^\top \mathbf{d}_i}{2n\lambda_2} \mathbf{1}$ and $\hat{\nu} = \frac{\mathbf{1}^\top \nu}{n} \mathbf{1}$. Then, the Eq. (14) can be rewrite as below:

$$s_i^* = \hat{\mathbf{d}}_i - \hat{\nu}^* + \nu^*. \quad (15)$$

Obviously, for $\forall j$, we get:

$$s_{ij}^* = \hat{\mathbf{d}}_{ij} - \hat{\nu}_j^* + \nu_j^* = \max(\hat{\mathbf{d}}_{ij} - \hat{\nu}_j^*, 0). \quad (16)$$

Then, we obtain $s_j^* = \max(\hat{\mathbf{d}}_j - \hat{\nu}^*, 0)$. Similarly, we have $\nu_j^* = \max(\hat{\nu}^* - \hat{\mathbf{d}}_j, 0)$. To obtain the solution $\hat{\nu}^*$, we define a function as below:

$$g(\hat{\nu}) = \frac{1}{n} \sum_{j=1}^n \max(\hat{\nu} - \hat{\mathbf{d}}_j, 0) - \hat{\nu}. \quad (17)$$

According to $\nu^* \geq 0$, $g'(\hat{\nu}) \leq 0$ and $g'(\hat{\nu})$ is a convex function and piecewise linear function, the Newton method is utilized to acquire the root $\hat{\nu}^*$ of $g(\hat{\nu}) = 0$, which means:

$$\hat{\nu}^{t+1} = \hat{\nu}^t - \frac{g(\hat{\nu}^t)}{g'(\hat{\nu}^t)}, \quad (18)$$

where t is the number of iteration.

Step 3: The variable \mathbf{C} of problem (6) can be updated by:

$$\mathbf{C}^{t+1} = \arg \min_{\mathbf{C}} \|\mathbf{S}^{t+1} - \mathbf{S}^{t+1} \mathbf{C}\|_F^2 + \lambda_1 \|\mathbf{C}\|_F^2, \quad (19)$$

Calculating the derivative of the above Eq. (19) *w.r.t.* \mathbf{C} and setting it to zero, we obtain:

$$\mathbf{C}^{t+1} = \left[(\mathbf{S}^{t+1})^\top \mathbf{S}^{t+1} + \lambda_1 \mathbf{I} \right]^{-1} (\mathbf{S}^{t+1})^\top \mathbf{S}^{t+1}. \quad (20)$$

Overall, we summarize the whole optimization procedure in Algorithm 1.

Complexity Analysis

We assess the computational complexity of Algorithm 1. The major computational burden depends on the optimiza-

tion of \mathbf{C} and \mathbf{S} . Specifically, step 1 contains matrix multiplication and matrix inversion operations for obtaining matrix \mathbf{C} , and its complexity is $\mathcal{O}(mn^2 + n^3)$, where m and n are the dimension of the data and the number of the data points, respectively. We need $\mathcal{O}(mn + n^2)$ for updating matrix \mathbf{S} of step 2. The step 3 can be solved by Eq. (20), whose computational complexity is $\mathcal{O}(n^3)$. To summarize, the overall computational complexity of Algorithm 1 is $\mathcal{O}(mn^2 + n^3 + t(mn + n^3))$, here t is the number of collaborations.

Experimental Results

Experimental Settings

Datasets: In our experiments, five benchmark datasets are selected, including two handwritten digits datasets MNIST¹ and USPS², two object recognition datasets COIL100³ and CIFAR10⁴, and an face image dataset Extended Yale B(ExYaleB)⁵.

In addition, to achieve better performance on the CIFAR10 dataset, we extract features from the CIFAR10 dataset following the same setting in (Xu et al. 2020). In this paper, we randomly selected 1000, 500, and 800 samples for each class from the MNIST, USPS and CIFAR10 datasets, respectively. Table 1 introduces the information of these benchmark datasets, which are used in our experiments.

Compared Methods: We compare our proposed LASC algorithm against the following four spectral-based subspace clustering methods, including SSC (Elhamifar and Vidal 2013), LSR (Lu et al. 2012), LRR (Liu et al. 2013) and EnSC (You et al. 2016). Three metric-based subspace clustering strategies are also adopted, including FGNSC (Yang et al. 2020), OSC (Wang et al. 2019) and autoSC (Liang et al. 2019).

For all above mentioned algorithms, the parameters are tuned by the cross validation technique to guarantee their possibly optimal performance⁶.

Evaluation Measures: To evaluate the performance of different subspace clustering methods, we employ two popular metrics to evaluate clustering performance, each of which favors different properties of clustering, including Accuracy (ACC) and Normalized Mutual Information (NMI), the detailed definitions of ACC and NMI can be seen in (Xu et al. 2021). The above two metrics both lie in the range of $[0, 1]$, and the higher value indicates better clustering performance. Note that for fair comparisons, we report the mean values and standard derivations of 10 independent trials.

¹<http://yann.lecun.com/exdb/mnist/>

²<https://paperswithcode.com/dataset/usps>

³<https://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php>

⁴<https://www.cs.toronto.edu/~kriz/cifar.html>

⁵<http://vision.ucsd.edu/leekc/ExtYaleDatabase/ExtYaleB.html>

⁶The following parameters should be tuned: SSC (γ and δ), LSR (λ), LRR (λ), EnSC (λ and γ), FGNSC (η , γ and μ), OSC (α , β and δ), and autoSC (m and \hat{K}).

Classes	Metrics & Para.	SSC	LSR	LRR	EnSC	FGNSC	OSC	autoSC	LASC
6	ACC	0.565±0.014	0.642±0.008	0.664±0.015	0.655±0.023	0.679±0.026	0.685±0.024	0.694±0.022	0.727±0.021
	NMI	0.498±0.035	0.532±0.011	0.596±0.009	0.648±0.017	0.653±0.021	0.656±0.019	0.673±0.026	0.703±0.015
	Para.	0.001, 0.1	0.3	0.05	0.05, 0.1	20, 8, 1	0.2, 0.01, 0.002	8, 10	0.5, 1000
8	ACC	0.563±0.009	0.623±0.011	0.637±0.021	0.558±0.028	0.658±0.027	0.652±0.027	0.679±0.028	0.693±0.025
	NMI	0.594±0.007	0.568±0.015	0.588±0.017	0.616±0.022	0.622±0.020	0.630±0.019	0.651±0.026	0.678±0.020
	Para.	0.002, 0.05	0.6	0.02	0.1, 0.001	20, 8, 1	0.5, 0.04, 0.002	8, 15	1, 5000
10	ACC	0.465±0.012	0.486±0.016	0.524±0.009	0.563±0.025	0.554±0.015	0.548±0.012	0.560±0.016	0.581±0.011
	NMI	0.527±0.021	0.445±0.014	0.509±0.006	0.572±0.009	0.547±0.018	0.535±0.017	0.559±0.012	0.594±0.016
	Para.	0.8, 0.01	0.7	0.1	0.5, 0.02	20, 8, 1	0.2, 0.04, 0.002	8, 20	0.1, 1000
6	ACC	0.804±0.003	0.722±0.006	0.721±0.002	0.742±0.008	0.824±0.012	0.842±0.005	0.872±0.008	0.861±0.009
	NMI	0.791±0.002	0.763±0.003	0.778±0.003	0.813±0.009	0.805±0.017	0.810±0.007	0.855±0.012	0.842±0.006
	Para.	1, 10	1	0.05	0.9, 0.001	25, 10, 1	0.3, 0.03, 0.002	15, 10	0.5, 1000
8	ACC	0.783±0.006	0.759±0.005	0.766±0.002	0.771±0.009	0.802±0.007	0.817±0.005	0.831±0.008	0.847±0.010
	NMI	0.812±0.007	0.747±0.003	0.758±0.008	0.815±0.003	0.827±0.010	0.806±0.004	0.814±0.004	0.836±0.005
	Para.	5, 0.1	0.1	0.0001	0.05, 0.001	25, 10, 1	0.1, 0.05, 0.002	15, 15	0.5, 500
10	ACC	0.776±0.006	0.721±0.002	0.727±0.004	0.769±0.006	0.784±0.021	0.785±0.013	0.796±0.012	0.821±0.012
	NMI	0.796±0.003	0.699±0.011	0.703±0.007	0.791±0.008	0.812±0.018	0.798±0.013	0.802±0.008	0.822±0.009
	Para.	0.1, 0.002	0.5	0.01	0.2, 1	25, 10, 1	0.2, 0.02, 0.002	15, 20	0.1, 500
80	ACC	0.395±0.056	0.384±0.036	0.397±0.044	0.430±0.047	0.452±0.042	0.445±0.047	0.438±0.051	0.505±0.046
	NMI	0.689±0.053	0.605±0.042	0.665±0.058	0.717±0.033	0.735±0.048	0.716±0.057	0.733±0.058	0.787±0.052
	Para.	0.002, 0.5	1	0.2	0.2, 2	20, 10, 1	0.4, 0.01, 0.002	20, 80	0.5, 10000
100	ACC	0.384±0.040	0.375±0.034	0.362±0.035	0.426±0.041	0.447±0.037	0.426±0.034	0.419±0.041	0.473±0.037
	NMI	0.674±0.049	0.589±0.046	0.602±0.041	0.682±0.040	0.701±0.035	0.696±0.048	0.675±0.031	0.755±0.031
	Para.	0.0001, 0.05	0.7	0.05	0.05, 0.01	25, 8, 1	0.5, 0.03, 0.002	20, 100	0.05, 50000
8	ACC	0.723±0.024	0.779±0.022	0.765±0.021	0.761±0.019	0.805±0.023	0.836±0.021	0.812±0.027	0.829±0.019
	NMI	0.618±0.012	0.634±0.016	0.625±0.023	0.630±0.017	0.651±0.031	0.678±0.025	0.656±0.022	0.682±0.023
	Para.	2, 0.002	0.4	0.2	0.8, 0.001	20, 8, 1	0.2, 0.04, 0.002	10, 15	1, 1000
10	ACC	0.745±0.014	0.783±0.017	0.779±0.022	0.773±0.014	0.792±0.018	0.804±0.026	0.803±0.021	0.814±0.018
	NMI	0.663±0.024	0.659±0.011	0.657±0.027	0.669±0.025	0.673±0.031	0.685±0.024	0.693±0.027	0.712±0.026
	Para.	5, 0.1	0.4	0.01	0.1, 0.5	25, 10, 1	0.2, 0.01, 0.002	10, 20	5, 500
30	ACC	0.658±0.014	0.642±0.027	0.639±0.029	0.662±0.018	0.676±0.018	0.685±0.028	0.703±0.024	0.735±0.032
	NMI	0.652±0.031	0.663±0.026	0.649±0.033	0.648±0.024	0.681±0.035	0.702±0.037	0.721±0.029	0.758±0.035
	Para.	0.001, 0.02	0.8	0.05	0.6, 0.005	20, 8, 1	0.4, 0.02, 0.002	8, 30	0.5, 1
38	ACC	0.627±0.021	0.665±0.004	0.624±0.028	0.672±0.028	0.686±0.025	0.699±0.027	0.715±0.019	0.742±0.024
	NMI	0.632±0.016	0.625±0.007	0.636±0.021	0.653±0.011	0.684±0.027	0.715±0.039	0.746±0.022	0.768±0.019
	Para.	0.01, 0.5	0.2	0.5	0.8, 0.1	20, 8, 1	0.3, 0.04, 0.002	8, 40	0.5, 0.5

Table 2: Performance comparison of all compared methods on the five benchmark datasets. From top to bottom, they are MNIST, USPS, COIL100, CIFAR10 and ExYaleB, respectively.

Clustering Performance and Analysis

To sufficiently evaluate the performance of our proposed method on the five benchmark datasets, for MNIST, USPS, COIL100, CIFAR10 and ExYaleB datasets, $k = 6, 8, 10$, $k = 6, 8, 10$, $k = 80, 100$, $k = 8, 10$, and $k = 30, 38$ classes are randomly selected as the data matrix \mathbf{X} , respectively. Table 2 presents the clustering results and the tuned parameters of all tested approaches.

From Table 2, we can draw the following conclusions. Our LASC performs much better than the compared methods of almost all metrics on the five benchmark datasets, validating the effectiveness and superiority of our proposed LASC method. For example, on the COIL100 data with 100 classes, LASC obtains about 0.47 ACC and 0 NMI, which are about 0.03 and 0.05 higher than those of the second-best algorithm, respectively. Metric-based subspace clustering algorithms (*i.e.*, FGNSC, OSC, auto-SC and LASC) yield better clustering results in comparison with spectral-based subspace clustering algorithms (*i.e.*, SSC, LSR, LRR and EnSC) in most cases, which shows the importance of integrating metric learning into the subspace clustering model. LASC achieves better than the advanced metric-based subspace clustering methods FGNSC, OSC and auto-SC in most cases. This demonstrates that LASC is superior to existing metric-based methods for subspace clustering.

In order to further provide an intuitive illustration of the

effectiveness of our models, we utilize t-distributed Stochastic Neighbor Embedding (t-SNE) to show the distribution of the similarity matrices learned by different algorithms on the handwritten digits dataset USPS. From the experimental results shown in Figure 5, we can observe that our LASC model gains a more compact and accurate class cluster than the other methods (*i.e.*, SSC, LSR, LRR and OSC). It demonstrates that the similarity matrix learned by our model is more suitable than the other methods.

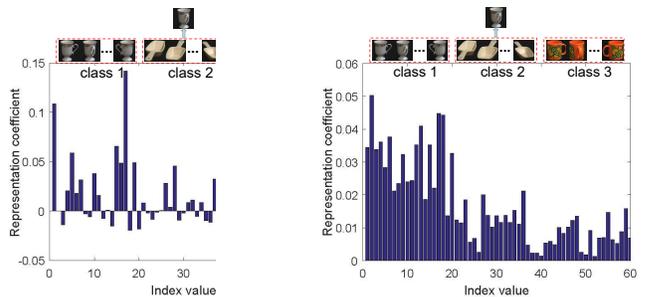


Figure 4: Visualization of the learned representation coefficients on the COIL100 dataset with 3 classes and 60 samples. From left to right, they are SSC and LASC methods, respectively.

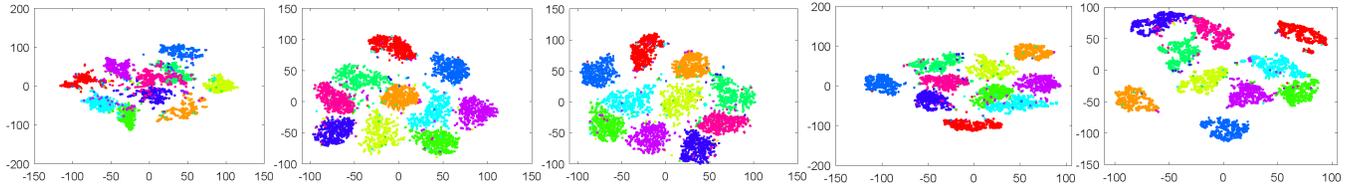


Figure 5: Visualization of the learned similarity matrices of different methods on the USPS dataset via t-SNE. From left to right, they are SSC, LSR, LRR, OSC and LASC, respectively.

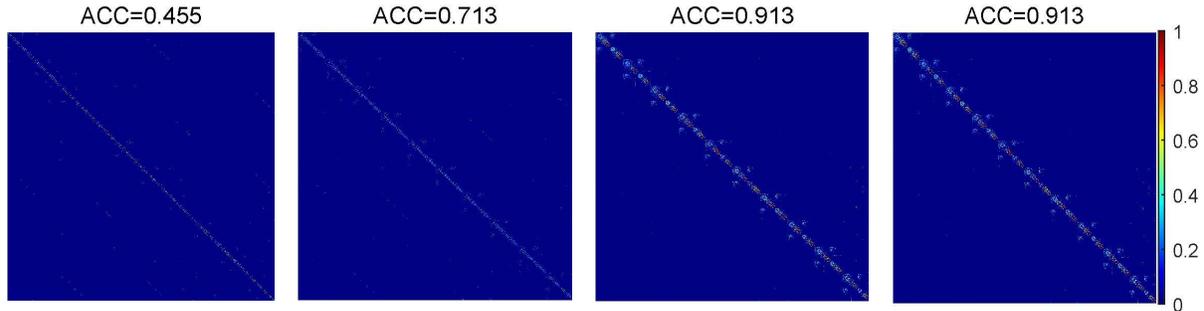


Figure 6: Visualization of the learned similarity matrices by using the proposed LASC method on ExYaleB dataset with 10 classes. From left to right, $t = 1, 5, 20$ and 50 , respectively.

The Analysis of Representation Coefficient

To further investigate the effectiveness of our LASC approach, we conduct experiments by showing the learned representation coefficient on the COIL100 dataset. Here, 10 classes are utilized in this example. We randomly select samples from each class, and treat each sample as a column to form original data $\mathbf{X} \in \mathbb{R}^{1024 \times 60}$. Then, the representation coefficients for a sample \mathbf{x}_2 are obtained by SSC, LASC methods, and are illustrated in Figure 4. The x-axis represents index value, and y-axis represents representation coefficient. From the results, the large coefficient value not ideally concentrate on the first subpart (*i.e.*, the sample \mathbf{x}_2 is derived from class 1) as expected for SSC. However, the coefficient values obtained by our method are very large in class 1, which accurately reflects the similarity between samples. It will contribute to the subsequent clustering tasks.

Influence of the Number of Collaboration

To analyze the benefit of collaborative learning the similarity matrix \mathbf{S} of our LASC, in this subsection, we visually present some examples of the learned similarity matrix by different numbers of collaborations t . Without loss of generality, we show the similarity matrix on ExYaleB dataset with 10 classes, where we consider the cases of $t = 1, 5, 20$ and 50 , respectively. Figure 6 shows these learned similarity matrices. Obviously, we can observe that the collaborative learning strategy can improve the clustering performance because the performance increases with t . Hence, it well demonstrates the effectiveness of the collaborative strategy.

Comparison of Running Time

Figure 7 illustrates the running time of different algorithms on five benchmark datasets. We can find that the running

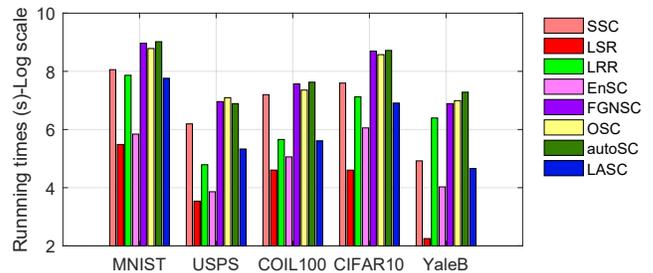


Figure 7: Running time of different approaches on the five benchmark datasets.

time of our LASC model is roughly at the same level as most self-expressive-based algorithms (*e.g.*, SSC and LRR) under comparison. In addition, our LASC model is much faster than metric-based methods on the five datasets. We can conclude that, compared with the state-of-the-art methods, our model is able to greatly improve the clustering performance without increasing the running time.

Conclusion

To enhance self-expressive to characterize the linear correlation between samples in subspace clustering, this article proposed a linearity-aware metric that overcomes the difficulty mentioned above, and the proposed metric is integrated into the subspace clustering model. In addition, we theoretically illustrated that the proposed linearity-aware metric could measure the linear correlation between samples. Then, we utilized collaboration strategy to learn the self-expressive subspace and linearity-aware subspace, and discovered an ideal similarity matrix, thereby leading to the superior performance of the subsequent clustering results.

Acknowledgments

This work was supported by the National Science Foundation of China under Grant U1713208, 62072242, 61876083, 62176124, and the China Postdoctoral Science Foundation under Grant 2020M681606.

References

- Benesty, J.; Chen, J.; Huang, Y.; and Cohen, I. 2009. Pearson correlation coefficient. In *Noise reduction in speech process.*, 1–4.
- Chen, S.; Gong, C.; Yang, J.; Li, X.; Wei, Y.; and Li, J. 2018. Adversarial metric learning. *Proc. Int. Joint Conf. Artif. Intell.*, 2021–2027.
- Chen, S.; Niu, G.; Gong, C.; Li, J.; Yang, J.; and Sugiyama, M. 2021a. Large-Margin Contrastive Learning with Distance Polarization Regularizer. In *Proc. Int. Conf. Mach. Learn.*, 1673–1683.
- Chen, S.; Yang, J.; Wei, Y.; Luo, L.; Lu, G.-F.; and Gong, C. 2021b. δ -Norm-Based Robust Regression With Applications to Image Analysis. *IEEE Trans. Cybern.*, 51(6): 3371–3383.
- Chen, Y.; Wang, S.; Xiao, X.; Liu, Y.; Hua, Z.; and Zhou, Y. 2021c. Self-paced Enhanced Low-rank Tensor Kernelized Multi-view Subspace Clustering. *IEEE Trans. Multimedia*, <https://doi.org/10.1109/TMM.2021.3112230>.
- Elhamifar, E.; and Vidal, R. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11): 2765–2781.
- Fan, J. 2021. Large-Scale Subspace Clustering via k-Factorization. In *Proc. 27th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 342–352.
- Kang, Z.; Lin, Z.; Zhu, X.; and Xu, W. 2021. Structured Graph Learning for Scalable Subspace Clustering: From Single View to Multiview. *IEEE Trans. Cybern.*, <https://doi.org/10.1109/TCYB.2021.3061660>.
- Kang, Z.; Zhao, X.; Peng, C.; Zhu, H.; Zhou, J. T.; Peng, X.; Chen, W.; and Xu, Z. 2020a. Partition level multiview subspace clustering. *Neural Netw.*, 122: 279–288.
- Kang, Z.; Zhou, W.; Zhao, Z.; Shao, J.; Han, M.; and Xu, Z. 2020b. Large-scale multi-view subspace clustering in linear time. In *Proc. AAAI Conf. Artif. Intell.*, 4412–4419.
- Li, J.; Kong, Y.; and Fu, Y. 2017. Sparse subspace clustering by learning approximation ℓ_0 codes. In *Proc. AAAI Conf. Artif. Intell.*, 2189–2195.
- Li, J.; Kong, Y.; Zhao, H.; Yang, J.; and Fu, Y. 2016. Learning fast low-rank projection for image classification. *IEEE Trans. Image Process.*, 25(10): 4803–4814.
- Li, J.; Liu, H.; Tao, Z.; Zhao, H.; and Fu, Y. 2020a. Learnable subspace clustering. *IEEE Trans. Neural Netw. Learn. Syst.*, <https://doi.org/10.1109/TNNLS.2020.3040379>.
- Li, J.; Sun, G.; Zhao, G.; and Li-wei, H. L. 2020b. Robust low-rank discovery of data-driven partial differential equations. In *Proc. AAAI Conf. Artif. Intell.*, 767–774.
- Li, J.; Tao, Z.; Wu, Y.; Zhong, B.; and Fu, Y. 2021. Large-Scale Subspace Clustering by Independent Distributed and Parallel Coding. *IEEE Trans. Cybern.*, <https://doi.org/10.1109/TCYB.2021.3052056>.
- Liang, J.; Yang, J.; Cheng, M.-M.; Rosin, P. L.; and Wang, L. 2019. Simultaneous subspace clustering and cluster number estimating based on triplet relationship. *IEEE Trans. Image Process.*, 28(8): 3973–3985.
- Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; and Ma, Y. 2013. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1): 171–184.
- Liu, G.; and Yan, S. 2011. Latent low-rank representation for subspace segmentation and feature extraction. In *Proc. IEEE Int. Conf. Comput. Vis.*, 1615–1622.
- Lu, C. 2021. Transforms Based Tensor Robust PCA: Corrupted Low-Rank Tensors Recovery via Convex Optimization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1145–1152.
- Lu, C.; Feng, J.; Lin, Z.; Mei, T.; and Yan, S. 2018. Subspace clustering by block diagonal representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(2): 487–501.
- Lu, C. Y.; Min, H.; Zhao, Z. Q.; Zhu, L.; Huang, D. S.; and Yan, S. 2012. Robust and efficient subspace segmentation via least squares regression. In *Proc. Eur. Conf. Comput. Vis.*, 347–360.
- Lv, J.; Kang, Z.; Lu, X.; and Xu, Z. 2021. Pseudo-supervised Deep Subspace Clustering. *IEEE Trans. Image Process.*, 30(3): 5252–5263.
- McWilliams, B.; and Montana, G. 2014. Subspace clustering of high-dimensional data: a predictive approach. *Data Mining Knowl. Discov.*, 28(3): 736–772.
- Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *Proc. Neural Inf. Process. Syst.*, 849–856.
- Nie, F.; Chang, W.; Hu, Z.; and Li, X. 2020. Robust Subspace Clustering with Low-Rank Structure Constraint. *IEEE Trans. Knowl. Data Eng.*, [10.1109/TKDE.2020.2995896](https://doi.org/10.1109/TKDE.2020.2995896).
- Nie, F.; Wang, X.; and Huang, H. 2014. Clustering and projected clustering with adaptive neighbors. In *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 977–986.
- Pan, E.; et al. 2021. Multi-view Contrastive Graph Clustering. In *Proc. Neural Inf. Process. Syst.*
- Peng, X.; Feng, J.; Xiao, S.; Yau, W.-Y.; Zhou, J. T.; and Yang, S. 2018. Structured autoencoders for subspace clustering. *IEEE Trans. Image Process.*, 27(10): 5076–5086.
- Peng, X.; Feng, J.; Zhou, J. T.; Lei, Y.; and Yan, S. 2020. Deep subspace clustering. *IEEE Trans. Neural Netw. Learn. Syst.*, 31(12): 5509–5521.
- Peng, X.; Zhu, H.; Feng, J.; Shen, C.; Zhang, H.; and Zhou, J. T. 2019. Deep clustering with sample-assignment invariance prior. *IEEE Trans. Neural Netw. Learn. Syst.*, 31(11): 4857–4868.

- Peng, Z.; Jia, Y.; Liu, H.; Hou, J.; and Zhang, Q. 2021. Maximum Entropy Subspace Clustering Network. *IEEE Trans. Circuits Syst. Video Technol.*, <https://doi.org/10.1109/TCSVT.2021.3089480>.
- Wang, J.; Suzuki, A.; Xu, L.; Tian, F.; Yang, L.; and Yamashita, K. 2019. Orderly Subspace Clustering. In *Proc. AAAI Conf. Artif. Intell.*, 5264–5272.
- Wen, J.; Zhang, Z.; Zhang, Z.; Zhu, L.; Fei, L.; Zhang, B.; and Xu, Y. 2021. Unified Tensor Framework for Incomplete Multi-view Clustering and Missing-view Inferring. In *Proc. AAAI Conf. Artif. Intell.*, 10273–10281.
- Wu, J.; Xie, X.; Nie, L.; Lin, Z.; and Zha, H. 2020. Unified graph and low-rank tensor learning for multi-view clustering. In *Proc. AAAI Conf. Artif. Intell.*, 6388–6395.
- Xiao, X.; Gong, Y.-J.; Hua, Z.; and Chen, W.-N. 2021. On Reliable Multi-View Affinity Learning for Subspace Clustering. *IEEE Trans. Multimedia*, 23(2): 4555–4566.
- Xu, Y.; Chen, S.; Li, J.; Han, Z.; and Yang, J. 2020. Autoencoder-based latent block-diagonal representation for subspace clustering. *IEEE Trans. Cybern.*, <https://doi.org/10.1109/TCYB.2020.3031666>.
- Xu, Y.; Chen, S.; Li, J.; Luo, L.; and Yang, J. 2021. Learnable Low-Rank Latent Dictionary for Subspace Clustering. *Pattern Recognit.*, <https://doi.org/10.1016/j.patcog.2021.108142>.
- Yang, J.; Liang, J.; Wang, K.; Rosin, P. L.; and Yang, M.-H. 2020. Subspace clustering via good neighbors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(6): 1537–1544.
- Yang, J.; Luo, L.; Qian, J.; Tai, Y.; Zhang, F.; and Xu, Y. 2016. Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(1): 156–171.
- You, C.; Li, C.; Robinson, D.; and Vidal, R. 2020. Self-representation based unsupervised exemplar selection in a union of subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, <https://doi.org/10.1109/TPAMI.2020.3035599>.
- You, C.; Li, C.-G.; Robinson, D. P.; and Vidal, R. 2016. Oracle based active set algorithm for scalable elastic net subspace clustering. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 3928–3937.
- Zhang, C.; Cui, Y.; Han, Z.; Zhou, J. T.; Fu, H.; and Hu, Q. 2020a. Deep partial multi-view learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, <https://doi.org/10.1109/TPAMI.2020.3037734>.
- Zhang, C.; Fu, H.; Hu, Q.; Cao, X.; Xie, Y.; Tao, D.; and Xu, D. 2018. Generalized latent multi-view subspace clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(1): 86–99.
- Zhang, C.; Fu, H.; Wang, J.; Li, W.; Cao, X.; and Hu, Q. 2020b. Tensorized multi-view subspace representation learning. *Int. J. Comput. Vis.*, 128(8): 2344–2361.
- Zhang, S.; You, C.; Vidal, R.; and Li, C.-G. 2021. Learning a Self-Expressive Network for Subspace Clustering. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 12393–12403.
- Zhou, L.; Xiao, B.; Liu, X.; Zhou, J.; Hancock, E. R.; et al. 2019. Latent distribution preserving deep subspace clustering. In *Proc. Int. Joint Conf. Artif. Intell.*, 4440–4446.
- Zhou, T.; Fu, H.; Gong, C.; Shen, J.; Shao, L.; and Porikli, F. 2020. Multi-mutual consistency induced transfer subspace learning for human motion segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 10277–10286.